Mikko Hyvärinen

Detection of Distributed Denial-of-Service Attacks in Encrypted Network Traffic

Master's Thesis in Information Technology

December 9, 2016

University of Jyväskylä

Department of Mathematical Information Technology

Author: Mikko Hyvärinen

Contact information: hyvarinen.mikko@gmail.com

Supervisor: Timo Hämäläinen & Mikhail Zolotukhin

Title: Detection of Distributed Denial-of-Service Attacks in Encrypted Network Traffic

Työn nimi: Hajautettujen palvelunestohyökkäysten havainnointi salatussa verkkoliikenteessä

Project: Master's Thesis

Study line: Software Development

Page count: 122+9

Abstract: Context: Distributed denial-of-service attacks have existed for two decades. Various strategies have been developed to combat the increasing volume of attacks over the years. Application layer attacks are becoming more common, and they are harder to detect. Current detection methods analyze traffic features. The packet payload is encrypted in an SSL/TLS traffic, and it cannot be analyzed. Objective: The thesis studies the current situation of detection of DDoS attacks in an SSL/TLS encrypted traffic. Also, the thesis presents a K-means++ clustering-based detection method and comparable simulation results with the previous literature. Methods: The author conducted a light systematic mapping study by searching common computer science literature libraries. The author ran experiments with the clustering-based method in a virtual network. Results: The mapping study found that the detection methods concentrate on clustering and statistical anomaly detection methods. In the experiments, denial-of-service attack simulations revealed that the K-means++ clustering detects trivial DDoS attacks with near 100% accuracy. Datasets were found to be an important part when comparing results. Conclusion: The mapping study revealed encrypted denial-of-service research study areas where more research is needed when compared to the non-encrypted counterpart.

Keywords: DDoS, denial-of-service, encryption, network security, SSL, TLS, anomaly detection, systematic mapping study, simulation

Suomenkielinen tiivistelmä: Tausta: Hajautetut palvelunestohyökkäykset ovat jo kaksi vuosikymmentä vanhoja. Useita strategioita on kehitetty taistelemaan niiden kasvavaa määrää vastaan vuosien varrella. Sovelluskerroksen protokollien hyökkäykset yleistyvät, ja niitä on hankalampi havaita. Nykyiset havainnointimenetelmät analysoivat tietoliikenteen piirteitä. Paketin sisältö on salattua SSL/TLS liikenteessä, josta syystä sitä ei voida analysoida. Tavoitteet: Tutkielma tarkastelee salatun liikenteen palvelunestohyökkäysten havaintometodien nykyistä tilaa. Tutkielma esittelee myös klusterointiin perustuvan menetelmän ja aikaisemman tutkimuksen kanssa vertailtavissa olevia simulaatiotuloksia. Metodit: Kirjoittaja laati kevyen systemaattisen kirjallisuuskartoituksen etsien lähteitä tietotekniikan kirjallisuustietokannoista. Hän myös teki tutkimuksia klusterointimenetelmän (K-means++) kanssa käyttäen virtuaaliverkkoa. Tulokset: Kirjallisuuskartoitus löysi, että havainnointimenetelmät keskittyvät klusterointiin perustuviin ja tilastollisiin poikkeamienhavainnointimenetelmiin. Esitetty klusterointimenelmä havaitsi yksinkertaiset hyökkäykset lähes sadan prosentin tarkkuudella. Tietoaineiston laatu huomattiin tärkeäksi tulosten vertailun kannalta. Johtopäätökset: Kirjallisuuskartoitus havaitsi aukkoja tutkimuksessa verrattaessa sitä salaamattomien hyökkäysten havainnointiin. Näillä alueilla lisää tutkimusta tarvitaan.

Avainsanat: palvelunestohyökkäys, salaus, verkkoturvallisuus, SSL, TLS, poikkemien havainnointi, systemaattinen kirjallisuuskartoitus, simulaatio

Glossary

ACI	Availability, integrity and confidentiality. See also CIA or AIC
ACK	Acknowledgment-packet of the TCP handshake
ACM DL	The Association for Computing Machinery Digital Library
AIC	Availability, integrity and confidentiality. See also ACI or CIA
ANOVA	Analysis of variance
ARPANET	The ARPA (Advanced Research Projects Agency) Network
AUC	Area under the curve
AVG	Short for average
C&C	Short for command & control
CIA	Availability, integrity and confidentiality. See also AIC or ACI
CNSS	The Committee on National Security Systems
CPU	Central processing unit
CUSUM	Cumulative sum
DARPA	The Defense Advanced Research Projects Agency
DBSCAN	Density-based spatial clustering of applications with noise
DDoS	A distributed denial-of-service
DHCP	Dynamic host configuration protocol
DMZ	Demilitarized zone, a network segment
DNS	The domain name system
DOCSIS	The data over cable service interface specification
DoD	The Department of Defense
DoS	A denial-of-service or a denial of service
EC	Exclusion criteria, the mapping study
F5	A function 5 -button on a keyboard
FN	False negative
FP	False positive
FPR	False positive rate
FSA	A finite state automata
GET	An HTTP GET-request

Gbps	Gigabits per second
HIDS	A host-based intrusion detection system
HTML	Hypertext markup language
HTTP(S)	Hypertext transfer protocol, HTTPS over SSL/TLS
IC	Inclusion criteria, the mapping study
ICMP	The internet control message protocol
IDEVAL	Intrusion Detection Evaluation
IDPS	Intrusion detection and prevention system
IDS	An intrusion detection system
IDSSD	An intrusion detection Scenario Specific Dataset
IEC	International Electrotechnical Commission
IEEE	The Institute of Electrical and Electronics Engineers
IETF	The Internet Engineering Task Force
IGMP	The internet group management protocol
IMP	Interface message processor
IP	Internet protocol, IPv4 and IPv6
IPS	Intrusion prevention system
IRC	Internet relay chat, an instant messaging service
ISBN	International standard book number
ISO	International Standards Organization
ITU-T	International Telecommunications Union
IoT	Internet of things
JYU	Acronym for University of Jyväskylä
KDD	Knowledge discovery from data
LBNL	Lawrence Berkeley National Laboratory
LOIC	Low Orbit Ion Cannon
LTE	Long-Term evolution network standard
MANET	Mobile ad-hoc Networks
ML	Maximum likelihood
MLP	Multilayer perceptron
MRI	A magnetic resonance imaging -machine

M.Sc.	Master of Science
NAT	Network address translation
NCP	Network control protocol
NIDPS	Network intrusion detection and prevention system
NIDS	Network intrusion detection system
NN	Neural network
OC	Overall contribution
OLAP	Online analytical processing
OSI	Open Standards Interconnection
OpNet	Opportunistic networks
P2P	Peer to peer
PCA	Principal component analysis, Princ. Comp. An.
PCAP	Packet capture -file
PICO	Population, intervention, control, and outcome
PLC	Programmable logic controller
POST	An HTTP POST-request
Ph.D.	A Doctor of Philosophy
QC	Quality criteria
RAM	Random-access memory
RFC	A request for comments -publication
RGCE	Realistic Global Cyber Environment
RNN	Replicator neural network
ROC	Receiver operating characteristics
RQ	Research question
RUDY	R-U-Dead-Yet, a DoS tool
SAE	Stacked auto-encoder
SDN	Software-defined network
SMS	Short message service
SNA/IP	Systems network architecture over internet protocol
SOM	Self-organizing map
SQL	Structured query language

SSH	Secure shell
SSL	Secure sockets layer
SVDD	Support vector data description
SVM	Support vector machines
SYN	Synchronize-packet of the TCP handshake
SYN-ACK	Synchronize-acknowledgment-packet of the TCP handshake
TCP	Transmission control protocol
TCP/IP	Transmission control protocol over internet protocol
TFN2K	The Tribe Flood Network
TLS	Transport layer security
TN	True negative
TP	True positive
TPR	True positive rate
TTL	Time to live
UDP	The user datagram protocol
UNB ISCX	University of New Brunswick Information Security Centre of Excellence
US	The United States of America
US-CERT	United States Computer Emergency Readiness Team
VPN	Virtual private network
WBAN	Wireless body area networks
WSN	Wireless sensor network
XML	eXtensible markup language
k-NN	K-nearest neighbors

List of Figures

Figure 1. Centralized botnet 22
Figure 2. Decentralized botnet 22
Figure 3. A distributed denial-of-service attack using a botnet 25
Figure 4. Bandwidth of the volumetric attacks reported yearly since 2002 (Arbor Networks 2011, 15.) (Arbor Networks 2016, 24.) (Krebs 2016) 26
Figure 5. An example of an anomaly in clustered data in a 2-feature plane 33
Figure 6. A ROC-curve and an AUC-value calculated 41
Figure 7. Classification of intrusion detection and prevention systems and their detection methods (Mirkovic and Reiher 2004, 49.) (Whitman and Mattord 2011, 293-305.) 47
Figure 8. Classification of anomaly-based detection methods as extended by Patcha and Park (2007) and adopted hybrid methods from Tama and Rhee (2015, 3742.) .. 49
Figure 9. Selected papers published by year 66
Figure 10. The publication venue distribution of the included studies approximately 68
Figure 11. Detection methods by class in a bubble plot over the years 74
Figure 12. Detection methods classified in hybrid classes over the years 74
Figure 13. Virtual network simulation architecture 83
Figure 14. ROC DARPA'99 with K-means# & K-means++ 87
Figure 15. ROC slow HTTPS POST (RUDY) with K-means# & K-means++ 87
Figure 16. ROC Slowloris with K-means# & K-means++ 87

List of Tables

Table 1. Comparison of the OSI and the TCP/IP models (Blank 2006, 24.) 17
Table 2. A confusion matrix (Bradley 1997, 1146) 40
Table 3. Search term formulation 54
Table 4. Summary of the search terms by database 56
Table 5. Search results and paper yield per database 57
Table 6. Evaluation metrics from each data source 57
Table 7. Overlap matrix for each of the data source 57
Table 8. Form for data extraction 61
Table 9. Selected studies 63
Table 10. Included studies and their publication forums 65
Table 11. Detection methods in encrypted networks from included studies 67
Table 12. Applicable methods from non-encrypted research in included studies 68
Table 13. Sample DDoS attacks and datasets used in included studies 75
Table 14. K-means# & K-means++ detection accuracy comparison with other attacks 89
Table 15. Detection accuracy comparison with other methods 89
Table 16. Change record 111
Table 17. Scopus studies after inclusion criteria 116

Table 18. ACM Digital Library studies after inclusion criteria 117
Table 19. IEEE included studies after inclusion criteria 118
Table 20. ScienceDirect studies after inclusion criteria 119

Contents

1 INTRODUCTION 1
1.1 Background 1
1.2 Aim of this thesis 2
1.3 Research questions 3
1.4 Research methods 4
1.5 Structure of the thesis 5

2 RESEARCH METHODS 6
2.1 Literature review method: a systematic mapping study 6
2.2 Simulation experiment method 9

3 NETWORK SECURITY 11
3.1 Information security concepts 11
3.2 Network security terminology 13
3.3 Network stack 15
3.4 Summary 17

4 DISTRIBUTED DENIAL-OF-SERVICE ATTACKS 18
4.1 Denial-of-service attacks 18
4.1.1 Definition 18
4.1.2 Brief history of denial-of-service 19
4.2 Botnets 20
4.2.1 Definition 20
4.2.2 History 21
4.2.3 Different types of botnets 21
4.2.4 Botnet usage 23
4.3 Distributed denial-of-service attacks 24
4.3.1 Definition 24
4.3.2 Current situation 26
4.3.3 Example attacks 28
4.4 Summary 28

5 ANOMALY DETECTION 29
5.1 Data mining and machine learning 29
5.1.1 Definitions 29
5.1.2 History of machine learning and data mining 30
5.2 Anomaly detection techniques 31
5.2.1 Definition 31
5.2.2 Anomaly detection concepts 32
5.2.3 Classification-based techniques 35
5.2.4 Nearest neighbor methods 36
5.2.5 Cluster analysis -based outlier detection 37
5.2.6 Statistical, information theory and spectral methods 38

5.2.7 Contextual and collective anomaly detection 39
5.3 Evaluating the results with ROC-graphs 40
5.4 Summary 42

6 DETECTION OF DISTRIBUTED DENIAL-OF-SERVICE ATTACKS 44
6.1 Intrusion detection systems 44
6.2 Anomaly-based detection methods 47
6.3 The systematic mapping study 49
6.3.1 Related work 49
6.3.2 The mapping study protocol & the research question 52
6.3.3 Collection process 53
6.3.4 Screening process 55
6.3.5 Evaluation of quality 60
6.3.6 Data extraction and mapping process 61
6.4 DDoS attack detection methods in encrypted network traffic 62
6.4.1 The results of the mapping study 62
6.4.2 Summaries of the included studies 69
6.4.3 Datasets and sample DDoS attacks 73
6.5 Answer to the research question 1 77
6.6 Summary 78

7 SIMULATION EXPERIMENT WITH CLUSTERING 79
7.1 Theoretical setting and implementation of the detection method 79
7.1.1 Feature selection and anomaly detection 79
7.1.2 K-means algorithm 80
7.1.3 K-means++ and K-means# 81
7.1.4 Analyzing traffic stream and detecting outliers 82
7.1.5 Implementation 82
7.2 Experimental setup 82
7.2.1 Setup of the botnet environment 83
7.2.2 Running attacks in the network 84
7.2.3 Sniffing traffic and generating the datasets 85
7.3 Results of the experiments 87
7.4 Answer to the research question 2 89
7.5 Summary 91

8 DISCUSSION 92
8.1 A validity evaluation of the systematic mapping study 92
8.1.1 Descriptive validity 92
8.1.2 Theoretical validity 93
8.1.3 Generalizability 94
8.1.4 Interpretive validity 94
8.1.5 Repeatability 95
8.1.6 Research bias and confidence in results 95
8.2 Limitations of the simulations 96

8.3 Discussion on the results of the thesis 96

9 CONCLUSION 98

BIBLIOGRAPHY 100

APPENDICES 111
A Systematic mapping study protocol 111
B Excluded studies after inclusion criteria 116

1 Introduction

The introduction briefly explains the background of distributed denial-of-service attacks and the motivation to research the field. Next, it presents the research questions, the methods and finally outlines the structure of the thesis.

1.1 Background

Denial-of-service (DoS) events have been recognized as a threat since there have been connections between computer systems (Birrell 1985) (928 F.2d 504 1991). A denial-of-service means denying or obstructing the proper access to the service and harming the availability of the service (Raghavan and Dawson 2011, 10). Later, these threats turned out to be real as they were used on purpose against organizations and businesses. The size of the bandwidth of the attacks is increasing year by year, and the types of attacks are becoming more varied (Arbor Networks 2016, 12). Current research focuses on countering various DoS attacks by developing mechanisms to prevent and detect malicious traffic. Instruments such as intrusion detection and prevention systems (IDPS) are being deployed to combat the attacks.

A particular kind of denial-of-service attack is a distributed denial-of-service (DDoS) attack, where a group of individual systems is coordinated to attack the target system at the same time, with the same goal to deny proper access to the service from its legitimate users. (Mirkovic and Reiher 2004, 1.) The aim of the DoS/DDoS attack is not to steal or compromise information. Many early attacks were flood attacks (i.e. a bandwidth saturation attack). Detection methods and countermeasures to these attacks have become widely studied and used in action. Attackers are turning to the application layer attacks that utilize botnets to send seemingly harmless packets that consume CPU cycles, memory or other resources in the target system. (Durcekova, Schwartz, and Shahmehri 2012, 1.)

Secure Socket Layer (SSL) is a protocol for negotiating encryption methods between a client and a server. It was designed to protect HTTP (Hypertext Transfer Protocol) traffic and allow sensitive information to be transported securely over the network. The TLS (Transport Layer Security) is the successor of the SSL. (Levillain et al. 2012, 11.) Since the traffic is encrypted,

a network intrusion detection system (NIDS) cannot analyze the content of the packets to determine whether it is sent by a bot or it belongs to an actual human user.

To evade the countermeasures deployed by the system administrators, attackers have increasingly turned to using encrypted connections to deliver the attack. Arbor Networks (2013, 25) announced that encrypted application layer attacks have risen since 2012. Because of the increase in incidents and the use of encryption, research in this field is needed.

There is a substantial amount of research on the detection of DoS/DDoS attacks. Zolotukhin et al. (2015) pointed out that much of the research on detection methods of DoS/DDoS attacks concentrates on the HTTP and other plaintext protocols. Thus, it is likely that the pool of methods to detect DDoS attacks in encrypted traffic is smaller and understudied. This thesis researches the state of denial-of-service attack detection methods in encrypted network traffic.

When traffic is encrypted, and the payload is unreadable, several other features can be extracted from the traffic. For instance, in the case of a "slow" distributed denial-of-service attacks on a secure network, where the disturbance in the network traffic is low and packets are sparse (Aiello et al. 2014), the number of open connections or packet arrival times can be analyzed. The papers by Zolotukhin et al. (2015) and Aiello et al. (2014) suggest methods for detecting such attacks. A method that uses clustering to analyze these kinds of metrics is presented as an example in the second part of the thesis.

1.2 Aim of this thesis

Zolotukhin et al. (2015, 275) conducted a literature review of methods for detecting DDoS attacks and noted that a majority of studies focus on detecting attacks with non-encrypted HTTP traffic. The aim of the thesis is to determine how DDoS attacks are detected in encrypted (e.g. SSL/TLS) network traffic and run experiments with a method to find out the major issues regarding the detection of DDoS attacks.

1.3 Research questions

The following research questions (RQ) are derived from the aim of the thesis. The RQ1 is necessary for the first part of the objective of the thesis. The purpose of the RQ2 is to present how one detection method works and gather results that are comparable with the current literature.

RQ1: What methods for detecting encrypted DDoS attacks are presented in the scientific literature?

RQ2: How do DDoS attack anomaly detection methods work and what are the main issues regarding their performance?

The method proposed by Zolotukhin et al. (2015) is based on the analysis of packet headers to form a baseline of the normal traffic in a network. The method uses algorithms such as K-means or DBSCAN to cluster vectors generated from the traffic. K-means is commonly mentioned as Lloyd's algorithm (Braverman et al. 2011, 2). DBSCAN is a clustering technique for spatial data, which compares the distance of a vector to its neighbors (He et al. 2013, 83). If a new traffic item does not belong to a cluster, it is flagged as an anomaly. Anomalies may or may not be DDoS attacks. This thesis builds on the work of Zolotukhin et al. (2015) by using the method presented in the paper as a basis and running experiments with an improved version of the method.

Based on the current research on detection methods in both encrypted and non-encrypted network traffic, new knowledge of the present state of methods and new research areas in this unexplored area would be welcomed. Tama and Rhee (2015) did a literature review of DDoS attack detection methods on data mining -based techniques, which did not take encryption into account. Also, their study only used automatic search from two online article databases. This thesis presents methods for detecting encrypted DDoS attacks by conducting a literature review. Based on the starting point method proposed by Zolotukhin et al. (2015), which is also based on a data mining technique, it is safe to assume many of the methods would be usable also in the case of encrypted traffic. Thus, this thesis identifies gaps in the scientific knowledge about methods that can be used in the detection of DDoS attacks in encrypted traffic.

I have a keen interest in network and information security. Research about novel detection methods of DDoS attacks and their applications are important. I chose this path to explore and develop professionally in the area of network security and especially denial-of-service attacks.

1.4 Research methods

The thesis research is divided into two parts: a secondary study and an empirical laboratory simulation of a detection method.

As a way to answer the RQ1, I conduct a light systematic mapping study using automatic search from multiple online research databases, including IEEE Explore, ACM Digital Library, Scopus, and ScienceDirect. I use the methods proposed by Petersen, Vakkalanka, and Kuzniarz (2015). The paper by Petersen, Vakkalanka, and Kuzniarz (2015) is an update to guidelines for mapping studies in software engineering field by Petersen et al. (2008). I used a Ph.D. thesis by Kaijanaho (2015) as a reference, as the author conducted a similar but more comprehensive systematic mapping study on programming language design.

A systematic mapping investigates the current state of detection methods in a systematic and reproducible way. Because of the lack of research reported by Zolotukhin et al. (2015, 275), I chose a systematic mapping study as the method because of its compliance in this case. That is when an overview of a research area is needed, and a shortage of primary studies exists (Petersen et al. 2008, 9).

Detection methods in the literature do not always state that they study detection in encrypted traffic. I decided to include few studies that could be applied to encrypted traffic by using an idea that if only packet headers are involved, the method could theoretically detect encrypted DDoS attacks as well.

I present the protocol, the inclusion criteria, and the data extraction process in Section 6.3. The results of the study are shown in Section 6.4.1. I discuss the validity of the process in Section 8.1.

1.5 Structure of the thesis

The background theory is divided into chapters 3, 4, and 5 and partially at the beginning of 6. Chapter 2 presents the research methods. Chapter 3 discusses network security and sets the terminology. Chapter 4 introduces distributed denial service attacks. Chapter 5 explains the background and theory of anomaly detection, machine learning, and data mining. Chapter 6 answers to the first research question by explaining how existing detection methods work and presenting the systematic mapping study with its results. Chapter 7 includes the simulation experiments, including the test environment, experiments, and results of detecting DDoS attacks with the method. Chapter 8 discusses the validity of the thesis and its results. Chapter 9 concludes the thesis findings and discusses the future.

2 Research methods

2.1 Literature review method: a systematic mapping study

There are three kinds of systematic secondary studies: systematic literature reviews or meta-analyses, systematic mapping studies, and tertiary studies. A primary study investigates a phenomenon that the secondary studies aim to investigate. The purpose of secondary studies is to provide a synopsis of the current research or investigate possible gaps in knowledge by examining the research itself. A tertiary study is a survey of systematic reviews, where the aim is to answer even larger areas. (Kitchenham and Charters 2007, 3.) A systematic literature reviews and mapping studies differ from a regular literature review in the fundamental way the literature is acquired and what search methods are used (Dybå, Dingsøyr, and Hanssen 2007, 228).

Systematic review studies aim to answer research questions about a particular field of research by going through the literature in a systematic way documenting the process all the time to ensure reproducibility and validity. The literature can be found by using electronic search engines, manually going through the relevant journals or looking through the references list of related articles, at all times recording how the search is done. Once they have acquired a list of related papers, a screening process for articles to be included in the study starts. The inclusion has to be done systematically and by recording all the decisions that were made during the process. Finally, by similar methods, the researchers conclude from the selected studies and form an answer to their research question based on them. (Kaijanaho 2015, 82.)

Ideally, two or more researchers do the work to avoid mistakes and remain unbiased. The whole idea is that the process is as transparent as possible to let the reader assess the study, and possibly redo the same review to come to the same conclusions. (Kaijanaho 2015, 82.)

Systematic mapping studies are meant for getting an idea of the current research in a given field of research. To get the final overview of the area, a map or a listing of the studies are collected together. (Petersen et al. 2008, 2.) The idea is not to give an answer to a specific

question about details but rather what exists in the literature, where it has been published and when. The size of the set of studies does not necessarily have to be exhaustive if it is representative of the research field. (Petersen, Vakkalanka, and Kuzniarz 2015, 1.)

The primary process of the systematic mapping study is a 5-step process, which is summarized in this list below (Petersen et al. 2008, 2):

1. Definition of RQs $\rightarrow$ Scope of the review
2. Carrying out the search $\rightarrow$ Obtained literature
3. Vetting of found research $\rightarrow$ Applicable papers for the study after evaluation
4. Keywording from the metadata of the papers $\rightarrow$ Scheme of categorized articles
5. Extraction of information and mapping $\rightarrow$ Systematic map of the literature

The outcomes of each stage are shown in the list after the arrow. The planning phase should be documented and done carefully before the actual study begins. A protocol document should be created and maintained throughout the process. In the planning phase, the scope of the study should be defined along with the used databases, manual search methods and other ways to acquire literature. The underlying research question guides the search and determines the search terms. Often a broad question has to be divided into smaller sub-questions. (Petersen, Vakkalanka, and Kuzniarz 2015, 8-9.) Kitchenham and Charters (2007, 13) suggest for individual researchers that the protocol document should be shown to a supervisor. This way, any inherent flaws can be spotted before the search starts.

The databases and starting articles should be chosen from various sources and publication venues. Dybå, Dingsøyr, and Hanssen (2007, 228) list ACM DL, Compendex, IEEE Xplore, Web of Science, Kluwer Online, ScienceDirect, SpringerLink, Wiley Inter Science Journal Finder as well suited for software engineering research. Kitchenham and Charters (2007, 17) enumerate the same sources and add Google Scholar, Inspec, and Scopus to the list. These are some of the sources online that majority of the computer science literature can be found in.

The initial collection of papers should be as large as possible if the size of the selection is unknown. The search should not be limited only to some years or researchers, but it should be restricted to known years, considering what the aims of the study are. (Petersen, Vakkalanka,

and Kuzniarz 2015, 10.) As an example, there is no point in including studies before a year when the studied phenomenon was introduced to the field. That simply adds to the number of papers to go through, i.e. noise. Every limitation of the scope and conscious decision to limit the search should be documented.

The search, including manual, electronic or automatic and snowball search, should be well documented. Meaning disclosing the full search terms, times and results of the searches in the reporting phase. (Kaijanaho 2015, 86.) Keeping track of the variables and results is crucial for the credibility of the study. Kitchenham and Charters (2007, 16) also advise asking the current researchers in the field for comments on the search terms and any gray literature they may be aware. It helps if the researchers know what kind of papers to expect, thus defining some of the papers as examples work as a validation method for the search itself.

Other metrics proposed by Chen, Ali Babar, and Zhang (2010, 2) are an overall contribution, overlap of results across sources and exclusive contribution of each source metrics. The overlap is simply the number of papers included from two or more sources. The overall contribution (OC) is simply the measure of how many studies were included from that source (I), and the percentage is simply that divided by all the included studies (A) after the exclusion criteria: $OC = I/A$. Furthermore, the exclusive contribution is the number of studies that were not found by any other source, i.e. sum of overlaps with other sources. Thus, the percentage is the ratio of articles to all the included studies A.

Sensitivity and specificity of all the sources also help to determine the validity of the study. Sensitivity can be calculated $sen. = |F \cap A|/|A|$, where A is the set of all relevant studies. Specificity may be estimated by using formula $sp. = |F \cap A|/|F|$. In both equations, $F \cap A$ is the set of found studies from the set of all studies. The size of set A is impossible to know without comprehensive knowledge of the research field, but it can be estimated. (Kaijanaho 2015, 87.)

Snowball or backward searching means that the researchers take the reference lists of the studies that they know should be included in the study and see if more papers should be included. Furthermore, these studies are then evaluated in a similar manner to get a list of

publications going backward in the references of each paper. (Kaijanaho 2015, 88.)

More than one person should make the selection of papers as well as extraction of the methods and details from the papers or at least checked by someone else (Petersen, Vakkalanka, and Kuzniarz 2015, 4). This way the mistakes in evaluating the content in unclear cases and be minimized and the synthesis of the mapping study becomes more reliable. However, in a case a single researcher is working alone, a random retest of a sample or discussing the decisions with a supervisor are enough to ensure some degree of credibility in the findings (Kitchenham and Charters 2007, 20).

The actual thematic map and the synthesis of the findings can be done in many ways. Petersen et al. (2008) suggest that for mapping studies the number of publications per year at least in a bar chart. They continue, however, that bubble plot with more aspects than simply the year of publication is more interesting. They encourage the researchers to explore how to best represent the data and the included trends in the data.

Reporting phase, according to Kitchenham and Charters (2007), should be done in both journal or conference proceeding and a thesis or technical report because of the article length limitations of many journals. In the reporting phase, the authors are advised to evaluate the validity of their study. During reporting, several validity measures should be taken into consideration. According to Petersen, Vakkalanka, and Kuzniarz (2015) the author of a mapping study should discuss at least: (1.) the validity of the description of the findings, (2.) theoretical validity of the mapping method, (3.) generalizability of the results, (4.) validity of the explanations and the synthesis of the extracted data, and (5.) repeatability of the study.

2.2 Simulation experiment method

The second part of the research of this thesis, to answer the RQ2, is done by applying a simulation experiment method. A simulation is a controlled experimentation method in which a hypothesis is tested against artificial data. The main fault of this approach is that the experiments might not apply to the real world. However, the experiments can be done in a controlled and safe setting. (Zelkowitz and Wallace 1998, 24-25.) Jarvinen (2000) categorizes the method as empirical studies' theory testing approach.

In the simulation method, a model of the actual situation or scenario is constructed to gather data. Depending on the accuracy of the model, the researchers can hypothesize how the method works in reality. The simulation method is cheaper than running experiments in a production environment, and it is used especially for new methods that are presented. (Zelkowitz and Wallace 1998, 28.)

In this thesis, the simulation is conducted by setting up a virtual network of Linux machines, where normal and malicious network data are simulated. The network works as a test bed for various scenarios of denial-of-service attacks. The setup consists of a small botnet and a webserver. Both the legitimate and the malicious traffic use SSL/TLS encryption to communicate with the webserver. The attacking bots IP (Internet Protocol) addresses are known in this simulation, thus creating an identified set of malicious connections. The detection method evaluates the dataset to detect malicious traffic. The thesis presents a theoretical background of a method, the simulation environment and the results in detail in Chapter 7.

3 Network security

This chapter describes the background terminology of network and information security. DDoS attacks are a threat to network security. The chapter also explains network stacks, which are related to the types of DDoS attacks.

3.1 Information security concepts

In an organizational context, different security has to be implemented in many layers. These are physical, personnel, operations, communications, network and information security. Physical security refers to the implementations of physical access control barriers. Personnel should be guarded against physical or digital harm. Operations of the organization are to be kept safe from outsiders to combat espionage. Communication and networks should be kept secure to transfer information securely. Information security means the safekeeping of information resources in all stages. Network security is, therefore, part of information security, as it involves the guarding the safe transfer of information over networks. Information security can be described as a combination of policies, network, computer and data security as well as management of information security. (Whitman and Mattord 2011, 8-9.)

Common information security goals are availability, integrity and confidentiality, and information security can be seen as the conservation of these (ISO/IEC 27000 2016). These terms have been identified by Saltzer and Schroeder (1975) and for many years being used as a basis for understanding information security. They are referred as a so-called AIC triad, a CIA triad (Cherdantseva and Hilton 2013, 547) or an ACI triad (Tirthani and R 2013, 1). The concept is the same, regardless of the order of the goals. Problems in any one of the corners of the triangle reduces security as a whole in the case of a secret document for example. A user of such a document should be able to access the information, be sure that it has not been altered and confirm that no one without proper access rights can view the it. A public document should be accessible by the public. The confidentiality, however, is less important than the availability and the integrity of a public document. (von Solms and van Niekerk 2013, 3).

Depending on the definition, the list may also include accuracy, accountability, authentication, authenticity, non-repudiation, possession, reliability, and utility (Whitman and Mattord 2011, 12) (ISO/IEC 27000 2016) (Cherdantseva and Hilton 2013, 548). Whitman and Mattord (2011) point out that the AIC model is does not address the information security in a satisfactory level in today's fast paced world. However, in the context of this thesis, the model offers a way to discuss the effects of DDoS attacks. Information security consists of various concepts related to securing the use of information. These include encryption and authentication to ensure the confidentiality, calculating hashes to ensure the integrity and implementing fault tolerant data storage or intrusion detection and prevention systems to ensure the availability (Tirthani and R 2013, 1).

Attacks against information systems target various aspects of information security at different stages in the lifetime of information. Information can be transported, persisted onto a data storage, being created, being handled by an operator or being erased completely (Cherdantseva and Hilton 2013, 550). Information can be seen as any information, whether it is written, printed, bits on a magnetic disc, being sent by an electronic or physical ways or a spoken word. (ISO/IEC 27002 2013). DDoS attacks target the transfer of information from the server to the client within a network, thus damaging the availability of the information. Motives denying access to certain information resources may vary from monetary gains to an urge to show-off.

Cybersecurity, according to the ISO/IEC 27032 (2012) standard, is a synonym for information security. von Solms and van Niekerk (2013, 2) argue that cybersecurity, although used interchangeably with information security, does not equate with information security. They conclude that a cyberattack may hurt individuals or societies, contrary to most information security threats whose only secondary effect can cause injury to the victim. (von Solms and van Niekerk 2013, 2). Few examples of cyberattacks that do not cause unavailability integrity problems or confidentiality issues are cyberbullying, home automation attacks, illegal sharing of digital media and cyberterrorism (von Solms and van Niekerk 2013, 3). ISO/IEC 27032 (2012) standard defines another term called cybersafety, which includes psychological effects of bullying, physical effects of home automation attacks, financial harm caused by sharing of intellectual property and political aspects of cyberterrorism, in conjunction

with many more consequences of attacks. Although the definitions may vary, they help us to understand complex motives of DDoS attacks and the cyberspace in which they happen.

3.2 Network security terminology

Network security, such as physical security, is about making calculated risks based on threats and vulnerability. One can never be completely safe on the Internet nor at home. (Krawetz 2007, 3.) Such as a crowbar can be used for both good and bad, a network analysis tools can be utilized for debugging for mistakes in configuration, or for comprising identities of others (Krawetz 2007, 31). Krawetz (2007) continues that ethics have a lot to do with the security education and tools available nowadays. Depending on the purpose of the action, sniffing other peoples' network traffic with a tool can be considered legal, illegal, ethical or unethical. The tool is the same, regardless of the use. This section defines the terms most commonly used in the Internet security literature and research papers.

According to Schneider (1999) and Krawetz (2007) terms in Internet security research are: a vulnerability, a threat, an attack, an attacker, an exploit, a target, an attack vector, a defender, a compromise, a risk. Following paragraphs explain the terms shortly.

A vulnerability translates to a flaw in some of the aspects of an application or organization. These aspects can be design, code, servicing or general management (Krawetz 2007, 4). For instance, passwords may be stored in the database without encryption or the encryption method is so old that it has been known for years to be vulnerable. The latter is a case of poor management and maintenance. There is no system which could be immune to any attack, but in normal conditions, all the vulnerabilities should be mitigated by knowing the threats associated with that vulnerability.

A threat is someone or something which has the ability and a reason to use a vulnerability. Identified threats may be attackers or events that might lead to an adverse outcome for the system. (Krawetz 2007, 4.) To illustrate, a natural disaster might cause loss of data or even breach of the physical security of the servers. A threat from the inside of the company might be an uneducated employee. To give an example of a physical threat, such a threat to security are rodents that cause up to 25% of failures cell and energy networks (Krawetz 2007, 106).

An attack is an act of taking advantage of the vulnerability, and an **attacker** (i.e. threat) is a something or someone who starts the attack (Krawetz 2007, 5). Whether the company has taken preventative measures and the vulnerability and the threat associated with that vulnerability have been identified, an attack is taking advantage and the threat has been realized. An attack starts when the attacker has identified the vulnerability, chosen the tools and acts.

An exploit is a tool with what the vulnerability can be attacked. There may be several exploits to a single vulnerability (Krawetz 2007, 5). An exploit can be simply a program designed to open a backdoor to a system, and deploying the exploit successfully to the target system can be considered as the start of the attack. **A target** means the individual, corporate actor or an application who suffer from the exploit being attacked. There may be prime and consequential targets, depending on the exploit. **A compromise** happens when an exploit has been used effectively on a target. (Krawetz 2007, 5.)

An attack vector refers to the approach taken by the attacker including the exploits and ways or procedures to reach the target (Krawetz 2007, 5). Many companies require a high level of security with passwords (i.e. requiring people to use lower case, upper case, numeric and special characters), making them difficult to remember. Therefore, many people write their passwords on a piece of paper, exposing an alternative attack vector to acquire a password to the system (Krawetz 2007, 74).

A defender is an actor who tries to lessen the effects or inhibit an attack in the first place (Krawetz 2007, 5). A security professional at the IT department might have installed various methods for detecting attacks and preventing large scale compromise of corporate data. These measures might include implementing intrusion detection systems (IDS), and intrusion prevention systems (IPS) and other measures such as secure protocols (e.g. Knock-knock protocols, SSH or SSL/TLS) to create a defense-in-depth system to combat the weaknesses of any security layer. (Krawetz 2007, 498-500.) The level of security measures is usually defined concerning usability, convenience, threat and available resources in the form of risk analysis.

A risk is an evaluation of the probability that an attacker can go around the defender utilizing

an exploit to attack the target with compromising it (Krawetz 2007, 5). Krawetz (2007) presents one way of determining the risk for a given vulnerability, which uses score between one and three (one being the lowest) to five different metrics. These variables include ease of exploitation (E), scope of affected systems (S), impact in case of an attack (I), future ramifications if left untreated (F) and actions taken to mitigate already (M). The M value should be assigned a value of zero if countermeasures have been taken. The risk factor can be calculated be summing E, S, I and F, and then reducing the M value from the sum. (Krawetz 2007, 518.)

3.3 Network stack

When talking about networks and communication, it is convenient to divide the actions that are needed in the communication between systems into groups. These groups of specified problems along the way to a successful communication to happen are easier to manage. These groups are called layers and the model that combines protocols that work together is known as a network stack or a network suite. (Blank 2006, 2,18.) The layers may be seen as concepts, and the stack is a representation of the layers. In principle, the layers are independent and switching the protocols in a layer does not affect the other layers and their protocols. For instance, two routers from two different vendors can easily talk to each other since their network stacks have the same protocols, even if they have no common hardware or software. (Krawetz 2007, 51.)

The purpose of each layer is to offer services to the upper layer and protect it from what takes place underneath the layer. The layers near the top are not required to be aware of where the data came from and how it reached the top layers. (Blank 2006, 19.) Because of the modularity and independence of layers, many of the protocols implement their own error correction mechanisms. This might seem unnecessary, but this increases the reliability of the whole stack, since one layer only needs to take care of errors that might occur on that layer, and a possibly malicious connection needs to pass through several layers of checks before reaching the target. Advanced protocols take advantage of encryption where as lower lever protocols simply use checksums to approve packets. Some errors are intentional. The data may be modified on purpose to cause loss of data or connectivity, resulting in denial-

of-service. (Krawetz 2007, 52-53.) Usually, it is only one protocol that has to deal with the attack, as the packets seem normal to the others.

Several network stacks exist, but a TCP/IP (Transmission Control Protocol over Internet Protocol) stack is the most common as it is the standard for the Internet. OSI (Open Standards Interconnection) reference model is most widespread network stack comparison model in literature. The TCP/IP has been developed by the US Department of Defense (DoD) and the OSI reference model by the International Standards Organization (ISO). (Blank 2006, 18.) Each of the stacks describes functions, standards, protocols and agreements of their layers. Other stacks include e.g. SNA and SNA/IP developed by IBM, DOCSIS developed by Cable Television Laboratories or Network Control Protocol (NCP) developed by DoD before the TCP/IP (Ferguson, Clouston, and Talerico 2003, 2) (Fellows and Jones 2001, 202) (Blank 2006, 4). In many stacks, the communication may also be described as stacks in succession or nested within each other. (Krawetz 2007, 51.)

The TCP/IP stack part are named Network interface, Internet layer, Transport layer and Application layer. For reference, the seven layers of the OSI model are Application, Presentation, Session, Transport, Network, Data-link and Physical layers. Application layer contains the protocols that are used to communicate from one application to another. The Presentation layer makes sure that the syntax of the message is understandable by the recipient and possibly adds encryption to the message. The Session layer administers sessions during multiple consecutive connections. Transport layer takes care of starting, preserving and ending the connections as well as keeps track of all the packets received and sent. The Network layer takes care of the routing of packets and sending them to the correct logical address. The Data-link layer finally prepares the packages or frames of ones and zeroes from the packets from above and sends them to the physical medium. The Physical layer is the network and the movement of bits across the cable as pulses of electricity or through the air as radio waves. (Blank 2006, 19-24.)

This thesis focuses on attacks which target the application layer, e.g. a slow HTTP GET DDoS attack. Therefore, knowledge of the different layers and the protocols that are related to which layer is crucial to talk about the effects and classification of various malicious traffic. The layers of the OSI model and the TCP/IP stack are shown in Table 1. Arbor Networks has

Table 1. Comparison of the OSI and the TCP/IP models (Blank 2006, 24.)

The OSI model	The TCP/IP model
Application layer Presentation layer Session layer	Application layer
Transport layer	Transport layer
Network layer	Internet layer
Data-link layer Physical layer	Network interface layer

published statistics about denial-of-service attacks for more than a decade, and the majority of attacks are still targeted to the transport layer of the OSI model (Arbor Networks 2016). OSI reference model is commonly used in DoS literature to classify attacks. The TCP/IP stack is used as an example because of its commonality in the Internet-based communication and many attack vectors target features of the TCP/IP stack protocols.

3.4 Summary

This chapter presented the context of network security, including the terms, concepts, and insights. Then it explained the network stack. These security terms and definitions are used in the DDoS literature. Next chapter describes denial-of-service attacks and the current state from a network security stand point.

4 Distributed denial-of-service attacks

The purpose of this chapter is to familiarize the reader with the definition, history and types of denial-of-service attacks. This chapter explains the relation between denial-of-service and distributed denial-of-service attacks and shortly discusses the enabling technology for DDoS, botnets.

4.1 Denial-of-service attacks

4.1.1 Definition

A normal user uses services from the Internet constantly and even a short loss of connection to a service may have tremendous effects if it happens in the right moment. This disruption may be an accident of one or more natural causes. However, when a service is unreachable to the real users because of a intentional attack against the availability of the service, denial-of-service or DoS attack is taking place. (Raghavan and Dawson 2011, 1.) How the regular user usually sees this type of attack is a problem in the connection to their favorite service or website. In this thesis, attacks to the availability of the service caused by an attack through the network is a sufficient definition to denial-of-service.

One of the proposals of the Finnish government states: *"Denial-of-service attack means intentional complete denial or limiting of the operation of the target system such as an email server* (”HE 153/2006” 2006, my translation)"

United States Computer Emergency Readiness Team (US-CERT) defines: *"In a denial-of-service (DoS) attack, an attacker attempts to prevent legitimate users from accessing information or services* (US-CERT 2013)."

Many other organizations such as the International Telecommunications Union (ITU-T) and the Committee on National Security Systems (CNSS) have also defined denial-of-service attacks, and the definitions follow the same pattern. The main idea is that there are legitimate authorized users who are unable to access a service under attack and complete a task promptly because deliberate actions taken by the attacker. Multiple attack vectors exist tar-

geting the bandwidth, the CPU, various packet buffers, features of protocols or business logic to render the service incapable of continuing regular service. (Raghavan and Dawson 2011, 10.)

Motives to carry out a denial-of-service attack vary from personal to political causes or from reputation for a successful attack to financial reasons. Sometimes the main target of the attack may be the user of the service being attacked, not the service itself. (Mirkovic and Reiher 2004, 41.)

Three categories of denial-of-service attacks exist based on the attack vector: volumetric (or bandwidth saturation), protocol related or application layer attacks. Volumetric attack consumes the physical capacity of the network to deliver packages, thus preventing the legitimate users' connections. Protocols are susceptible to attacks that exploit the features of the protocol such as connection tables or wait timeouts when the connection is open. When the underlying network infrastructure and protocols are working well, the final attack vector lies in the applications such as the webserver or the website itself. These attacks exploit the vulnerabilities that cause the application to run out of resources such as CPU cycles or RAM. (Petiz et al. 2014, 1.)

4.1.2 Brief history of denial-of-service

Even during the times of the ARPANET, the issues in the protocols were acknowledged and the likelihood of a denial-of-service attack was raised. Namely, the RFC706 (1975) points out a potential problem with the Host/IMP interface protocol. A mere 10 years later Birrell (1985) mentions the risk of intentional denial-of-service in his paper about a secure communication protocol.

As the ARPANET was used mainly by professionals, there were only accidental denial-of-service attacks in the network. In late 1988, one such incident brought the ARPANET to its knees. In November 2nd, 1988, a Ph.D. candidate at Cornell University, Robert Morris developed a computer worm whose purpose was to show how poor the security of the network was. The worm ended up crashing several computers across the country and jamming the whole network. Costs of cleaning the worm were from $200 up to $50,000 depending on the

affected machine. Morris was convicted of computer fraud. (928 F.2d 504 1991)

Lin and Tseng (2004) say the first event happened in the summer of 1998 and Garber (2000) tells a story of the first DDoS attack in the US which took place in August of 1999 against the University of Minnesota. The first DDoS attack happened around that time.

By the change of the millennium, attacks had gotten more sophisticated, and botnets were starting to appear in the attacks, making them distributed denial-of-service attacks rather than works of individual agents. Many companies such as Amazon or Yahoo had experienced a distributed denial-of-service attack. (Garber 2000, 12.) (Raghavan and Dawson 2011, 10.)

4.2 Botnets

4.2.1 Definition

Botnet is a network of malicious software (malware) that have taken over the host machine and execute the orders that so-called botmasters send them. Most of the time they are used to carry out fraudulent actions that benefit the botmasters' aims (Silva et al. 2013, 380). There are other kinds of malware, and the most predominant distinction them and bots is the so-called C&C (short for Command & Control) channel to the malware (Shanthi and Seenivasan 2015, 1). If the bot does not receive commands, it simply sits still and keeps out of way of the legitimate user of the machine.

Botnets are increasing in size and numbers. They have become a major problem on the Internet, and it has estimated that the number of bots or zombie machines connected to the Internet is already 15-25% of all devices, of which very few people know that the device they own is a member of a botnet (AsSadhan et al. 2009, 156). The malware tries to hide from anti-virus software and the user. For example, Srizbi trojan bot halted its transmission if the person sitting at the computer touched any controls, to minimize the effects of the spamming to the compromised machine (Stern 2009). If the bot does not send large amounts of data, the ISP might also not suspect anything.

4.2.2 History

First botnets used IRC (Internet Relay Chat, an instant messaging service) channels as C&C channels. The purpose of the first botnets was to have an overview and manage the chats (i.e. channels) and private messages. Some of the services the botnets offered were commands to administer the channels, amuse the users with text-based games and query meta data about the system, usernames or email addresses. (Silva et al. 2013, 380.)

The earliest identified bot was Eggdrop 1993, and the next bots were based on Eggdrop, but their purpose was to attack other users. At that point they got functions such as a DDoS attack. The next generation bot software had more sophisticated means of communication, stay undetected in the host and launch more state of the art attacks. AgoBot is seen as the change when the botnets changed to a more dangerous type from a mere helper network. (Silva et al. 2013, 380.)

Nowadays bot malware is distributed via email, downloaded files, torrents or websites infected with malicious JavaScript code. Communication methods have become more sophisticated as well, and they are using HTTPS and P2P (Peer to peer), although IRC is still used today. (Silva et al. 2013, 381.)

4.2.3 Different types of botnets

The topology of a botnet can be centralized or distributed. That means the commands to execute a task are either sent from a centralized C&C server or bots distribute them from one to another. In both cases, the communication tries to stay hidden in normal requests and blend in with regular traffic. Figure 1 depicts a centralized botnet and Figure 2 a decentralized botnet. In a decentralized version, the botmaster does not have a full control of the message distribution to the bots. In general, there are five type of botnets based on their communication channel and platform: IRC-, HTTP-, P2P-, cloud- and mobile-based botnets. (Shanthi and Seenivasan 2015, 1-2.)

IRC-botnets still use IRC channels or protocols to communicate with their botmasters. The botmaster can send a command to the bots in the same channel the bots are, which is so-called push-mechanism. HTTP-bots use and common-looking HTTP HTML -requests to

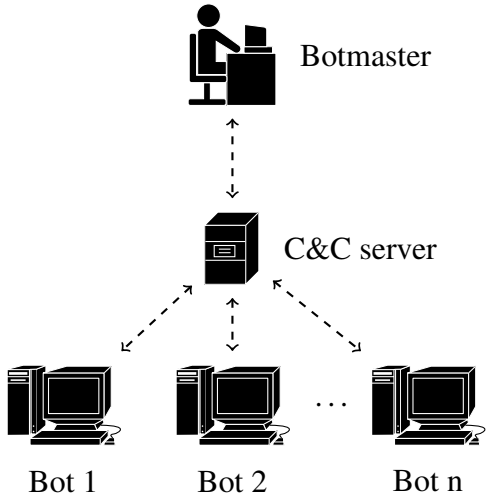

Figure 1. Centralized botnet

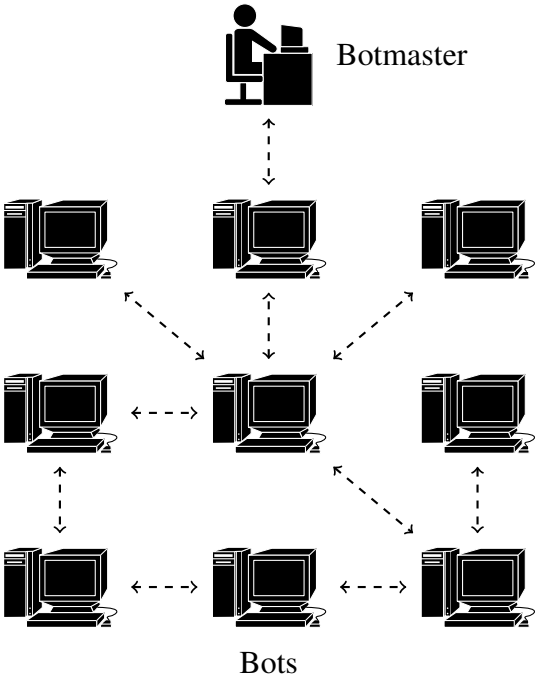

Figure 2. Decentralized botnet

receive commands. HTTP traffic is difficult to detect and recognize from regular traffic as they usually only visit the C&C server periodically to pull new instructions. HTTP-bots are mostly used for email spamming. (Shanthi and Seenivasan 2015, 2.)

P2P botnet does not have a hierarchy, but all the bots are both C&C server as well as bots. Anyone of them can get a new instruction, which it then passes on to the bots that it is aware. The distributed system does not allow the botmaster to control each of the bots at the same time, but it also offers an added level of obscurity to the security. (Shanthi and Seenivasan 2015, 2.)

Cloud botnets reside in a cloud that the attacker has acquired for another purpose. The advantages of cloud bots are the easy and quick setup of a bot network of virtual machines, all of them are available at all times and many cloud providers don't have ways to detect bots in their clouds. (Shanthi and Seenivasan 2015, 2.)

Mobile botnets exploit the Bluetooth and SMS services in smart phones to communicate. They are used for accumulating data from the users' devices rather than send spam or perform attacks. (Shanthi and Seenivasan 2015, 2.)

IoT (Internet of Things) devices are increasing popularity, and they are very popular among criminals as they are constantly online, have default passwords and do not run any anti-virus software (Pa et al. 2015, 1).

4.2.4 Botnet usage

When a black hat has managed to gather enough agents for a botnet, he can put it for sale on the black market for anyone to buy and use for their purposes (Shanthi and Seenivasan 2015, 1).

Many botnets transmit spam email as their first function, while others are used for denial-of-service attacks, click fraud, malicious banking operations, and work as remote proxy servers for other purposes (Dupont et al. 2016, 134). To illustrate, Srizbi botnet was liable for the majority of spam on the Internet sent between June of 2007 and February of 2009, reaching its highest of 60% in 2008 multiple times (Stern 2009). In a more general note, it has been

estimated that about 80% of email traffic is spam and while it gets caught in the spam filters, the traffic still uses network infrastructure (Silva et al. 2013, 378.).

An anonymous researcher conducted an Internet census to map the available IPv4 address space by scanning all the ranges with a botnet. He acquired by logging into servers using telnet credentials both left to the original setting, which made the job easier. He found that there were more than a million devices with this default set up and this is an issue worldwide. He installed a small bot program on about 400 thousand devices creating a massive botnet that would then scan the ports. It only took one day to scan the whole allocated 3.6 Billion address space. He concluded that approximately 1.3 Billion IPv4 addresses are being utilized. He also discovered that a botnet called Aidra was using the same method by checking that the temp-folder of these devices contained traces of files dedicated to e.g. SYN flood. (Botnet Carna 2013)

What he did was illegal, but this also begs the question about the security of a vast number of devices on the Internet. He took the easiest route to compromising the devices, and the black hats have also acknowledged this way.

4.3 Distributed denial-of-service attacks

4.3.1 Definition

A distributed denial-of-service (DDoS) attack is an adaptation of the broader denial-of-service term. A DDoS attack is characterized by many agents that are coordinated to attack the target system through the network. The agents are usually part of a botnet (see Figure 3, although there have been successful DDoS attacks performed by humans accessing a website in a coordinated manner. Bandwidth DDoS attacks are classified into human coordinated bandwidth attacks (e.g. F5 -key flood, LOIC-tool or a flash crowd event), automated or semi-automated bandwidth attacks (amplification e.g. Smurf or fraggle attack, reflection attacks or botnet-based attacks).

The 4chan-originated hacker group Anonymous has been responsible for orchestrating a DDoS attack against anti-Wikileaks companies, such as financial organizations and DNS-

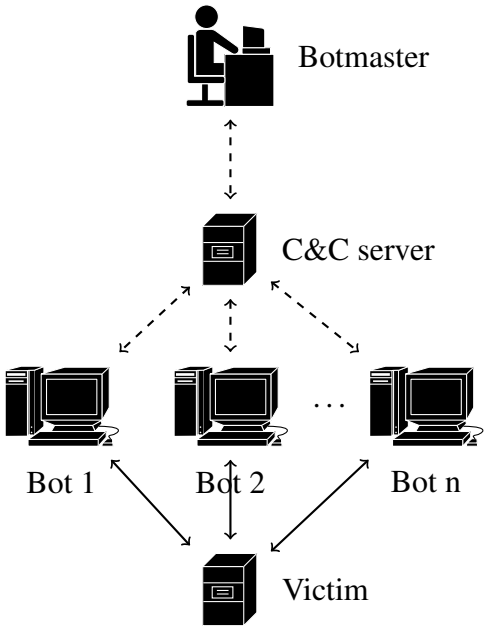

Figure 3. A distributed denial-of-service attack using a botnet

providers. In order to conduct this attack, they used a program called Low Orbit Ion Cannon. They asked people to download the application, choose a target and start the attack. A group of people joined a voluntary botnet to launch a collective DoS. (Mansfield-Devine 2011, 5.) This incident is related to another type of event, a flash crowd event, when a DoS happens involuntarily.

The Slashdot effect or the flash crowd event got its name from a science article site called Slashdot, which featured links to websites with inadequate capacity to handle high volume of surfers. The large crowd of readers of Slashdot often accessed the link simultaneously rendering the site unable to respond to new requests. The readers did not want to crash the site, but essentially they consumed the server completely. (Raghavan and Dawson 2011, 13.)

A distributed denial-of-service attack is performed in multiple steps. The first step is to acquire a botnet or agents that can be commanded by the attacker. Botnets can be bought on the black market for money or the attacker can build a botnet from scratch. In the next phase, the attacker accesses the C&C (Command & Control) server of the botnet and sends a command to the agents to start sending a certain type of data to the target machine. Depending on the purpose of the attack, a time window is chosen carefully, especially if the attacker is paying

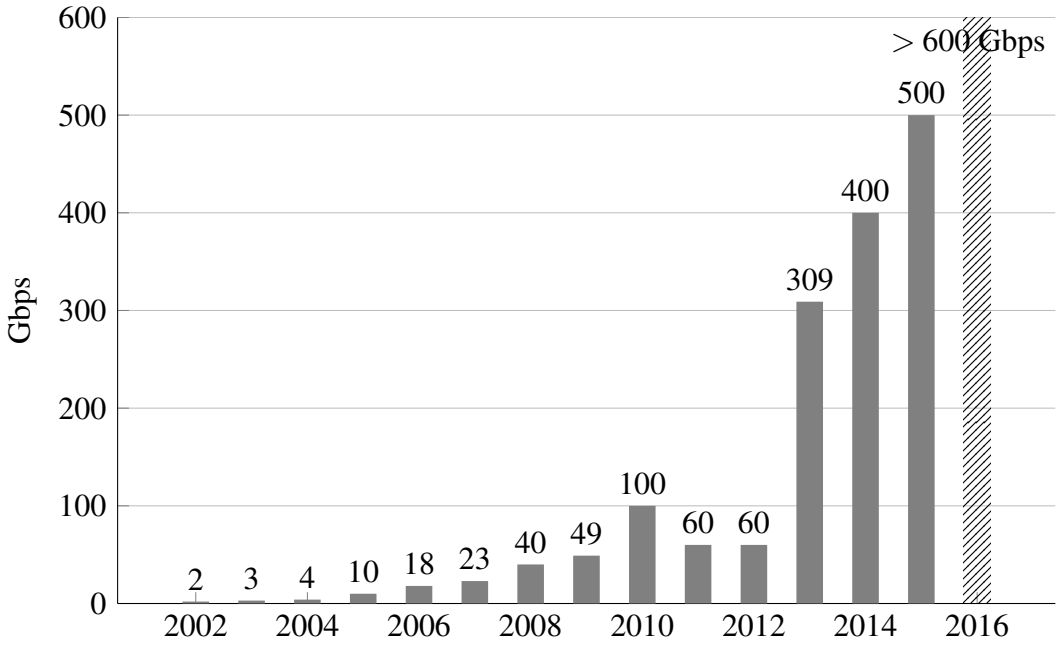

Figure 4. Bandwidth of the volumetric attacks reported yearly since 2002 (Arbor Networks 2011, 15.) (Arbor Networks 2016, 24.) (Krebs 2016)

an hourly fee for the usage of the botnet. IP spoofing techniques are used to cover the tracks to the agents and especially to the C&C server. After the goal of the attack has been reached, the agents stop sending the packets and the service returns to normal operation. (Mirkovic and Reiher 2004, 40.)

4.3.2 Current situation

In the year 2000 and onward, the number of attacks kept on rising and by the year 2010, the magnitude of volumetric attacks had reached 10 Gbps as the new norm. The biggest attacks reached 100 Gbps Arbor Networks (2011, 5) reported in their yearly security report. A new wave of Application layer (also known as Layer 7 attacks) started to emerge and more complicated multivector attacks where volumetric attack has Application layer attack vectors included. Main motives for a large scale attacks were ideology/political, gaming related or vandalism in 2012. (Arbor Networks 2011, 5.)

According to the report Arbor Networks (2011, 16), much of the reported bandwidth within the examination period from November 2010 to October 2011 was caused by the WikiLeaks

incident. In the Arbor Networks (2013, 25) report, attacks against encrypted services rose to 54% of all the answers to the survey.

In the year 2013, the largest bandwidth attack had risen to 309 Gbps and 2014 the same figure was already 400 Gbps according to Arbor Networks (2015), and in 2015 the largest attack was already 500 Gbps as stated by Arbor Networks (2016). Krebs (2016) wrote that the bandwidth attacks are already in the range of 600 Gbps and more for the year 2016. According to the 2016 report majority of the attacks (74%) are still less than 500 Mbps in size (Arbor Networks 2016). Still, this is a significant amount against targets without a sufficient DDoS mitigation infrastructure. See Figure 4 to get an idea of how fast magnitude of attacks has risen.

In 2014, volumetric attacks are the most prevalent attack method, at 2/3 of the attacks consuming bandwidth as their primary objective. Amplification or reflection attacks have been the leading cause of this surge in volume of traffic. (Arbor Networks 2015, 34-40.) While HTTP and DNS remain the most targeted protocols, HTTPS rose to about 50% of respondents reporting attacks in HTTPS services. (Arbor Networks 2015, 41.)

Mobile networks are seeing an increase in DDoS attacks as well. However, the difficulty of detecting a single source of traffic in a mobile network remains to be an issue. With the rise of LTE technologies and a NAT sitting between the possible targets and the source. (Arbor Networks 2015, 81.)

One average, a DDoS attack lasts less 30 minutes and costs a target organization $1.5 million (Jaffee 2016). The average loss for an Internet-based business is so high, that investing in a DDoS mitigation system pays off already if the company can fight off the most pathetic attempts.

Defense methods against DDoS attacks can be divided into three types: preventative, reactive and source-tracking (Nagaratna, Prasad, and Kumar 2009, 753). Many companies are offering their services to fight off the DDoS attacks with several detection and prevention methods that they rarely fully disclose in fear of competitors and attackers gaining an advantage.

4.3.3 Example attacks

A TCP SYN flood attack exploits the TCP three-way handshake. During the attack, the attackers sends a SYN packet, the server responds with a SYN-ACK packet and puts the connection to a half-open state table. The final ACK packet never comes, and the connection is left half-open until timeout. (Linge, Hope, et al. 2007, 55.)

A slow denial-of-service attack takes an advantage of how certain application layer protocols have been implemented and sends malformed or normal packets at a slow pace and low bandwidth. The connection tables of the server program fill up blocking any potential new connections. Examples of these attacks are Slowloris, SlowReq, SlowRead and Slow Next attacks. Slowloris opens the connection and sends an HTTP-request very slowly, but never completing it. The server is simply receiving packets and waiting for the end of the request to be able to fulfill it. SlowReq also opens connection to the HTTP-server, but sends tiny packets which do not comply with the protocol, e.g. one space. In the SlowRead attack, the responses to normal HTTP-requests are processed at a sluggish pace forcing the server to keep the connection open. Slow Next uses the Wait Timeout function of the connection after each normal request to occupy the connection table. (Aiello et al. 2014, 1-2.)

4.4 Summary

Volumetric distributed denial-of-service attacks are more powerful since more than one host executes them. Most of the services nowadays can combat simple denial-of- service attacks with ease. However, with the emergence of application layer attacks, where a single attacker can disable a server farm with simple incomplete HTTP-packets, the scheme has changed. When these stealthy attacks are coupled with a small undetectable botnet, the damage is still happening and needs more research. (Sourav and Mishra 2012, 749.)

So far the thesis has discussed network security and DDoS attacks. The next chapter introduces anomaly detection to get closer to the detection of DDoS attacks.

5 Anomaly detection

Anomaly detection chapter introduces the underlying mechanisms of machine learning, data mining and anomaly detection that are crucial for the methods presented and studied in this thesis. The chapter starts by defining data mining and machine learning and then explaining the context in which anomaly detection methods are based. It then introduces most common anomaly detection methods. Lastly, the chapter covers receiver operating characteristics (ROC) graphs as a means of comparing results of different classification algorithms.

5.1 Data mining and machine learning

5.1.1 Definitions

Data mining and machine learning stem from the study statistics. Data mining means analyzing existing data to find solutions to questions that could be answered from the data at hand (Fürnkranz, Gamberger, and Lavrač 2012, 2). Data mining can also be mentioned as knowledge discovery from data (KDD) (Han, Pei, and Kamber 2011, 1). Data mining aims to make sense of the data with the help of the human, whereas machine learning approaches seek to minimize the human factor and learn by changing the underlying decisions when presented with new information.

There are few differences between statistics and data mining. Data mining methods have been designed to make use of big data collections where statistical approaches start to be inefficient. The language of representation in data mining methods is usually more human oriented than in statistics. Lastly, statistical processes try to prove a hypothesis from a clear set of data, whereas data mining concentrates in creating hypotheses from often unstructured or unknown data. (Fürnkranz, Gamberger, and Lavrač 2012, 3.)

As an example, Google's flu trends have been aggregated from search strings to the popular search engine. They can predict the looming flu season with incredible accuracy, by looking at the search patterns of people. Seasonal patterns of influenza epidemics can be seen from the data and the differences between countries. (Han, Pei, and Kamber 2011, 2.)

Learning methods can be supervised, semi-supervised or unsupervised. An entirely supervised method, a maximum likelihood (ML) classifier, needs to know the classes in advance and how to classify the entries correctly (Jeon and Landgrebe 1999, 1079). When the classes are not defined, usually an unsupervised method such as clustering is used (Grira, Crucianu, and Boujemaa 2004, 1.)

Classification of records in the database or a dataset is the basis for many data mining approaches. A classifier is simply an algorithm or a function that categorizes the records into classes using the features from the records. Classifiers come in many forms, such as neural networks, decision trees or Bayesian networks. (Friedman, Geiger, and Goldszmidt 1997, 131.)

Bayesian classifier, a naïve Bayesian, is taught the dependent predictors for all features x_i for a certain record, i.e. what is the probability that a feature x_1 is set when it is known to belong to a record that is class B. (Friedman, Geiger, and Goldszmidt 1997, 131-132.) If the training data feature values have gaps, the predictions could turn out inaccurate. Methods such as Dirichlet prior can be used to even the probabilities. (Viaene, Derrig, and Dedene 2002, 204.) Classifying the next record into a class is done with calculating the likelihood with Bayes rule that the record belongs to a class based on the features it has. The class which has a big *posterior probability* marks the class which the record belongs to. (Friedman, Geiger, and Goldszmidt 1997, 131-132.)

5.1.2 History of machine learning and data mining

Already when computing was fairly new in the 50's, researchers wanted to know if the machine could acquire knowledge such as humans do and gradually get better at doing tasks. For now, algorithms have proven their power in specialized areas of research and with specific problems. Data mining and machine learning field continues to evolve, but not yet at the level of humans. (Mitchell 1997, 1.)

In the beginning, 70's and 80's, machine learning methods included perceptrons, decision trees, and rule learning. Data mining got its start in workshops at the beginning of 90's. There was an interest to create value for the relational database business. The developers

of large database engines wanted to enrich the capabilities of data management and analysis of their systems. At the introduction of data warehousing, there was a need for advanced analysis of the data that the current SQL could not handle. (Fürnkranz, Gamberger, and Lavrač 2012, 3.)

Why data mining techniques started to develop, was the ever-growing amount of data and the realization that the data can be used to generate more money to the business, improve processes, gain new information on the customers as well as scientific value. The surge in data and the complexity of it made it unrealistic to go through by humans (Han, Pei, and Kamber 2011, 5). So-called (OLAP) online analytical processing came about as a tool to gain an understanding of unspecified questions about the data stored. The concept of data mining is born to the understanding that not all the questions can be known in advance. The tools such as the OLAP and the SQL still have a structure and the person asking the question has to know something about the data. (Fürnkranz, Gamberger, and Lavrač 2012, 3.) OLAP included ways to summarize, aggregate and look at the information from different perspectives (Han, Pei, and Kamber 2011, 4)

There was a need for work that can understand unknown data. Thus, methods to deal with records to make a rational analysis of the data were developed. The question is not often well defined in data mining applications. (Fürnkranz, Gamberger, and Lavrač 2012, 3.) Anomaly detection uses statistics, machine learning methods, and data mining.

5.2 Anomaly detection techniques

5.2.1 Definition

Anomalies are samples of records that do not follow the general conduct or ways of the entire data as a whole. Anomaly detection is the act of searching for these by exposing patterns. Other common names for anomalies are outliers, exceptions or contaminants depending on the field of study and the use of the anomaly detection method. (Chandola, Banerjee, and Kumar 2009, 1.)

Anomaly detection is crucial in many fields of research, and there are many applications for

the anomaly detection methods. An MRI-machine (magnetic resonance imaging) and health care in general both make use of anomaly detection when looking for example cancerous growth. A typical application of anomaly detection can be found from the use of plastic money. It is often too easy to steal card information, but once the criminal uses the card, it can be seen as an anomaly in the ordinary use of that particular card. (Chandola, Banerjee, and Kumar 2009, 2.) Anomaly detection has been utilized in many fields including the DDoS and intrusion detection, bank fraud, healthcare, production, anomalies in a textual news article, speech recognition, surveillance and biology (Chandola, Banerjee, and Kumar 2009, 11-18).

Anomalies in the dataset are defined as instances that do not to fit in with the rest of the data points. Precisely put, they are not consistent with the other patterns. In Figure 5, the point *a* is considered as an anomaly, if the previous observations have led to the conclusion that normal behavior belongs to either *b* or *c* clusters. (Chandola, Banerjee, and Kumar 2009, 2.)

Methods of anomaly detection in various fields can broadly be divided into classification, nearest neighbor, clustering, statistical, information theory, and spectral methods. (Chandola, Banerjee, and Kumar 2009, 2.)

5.2.2 Anomaly detection concepts

Anomaly detection starts with the information, the data. Data is collected from various sources for anomaly detection, or it is in a format that makes sense for humans. In any case, the data is a set of records, events, examples, files, accounts, or remarks of some sort. These records have features or attributes that distinguish them from each other. These features come in various forms, such as a set of increasing identity numbers, in binary (yes or no) form or textual content. There can be one or more features in a record, and the nature of data determines the method of anomaly detection used. (Chandola, Banerjee, and Kumar 2009, 6.)

Depending on the way the data has been structured, a method can be chosen to find the anomalies best. For data that is a continuous set of values, e.g. a temperature measure at a place, statistical methods are useful. This type is sequential data. Others include spatial, spatiotemporal or graph data types. Individual records that are connected to each other form

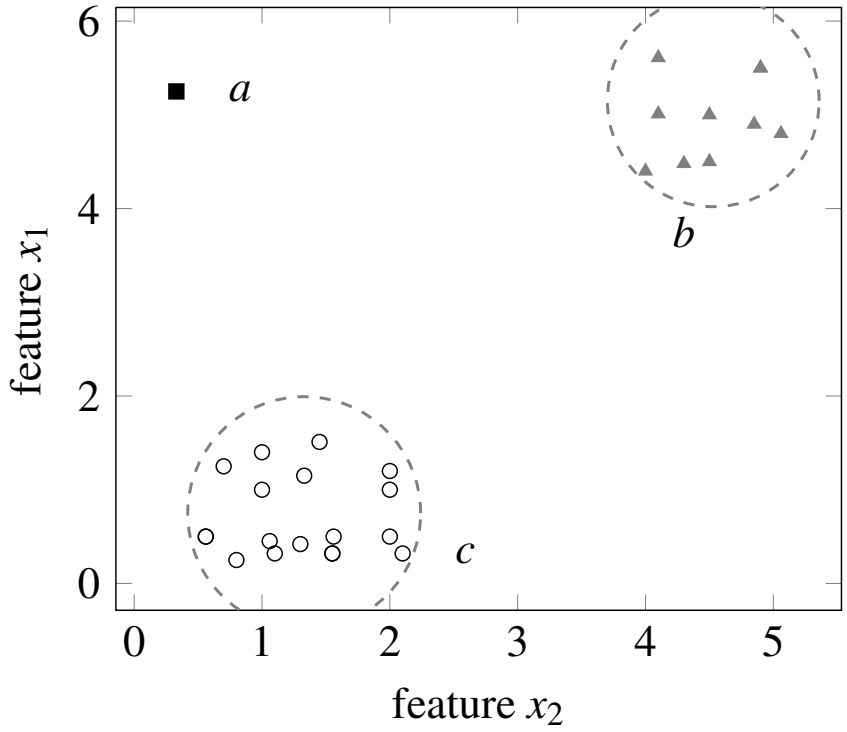

Figure 5. An example of an anomaly in clustered data in a 2-feature plane

a spatial dataset. In graph data, the point of interest are the vertices and edges of the graph. (Chandola, Banerjee, and Kumar 2009, 6-7.)

Before the anomaly detection process can start, the anomaly type has to be defined. Such as data, different kinds of anomalies determine the way one approaches the detection. They are a point, contextual or collective types. Point anomalies are the kind of records of data that stick out from the data no matter how and where it is located in the overall picture. An example of a point anomaly can be seen in Figure 5. Contextual anomalies are what the name suggests, for there to be an anomaly in the behavior. The typical behavior has to be defined. A man shopping for a pair of shorts is not odd, but if he does it in the middle of winter in Finland, it becomes odd compared to what others are doing, i.e. the norm. He is an anomaly in the group. (Chandola, Banerjee, and Kumar 2009, 6-7.)

Another type, also very useful for the detection of DDoS attacks, are collective anomalies. This name comes from the idea that different events in a time series are considered normal, but when they appear in a sequence or near to each other, they constitute an anomaly with

each other. (Chandola, Banerjee, and Kumar 2009, 6-7.) An example in intrusion detection is a stateful protocol analysis, where the server records the incoming packets and matches them with the next ones. A typical TCP three-way handshake needs a sequence of packets: clients send the SYN, server responds with a SYN-ACK, and the client answers with an ACK-packet. With the absence of the ACK packet, a SYN flood is created, and there is an anomaly compared to the standard behavior of the protocol. (Harris and Hunt 1999, 887.)

As with machine learning, there are supervised, semi-supervised and unsupervised methods in anomaly detection. When the process involves a training data that has been classified and labeled into distinct groups, and it is evident which is and what is not anomalous behavior. The anomaly detection testing process involves simple comparing the records with the trained model that represents the data. The issues with this approach have to do with the labeled data itself. It is hard to get data where the anomalies or even the normal behavior are known in advance. There is also a disparity between the number of trained normal behavior and anomalies, which might lead to discrepancies in detection. A possibility to use a supervised system is to introduce anomalies into a perfect dataset of normal behavior. (Chandola, Banerjee, and Kumar 2009, 10.) However, the mere existence of an ideal dataset is problematic when the "normal" is changing.

The semi-supervised method works such as the supervised, but in this case, the standard for an expected behavior or a standard record is already defined, and anomalies are everything else that happens to be outside the class. Because of this attribute, this method is utilized in many fields for their suitability into situations where the anomaly class is not adequately known. (Chandola, Banerjee, and Kumar 2009, 10.)

When anomaly detection method is unsupervised, it only means that it has not received prior training to the data that it is analyzing. The method inherently presumes that outliers appear less often than normal behavior. Thus the occurrences of any differences or oddities are flagged as anomalies. However, in the case when anomalies are not as rare as they are presumed to be, the method falls apart as the anomalies get merged with the normal behavior. A semi-supervised idea, of having a normal class, can be harnessed to work in unsupervised when using a training data without the knowledge of anomalies in the data. For instance, the data does not include labels for even the normal class, but with the presumption of a normal

class being the dominating, it is possible to learn the behavior. (Chandola, Banerjee, and Kumar 2009, 10.) Applications such as the detection of DDoS attacks take advantage of this method since often the sole purpose is to detect zero-day attacks. The general features of an attack can be known, but the way they appear in the data cannot be predicted accurately for unknown intrusion methods.

5.2.3 Classification-based techniques

A classification-based methods rely on the training and testing phases of the detection, where the training data is labeled, and a model is built based on the training data. Based on this model, any further records of the same data can be classified into either normal or abnormal classes. There may be one or many normal classes, depending on the training data. (Chandola, Banerjee, and Kumar 2009, 19.) A typical multiclass classifier is displayed in Figure 5, where the classes *b* and *c* are labeled as normal classes. Anything that does not fall into any of the classes, it is classified as anomalous.

Neural networks can be used as classifiers in the typical sense, meaning that the training data is used to teach the neural network to classify the normal records. Then the new records are given to the neural network, and in case it the neural network fails to classify it, it is labeled as an outlier. Another type of neural networks is a replicator neural network (RNN) that can be used in single-class cases. (Chandola, Banerjee, and Kumar 2009, 19.) For example, Hawkins et al. (2002, 1) propose an RNN-based intrusion detection system with three hidden layers. The number of input and output layers or neurons is based on the number of features in the data. The RNN learns the normal class by recreating or constructing all the record. New records are then given to the NN, and a threshold value for the mistakes for the reconstructed records determines if the record is anomalous. (Hawkins et al. 2002, 2.)

Bayesian network -based anomaly detection works by using the posterior probability of a case belonging to a normal class as the anomaly measure. The method can be used for both multi- and univariate data. Essentially all the features are independent, but there are also Bayesian networks that take dependent features into account when determining the probability. (Chandola, Banerjee, and Kumar 2009, 20.) Viaene, Derrig, and Dedene (2002, 203.)

propose a boosted naïve Bayes for detecting insurance swindling related to car crashes. The data were binary values of various aspects of the accidents such as was treatment given, what was the level of destruction of the vehicle. The training data was labeled, and the employees checked the features.

Support Vector Machines (SVM) compute a non-linear transformation for the training vectors into another vector space with a mapping function Φ. Then the SVM finds a hyperplane between the classes that is as wide as possible, given the support vectors from each class. For a large number of features, the calculations get more and more difficult. (Hearst et al. 1998, 19.) With the hyperplane, it is then possible to find out whether a given test record maps to outside the normal area bounded by the hyperplane, meaning that it is anomalous. SVMs tend to require complex computations. (Chandola, Banerjee, and Kumar 2009, 21-22.) Hu, Liao, and Vemuri (2003) propose a *Robust* SVM for detecting anomalies in network traffic.

Such as the name indicates, rule-based methods use rules acquired by analyzing the training set to determine of a test record fits the rules describing the training data. In case it does not, it is labeled an abnormal. Decision trees as the algorithms are usually inexpensive operations. (Chandola, Banerjee, and Kumar 2009, 21-22.)

5.2.4 Nearest neighbor methods

The idea of the nearest neighbors is that the distance between normal behavior feature vectors is less from each other than anomalies from anything else. The idea is that the normal set is a densely packed group or neighborhood of records. In this method, the scale of similarity or in this case closeness is the distance between the vectors. Quantities such as Euclidean distance or Mahalanobis distance can be used.

"*The anomaly score of a data instance is defined as its distance to its* k^{th} *nearest neighbor in a given dataset* (Chandola, Banerjee, and Kumar 2009, 21-22.)" To give an example, if $k = 2$ is chosen, it is the distance of the second nearest that is evaluated. Other techniques also exist, such as: determining the all the distances to k number of neighbors and summing them up or counting all the neighbors that are within distance d away from the test record.

5.2.5 Cluster analysis -based outlier detection

There are three different kinds of anomaly detection method based on clustering with a different measure to determine the anomaly among the clustered data. (Chandola, Banerjee, and Kumar 2009, 26.) These are:

1. All anomalies cannot be part of any cluster, i.e. they do not form a cluster of similar entries.
2. Entries' distance to the cluster centroid should be small and anomalies appear further away from any centroids.
3. Density and size of the cluster as a measure. The presumption is that normal entries make a cluster of a dense and sizable nature and outliers can be found from smaller clusters. (Chandola, Banerjee, and Kumar 2009, 26-27.)

In an entirely unsupervised mode, a clustering algorithm that does not require all records to be part of a cluster is used. Then the 1. assumption is applied to pick out the outliers. Clustering is mostly seen in an unsupervised method, but it can be applied in a semi-supervised way. An example of a semi-supervised method, the clustered training data is the baseline and then the distance from the closest centroid is calculated to determine if the test entry is an anomaly. (Chandola, Banerjee, and Kumar 2009, 28.)

For the 3. assumption, value for the minimum cluster density is coined. Naturally, the outliers can be found from sparser clusters than the value. Other cluster attributes, such as the size, can also be applied to determine the outlier clusters from the normal ones. (Chandola, Banerjee, and Kumar 2009, 28.)

K-nearest neighbors and cluster -based methods are similar in the sense that they both distance measures in the feature space to determine anomalous behavior, but the difference lies in the concept of clusters, compared to the idea of neighbors of the test record. (Chandola, Banerjee, and Kumar 2009, 29.)

5.2.6 Statistical, information theory and spectral methods

Statistical anomaly detection can be applied when a proposition holds when all the training entries follow a stochastic model. The detection is then based on the idea that peculiarities do not qualify the test. Such a model would be for example where the training data follow a Gaussian distribution, and the test measure is the inverse probability score for the case. If the probability is lower than a predefined threshold of the difference from the mean, it is flagged as an anomaly. For instance, the distance measure could be set as 3-times the standard deviations from the average of the distances. (Chandola, Banerjee, and Kumar 2009, 29-30.)

All models that presume Gaussian distribution can be considered as a group. Members of this group are regression-model systems as an example. Together these are called parametric methods. Non-parametric methods do not assume anything about the shape of the data distribution. In non-parametric methods, the model is built directly from data itself without expectations. Non-parametric methods include histogram and kernel function methods. The histogram method with univariate records, e.g., estimates a histogram of the feature values. Testing happens by evaluating if the test entry resides in some of the defined ranges within the histogram. (Chandola, Banerjee, and Kumar 2009, 32-34.)

Information theory approaches presume that in the presence of anomalies, the data has roughness, asymmetry or inconsistencies that can be picked up by complexity or entropy measures. *Kolmogorov complexity* is an example of a rule of abnormality. (Chandola, Banerjee, and Kumar 2009, 36.)

Spectral methods take advantage of data that can be represented in a lower dimension that the features already are. In this dimension, the expectation is that the anomalies appear different, more than in the original state. (Chandola, Banerjee, and Kumar 2009, 37.) For such transformation can be *Principal Component Analysis* (PCA), where the most representative set of features are selected, and a linear combination is calculated for these values. These are called *principal components*. (Abdi and Williams 2010, 3.)

5.2.7 Contextual and collective anomaly detection

In various situations, the individual instance is not anomalous in a broad sense, but certain contextual or behavioral properties make it one or a part of an anomaly collective. These properties can be spatial, in a graph form, sequential and a profile defined anomalies (e.g. suspicious profile types within credit card use). For instance, the SYN flood example from Section 5.2.2 on page 34 can also be called a collective anomaly because the individual packets are not anomalous. (Chandola, Banerjee, and Kumar 2009, 38-39.) These types of anomalies need a different way to detect them since simple classification of entries would not yield any results.

There are two ways to go about this task: using a point anomaly detection in a smaller set of records where the anomaly should be visible and assuming a particular structure for the data. In the former, the data is broken down into sequences or a group where the expected anomaly is visible. An example of looking for anomalies in mobile phone use is to select the user as a context variable and within that context, search for anomalies in the behavior. (Chandola, Banerjee, and Kumar 2009, 39.) Thus, this example is a contextual anomaly detection using a reduction to a point anomaly case.

When data is not easy to split into smaller chunks, e.g. a time series data. In that case, a better way to approach the issue is to create a model from the training data that can distinguish the anomalies in the desired frame of reference. Finite State Automata (FSA) and Markov Models can be used to model the behavior of a sequence and predict the next observation. In a case of a mismatch between the observation and the model, the event is flagged anomalous. (Chandola, Banerjee, and Kumar 2009, 40.) Obviously, this applies only to data where the model is well known.

In graph data, Sun et al. (2005) propose a relevance score and normality score for detecting nodes from bipartite graphs. A bipartite graph is a set of vertices in two different sets, where the connections between vertices are only between the sets, not within. A vertex is anomalous in case it has edges with two other vertices that it does not share a group with.

5.3 Evaluating the results with ROC-graphs

A receiver operating characteristics (ROC) curves are used to visualize and analyze classifiers, such as the classification algorithms presented in this thesis (Fawcett 2006, 862). In a ROC-graph, a rate of true positives and rate of false positives are plotted in a two-dimensional ROC-space, where for example greater the area under the curve (AUC) means better accuracy in a classification of objects to classes (Bradley 1997, 1146). Psychophysical studies and signal detection evaluation of radars were the first purposes to use ROC-curves to analyze and understand data (Hankey 1989, 308). Earliest uses of these graphs in machine learning go to 1989 when they were used by Spackman (1989) in his paper as an evaluation method with algorithms.

A classification model, i.e. classifier, maps the inputs to anticipated classes based on predictions. The detection of DDoS attacks works as an example. There are two (2) categories of connections, and the problem is to classify them to malicious and legitimate classes. There are four outcomes in this process, where an instance is given class. The aim is to detect the malicious connections. When the classifier makes a positive match, in a case of a malicious connection, it is regarded as true positive (TP), but when it is classified as negative, it can be said that it is a false negative (FN). When the classifier correctly labels a negative (i.e. legitimate traffic) as negative, it is perceived as true negative (TN), but when it is categorized as positive, it is viewed as false positive (FP). (Fawcett 2006, 862.) These values can be then arranged into *a confusion matrix* (Bradley 1997, 1145). See Table 2.

Table 2. A confusion matrix (Bradley 1997, 1146)

True class	Predicted class		
	$-ve$	$+ve$	
$-ve$	T_n	F_p	C_n
$+ve$	F_n	T_p	C_p
	R_p	R_n	N

In ROC-curves, the TP-rate (sensitivity) is plotted on the y-axis and FP-rate (1-specificity) onto the x-axis. The 2-dimensional space is called a ROC-space and it illustrates connection between sensitivity and specificity for different modifiers for the classifier (van Erkel and

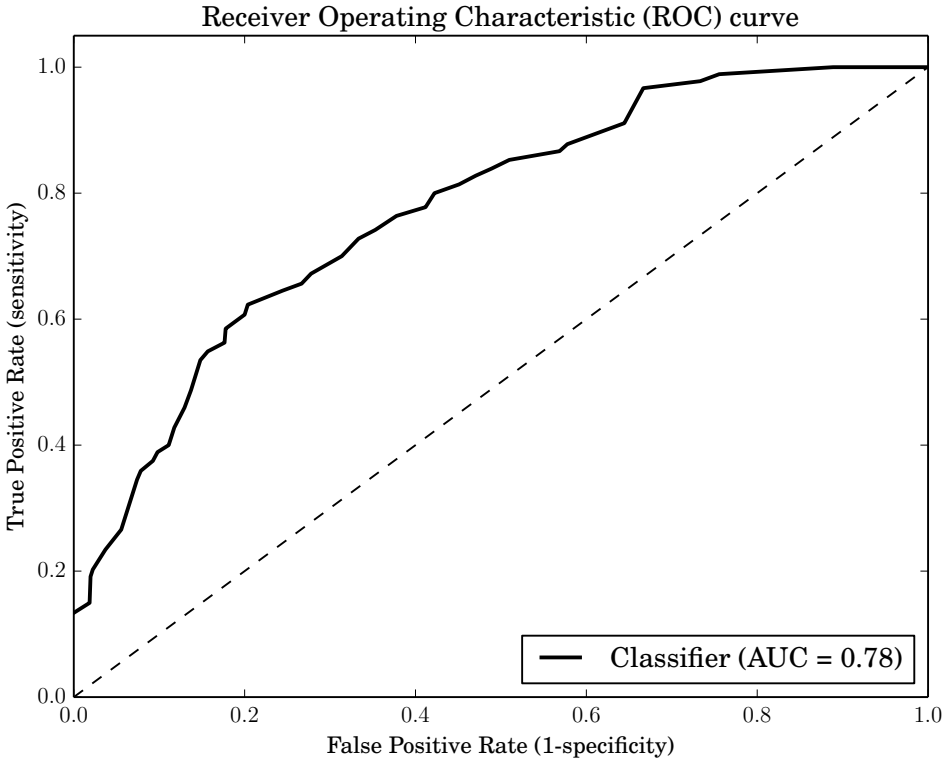

Figure 6. A ROC-curve and an AUC-value calculated

Pattynama 1998, 91). For example, point (0,1) represents a perfect classifier. The point (0,0) on the other hand does not classify anything as positive, thus it does not produce false positives either. In contrast, the other extreme, i.e. the point (1,1) symbolizes a classification, where the classifier would only announce positives. (Fawcett 2006, 861.)

The line crossing the ROC-space from lower left to the upper right corner marks a random classification of instances. When a classifier lies mostly on the left side of the ROC-space, it classifies positives with high confidence. For the classifiers in the upper right corner, false positive rates are high, but (almost) all of the positive instances are classified correctly as positive. The lower right-hand space below the dividing line suggests a classifier that is worse than random guessing, and thus the area usually remains free. (Fawcett 2006, 861.)

Since a ROC-curve portrays the behavior of a classifier in a plane, an area under the curve (AUC) may be calculated to compare different classifiers by a simple number (Fawcett 2006, 868). Bradley (1997) proposes that the AUC is obtained by trapezoidal integration (See Equation 5.1) as the most straightforward way. The AUC-value usually lies between 1 and

0,5, since the AUC of random classification is 0,5 (Fawcett 2006, 868).

$$\mathrm{AUC} = \sum \left\{ (1 - \beta_i \cdot \Delta\alpha) + \frac{1}{2}[\Delta(1-\beta) \cdot \Delta\alpha)] \right\}, \tag{5.1}$$

where

$$\Delta(1-\beta) = (1-\beta_i) - (1-\beta_{i-1}), \tag{5.2}$$

$$\Delta\alpha = \alpha_i - \alpha_{i-1} \tag{5.3}$$

One of the strengths of the AUC in classifier comparison is that it is free of the influence of limits of sensitivity and specificity. On the other hand, most of the ROC-graph values are irrelevant FPR and TPR tuples, e.g. high specificity, but low sensitivity. (van Erkel and Pattynama 1998, 92.) Bradley (1997) suggests that there are benefits for using an AUC. There include bigger sensitivity with Analysis of Variance (ANOVA), the decision threshold remains independent, an evidence of the distinction of negative, positive classes and earlier probabilities of classes do not effect it and represents the effort of the classifier rendering random, and weak classifier with low AUC-values.

In order to decide the best modifier for a classifier is much more than trying to pinpoint the closest to the perfect classifier point. There are clinical and financial matters to be taken into account. In the case of a diagnostic test for pneumonia, for example, tremendous benefits are attained by antibiotic medication with little or no side effects. In this case, false positive classifications are more tolerable, and false negatives should be avoided. That's why the limiting value or modifier is assigned a small number. However, in the case of expensive and dangerous treatments, the false positive rate should be kept as low as possible. Therefore, the choice operational value for the modifier should be made with the effects in mind. (van Erkel and Pattynama 1998, 93.)

5.4 Summary

This chapter introduced anomaly detection, machine learning, data mining and ROC-curves as a comparison tool. As can be seen from a quick look into the anomaly detection research, the are many ways to approach the problem. This chapter mentioned only a few of the

methods. The next chapters tie this knowledge to distributed denial-of-service attacks and the detection methods, most of which are based on machine learning and data mining. The knowledge about ROC-curves comes in handy in Section 7.3 which reviews the accuracy of various methods.

6 Detection of distributed denial-of-service attacks

This chapter starts by explaining intrusion detection systems and DDoS anomaly detection methods. Then the chapter builds a taxonomy of the detection methods from the DDoS literature. This taxonomy is helps understanding of what types of methods exist and eventually it is used to classify the detection methods in the systematic mapping study. This chapter presents the protocol and the phases that the author did during the mapping study. At the end of the chapter, the results and eventually also the answer to the first research question are presented.

6.1 Intrusion detection systems

Defending against a DDoS attack can happen either proactively or reactively (Mirkovic and Reiher 2004, 49). In this thesis, I speak about reactive means, i.e. detection and mitigation.

A DDoS attack detection can happen at different points on the network stack depending on what kind of DDoS attacks the detection method tries to identify, what is the purpose of detection and what kind of mitigation strategies the administrators want to implement. A simple measure to combat the simplest attacks is to implement preconfigured settings to a host or network firewall (e.g. an IP list, or ports that are not used), which blocks all requests from a host that matches the criteria (Raghavan and Dawson 2011, 283). However, this does not work with more sophisticated attacks that attackers are using nowadays, and the firewall itself might be vulnerable as it tries to combat an influx of packets and hosts that it needs to block. A separate detection system takes the responsibility to detect attacks to address the issue.

An intrusion detection system (IDS) can be deployed in the routers, switches hubs or separate analysis units on the network segments. They are usually called network intrusion detection systems (NIDS), but an IDS can also be deployed on the host. Thus they are named as a host intrusion detection system (HIDS). (Whitman and Mattord 2011, 298-302) An IDS on its own simply detects an attack and notifies the administrator. To handle attacks and remove any delay in response, intrusion prevention systems (IPS) have been developed which already

encompass a set of steps the system can take in case it detects an attack. These technologies are usually deployed together; they are called intrusion detection and prevention systems (IDPS) (Whitman and Mattord 2011, 293).

A Host IDPS is deployed on a host that the system observes and informs the administrators of any changes to the filesystem or a specific folder that might be of interest for the attackers. The strength of an HIDPS is that it can check the traffic going in and out in addition to all the changes happening to the system. It can also analyze encrypted packets, as it resides on the target host of the packet. They can perform the tasks of the network IDPS for the host traffic. (Whitman and Mattord 2011, 302-304.) An example of an open source HIDPS is OSSEC[1].

Network Intrusion Detection and Prevention Systems (NIDPS) oversee the traffic in the network, by analyzing it and notifying the administrator who can take appropriate actions to mitigate the attack. An NIDPSs falls into one of three categories: signature matching, anomaly detection and a hybrid model of both mechanisms. (Rastegari, Hingston, and Lam 2015, 1.) An example of an NIDPS is Snort, an open source tool[2].

A signature-based system works such as an anti-virus program, where a set of features (e.g. a known sequence of packets or a certain time interval of packets that is characteristic for a known attack) have already been programmed to a so-called signature database. This information is then compared to the packets that the systems come in contact with to see if they match any of the signatures. Other packet validity verification methods can also be used. For example, abnormal packets that do not fit the description of the TCP/IP protocol definition or application layer protocols such as the HTTP or the XML. Many denial-of-service attacks work by sending malformed requests to consume the processors and memory of the servers. (Whitman and Mattord 2011, 298-299.) It works relatively well and without causing normal traffic to labeled as intrusive very often, but it cannot detect completely unseen or zero-day attacks. (Amoli and Hämäläinen 2013, 1.) That is where anomaly detection works the best.

Anomaly detection is also divided into a statistical analysis and a stateful protocol analysis. The statistical analysis can also be called behavior analysis. The methods are based on a

1. ossec.github.io
2. www.snort.org

baseline traffic patterns which are gathered from the network when it is not under attack and the traffic is presumed to be "normal." A baseline is a set of features captured, e.g. types of packets, a number of flows or time of arrival. A section of the traffic in the live system is then captured and matched against the normal state of traffic using statistical analysis or machine learning techniques. When the packets do not fall into an acceptable range from the "normal," they are flagged as anomalous. At his point, the administrator gets a message, and further steps can be taken. (Whitman and Mattord 2011, 305.)

Anomaly detection does not need to know the attack before it happens, and it is not tied to a specific type of traffic, as long as the set of features under inspection contain the anomalies from which the attack can be detected. A statistical analysis may use resources extensively in the host, and it might not be able to tell the difference between normal and malicious traffic if the attacker disguises the attack traffic to look normal on purpose. Also, when the target system experiences a wide range of traffic from inactivity to high load and diverse to uniform traffic, anomaly-based systems cause many false positives. Thus, signature-based systems are frequently favored over anomaly-based. (Whitman and Mattord 2011, 305.)

A stateful protocol analysis (or what Mirkovic and Reiher (2004, 49) call Standard Anomaly Detection Strategy) is based on detecting deviations from a known set of features that are present in the normal use of a protocol. These features are defined by the vendor of the protocol. (Whitman and Mattord 2011, 306.) A method that relies on the protocol standard as a measure of normality is called standard mechanism. This method only produces true positives, as all normal use of the protocol does not cause an alarm. On the other hand, the method does not detect attacks that use the protocols normally, e.g. application layer DoS. (Mirkovic and Reiher 2004, 49.) As an example, the systems has knowledge of the number of HTTP packets a browser could send in a second when a user uses the website in a normal way. In a case of an abnormal influx if packets, a human user can be ruled out and the traffic can be marked as anomalous.

Several detection methods have been developed to notice DDoS attacks. Figure 7 shows categories of the detection methods. Few intrusion detection systems are deployed in an application layer firewall, right before a web application to mitigate any incoming attack. The firewall can inspect encrypted packet for malicious content and protect the application.

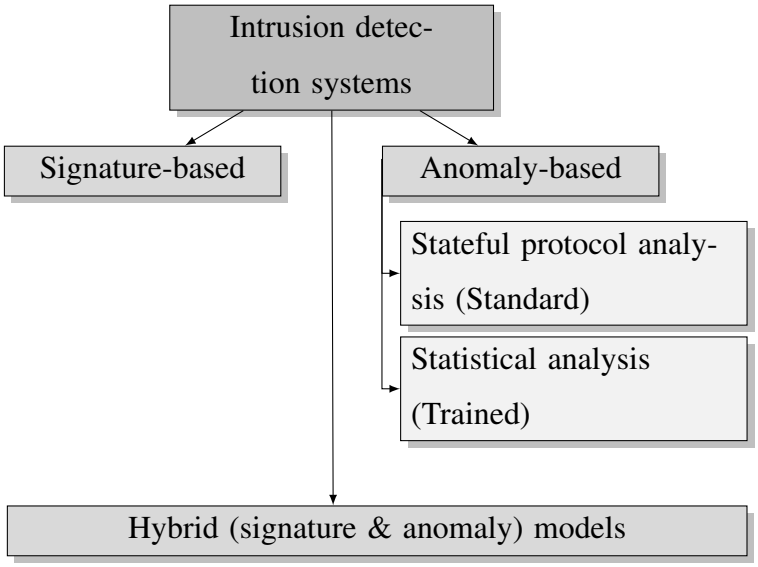

Figure 7. Classification of intrusion detection and prevention systems and their detection methods (Mirkovic and Reiher 2004, 49.) (Whitman and Mattord 2011, 293-305.)

These firewalls have to deployed so they do not become a bottleneck in case of high volumes of traffic to the application (Raghavan and Dawson 2011, 251).

6.2 Anomaly-based detection methods

There are always three phases to detection of DDoS attacks: capturing a flow (i.e. segment of the traffic) for analysis, analyzing the traffic and determining if the flow is malicious. Regardless of the system, anomaly or signature-based, this has to be done in a split second, meaning at the speed at which the packets come. That can be up to 10Gbps or more. (Raghavan and Dawson 2011, 133-134.) Anomaly-based systems tend to computationally expensive to run, and the complexity of the problem increases when more features from the traffic are taken under investigation.

A flow is characterized as a group of packets, which have attributes that link them together, that are routed through a detection station in a given time-period (Claise 2008, 1.). An example could be streaming server that is identified by a batch of UDP packets with the same source IP, the port, and a similar length.

Various detection methods usually define the features that they take into account when ob-

serving the traffic. These features can be the time interval, or connection flows, the volume of packets per second, IP addresses or packet types. A usual value under inspection is the amount of ingress traffic (i.e. inbound), but also the amount of egress traffic (i.e. outbound). (Raghavan and Dawson 2011, 133.) Anything that the observer (in this case the NIDPS) can see can be taken into the system as a set of features. The more features there are, the more complex and thus more demanding the decision process becomes.

Trained anomaly-based method is divided into two parts: training and testing. During training, the baseline is created to be able to detect the deviations. During testing, the baseline is compared to new data and the flows from new data are classified either to be malicious or normal. (Patcha and Park 2007, 3452.)

Anomaly-based methods try to get as close to perfect detection as possible. This is usually given as less than 1% false positives. Anything more and the system becomes an issue itself. (Hu, Liao, and Vemuri 2003, 4.) Even 1% false positive rate may cause unwanted alarms in high speeds requiring constant attention from the administrators.

The gradual improvement in doing a task, e.g. classifying faces into categories of emotions, for a computer application, is called machine learning. Machine learning techniques work in a similar manner as the statistical methods, creating a baseline and comparing it with the newest data. However, the focus is on building a model that is better than simply looking at the raw data, a model that learned from the earlier outcomes of the application. A machine learning approach gets better and may alter the underlying method of detection in the face of new data. (Patcha and Park 2007, 3455.)

Statistical analysis watches the behavior of the system and builds a baseline (or a profile) from network logs, intensity of the CPU and other parameters of interest. The profile of the system is re-evaluated or updated after some time interval. All the time, as new data comes in, the system runs a function using a given set of parameters from the flow to get a grade on how anomalous it is. A general threshold is set for the grade. If the grade exceeds the threshold, an alert is given. (Patcha and Park 2007, 3453.) The difference to a standard-based system is that the baseline comes from the system itself, not a general protocol standard.

Sequence analysis of system calls has the assumption that all the programs have a certain

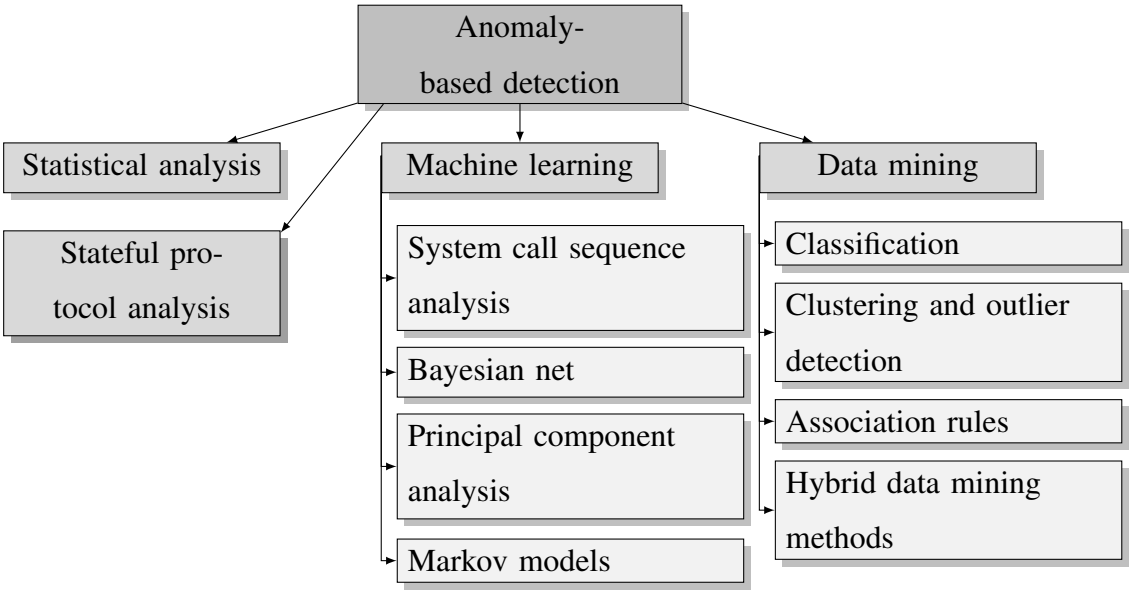

Figure 8. Classification of anomaly-based detection methods as extended by Patcha and Park (2007) and adopted hybrid methods from Tama and Rhee (2015, 3742.)

fingerprint in the way they make system calls and any divergence could mean a malicious use of the program. For instance, it is under attack or it is being utilized for a task that it normally does not do. The sequences have a certain length to be inspected at any given time. From the normal sequences it is possible to build a profile all programs, and once the pattern of calls would differ from the profile, an alarm was raised. (Patcha and Park 2007, 3455.)

6.3 The systematic mapping study

This section explains the protocol and walks through the phases of the mapping study that the author did to gather the publications.

6.3.1 Related work

Systematic mapping studies and systematic literature reviews are not very common in the DDoS attack literature. There has been much research on the DDoS detection methods, but not many systematic reviews appear in the literature to draw conclusions and to map the research area. This can be confirmed by searching for "systematic literature review" or "sys-

tematic mapping study" with variations of the term "denial-of-service attack". I was not able to find the mapping studies related to denial-of-service detection methods, and especially a DDoS attack in encrypted network traffic. Therefore, the related work section contains mapping studies that mention DDoS attacks and systematic literature reviews about DDoS attacks and methods.

da Silva et al. (2013) found 661 papers observing 7 different kinds of threats to cloud computing by performing a systematic mapping study based on the Petersen et al. (2008) guidelines. DDoS attacks were listed as one of the issues in the abuse threat group of the study. As their sources, they used IEEE Xplore, ACM, SpringerLink, ScienceDirect, Scopus, and Engineering Village. The study follows the guidelines well, and the article has been published in a peer-reviewed journal[3].

Zapata, Alemán, and Toval (2015) carried out a systematic mapping study according to the Petersen et al. (2008) guidelines. They found 344 papers and concluded with percentages of the most prominent security issues, the most prominent being data protection. They found only a few papers related to DDoS attacks. The literature search was performed on IEEE Xplore, ACM, ScienceDirect and Wiley InterScience databases. This is a quality study, although being published in a non-refereed journal[4].

Tama and Rhee (2015) conducted a literature review on data mining techniques related to detecting DDoS attacks, where they listed 35 methods presented in scientific articles. These publications were gathered and selected from two different online databases. Papers were selected between the years 2007 and 2015. They found different methods which were divided into four classes based on the machine learning application. These categories were: association, classification, clustering and hybrid methods. (Tama and Rhee 2015, 3740.) The categories align with the taxonomy presented by in Figure 8. The method is a systematic literature review, but the authors do not cite any particular method guidelines. The journal, where the article was published, is not peer-reviewed[5].

Gutiérrez and Branch (2014) performed a mapping study and a systematic literature re-

3. Ulrichsweb ISSN 1947-5500
4. Ulrichsweb ISSN 1820-0214
5. Ulrichsweb ISSN 1343-4500

view on machine learning techniques in detecting DDoS attacks. The method was based on Kitchenham and Charters (2007) guidelines. They searched papers from 2008 to 2012 from Scopus online literature database. They included 54 papers after conducting a mapping study of 141 articles. They found that popular methods described in the literature were the CUSUM (Cumulative Sum), the SVM (Support Vector Machines), the PCA (Principal Components Analysis) and Neural Classifiers. They note that application layer protocols should be researched more. (Gutiérrez and Branch 2014, 2.) However, there are limitations in this study. The protocol or audit trail for the systematic mapping study and systematic literature review are minimal. The journal (Revista NOOS Vol. 4) is not peer-reviewed[6].

Latif, Abbas, and Assar (2014) did a systematic literature review of DDoS attacks related to Wireless Body Area Networks and their defense mechanisms. The review was based on Kitchenham et al. (2009) guidelines. They did a comprehensive online literature search on ACM Digital Library, IEEE Xplore, Springer Link, Science Direct, and Elsevier Journals, limiting the search to years 2009-2014 January. The also searched gray literature. They included 31 papers. They found that a TCP SYN flood is the most common attack in the literature targeting a WBAN (wireless body area network). Jamming, collisions or routing were also addressed. (Latif, Abbas, and Assar 2014, 9.) The paper is only minor limitations, and it was published in refereed journal[7].

Mouli and Jevitha (2016) made an analysis of web service security using a systematic literature review research method. They conducted a search on a "data repository" but they do not disclose the name to the reader. They found 36 papers and concluded that the literature talks about DDoS attacks, with the XML injection coming close. Mouli and Jevitha (2016, 876.) For a conference paper published in a peer-reviewed conference proceeding journal[8], the level of detail in the audit trail is very limited. Main issue is that the authors do not tell how they obtained the literature.

6. Ulrichsweb ISSN 0123-5591
7. Ulrichsweb ISSN 0148-5598
8. Ulrichsweb ISSN 1877-0509

6.3.2 The mapping study protocol & the research question

As a systematic literature review method, I used a light version of a systematic mapping study or a systematic scoping study. A comprehensive version would be to conduct a full manual search, several snowball search rounds and involve several researchers to evaluate the papers and the process, such as Kaijanaho (2015) did in his Ph.D. dissertation. Because of time constraints, the nature of this thesis and field of interest, I decided to conduct a single automatic electronic search using several online computer science literature databases. These databases were IEEE Xplore[9], ACM Digital Library[10], Google Scholar[11], Scopus[12], and ScienceDirect[13]. Some of the databases were only accessible with the license of the University of Jyväskylä. Springer Link was considered, but it lacked a way to limit the search to only metadata, and the results returned thousands of articles with the full-text search. Because Google Scholar returns papers from Springer Link, I decided to drop this database.

The search was conducted on 13-15th of October 2016, and thus the time constraint for publications was chosen September 2016. Example article that should be found was identified to be the paper by Zolotukhin et al. (2015).

Before the mapping study started, I created a protocol and saved it to version control system alongside with the thesis to keep track of all the changes. See Appendix A for the full protocol and change record. The protocol includes the scope, the RQ, databases, examples studies, search terms used, roles, inclusion criteria, exclusion criteria, quality criteria, data extraction items from each study, synthesis guidelines, reporting and a schedule.

Before the actual search, I did a test search and tried all the parameters to find out what and how much I would find. It turned out that many of the initial search terms yielded too many papers to go through and analyze in the time that I had to conduct the study. Therefore, few additional constraints were introduced starting from the search terms all the way to the inclusion and exclusion criteria. In this way, I was able to get a sample that represents the

9. ieeexplore.ieee.org
10. dl.acm.org
11. scholar.google.com
12. www.scopus.com
13. www.sciencedirect.com

research area. After the constraints had been introduced and I was sure that I could analyze the papers that were found, I presented the mapping study protocol to my supervisor.

The research question: RQ1: What methods for detecting encrypted DDoS attacks are presented in the scientific literature?

From the RQ1, I identified the following terms *"detection methods," "DDoS attacks," "encrypted network traffic,"* and *"scientific literature."* To increase the sensitivity and specificity of the study, the following terms were subject to a thesaurus search: *"DDoS," "encrypted"* and *"detection method."* The following Thesaurus services were used: Thesaurus.com, MOT Collins Compact Thesaurus[14], Oxford Thesaurus of British English[15]. The search resulted in including the plural of *"detection method"* as well as *"detecting"* and *"detect."* Also both terms, *"denial-of-service"* and *"denial of service"* were included in the search term. This also makes sense as some of the example studies also discuss the terms in different forms and instead of *"detection method"* only say *"detecting DDoS attacks"* in their name or abstract.

Before the start of the evaluation, if several methods were presented in one paper, each of the methods would be evaluated and displayed separately. This also applies to duplicate studies which present the same experiment of a method in two different papers. They should be noted to be the same study.

6.3.3 Collection process

After the evaluation of the terms in the RQ1, Table 3 presents all the terms and their alternative forms that were used to construct the search terms. The purpose of this step was to increase the sensitivity and specificity of the study, i.e. take various ways of writing into account.

The word *"attack"* was dropped since the term denial-of-service implies an attack in the literature I have come across and if this would not be the case in a paper, that could easily be rejected. The word *"network traffic"* was also dropped for the same reasons as *"attack."* Today, the main body of literature is focused on DDoS attacks coming from the network, not

14. JYU access
15. Apple Dictionary

Table 3. Search term formulation

Distributed denial-of-service attacks	**Detection methods**	**Encrypted network traffic**
DoS	detection	encrypted
DDoS	detecting	encryption
denial of service	detection method	encrypting
denial-of-service	detection methods	SSL
	detect	TLS
		HTTPS

caused by other means such as physical threats. The scientific literature in the RQ1 limits the search to the literature which has been published in a journal, conference proceedings, a technical paper or theses. This rules out patents and gray literature for example. By searching the databases as mentioned above, this constraint is taken care of, as the databases include mainly scientific literature. If some non-scientific articles were in the search results, they were not included in the study. For instance, such cases were master's theses that were discussing detection methods of DDoS attacks or news articles that were included in the Google Scholar results.

The terms related to encrypted network traffic were chosen to expand to papers that assume the their reader knows that an *SSL/TLS* and an *HTTPS* refer to encrypted traffic. Also a *"Secure Socket Layer"* and a *"Transport Layer Security"* were considered, but the paper that would include these would also include the abbreviations, at least in the keywords section.

The final search term was then formulated in the following manner: *("DoS" OR "DDoS" OR "denial of service" OR "denial-of-service") AND ("detection method" OR "detection methods" OR "detecting" OR "detection" OR "detect") AND ("encrypted" OR "encryption" OR "encrypting" OR "SSL" OR "TLS" OR "HTTPS").* See Table 4 for adaptations for each of the search engines.

The search was limited to at least title, abstract, and keyword search. In IEEE Xplore this was done by selecting the "Metadata Only" option in the Command Search. In Scopus, the metadata option was defined in the search term in the Advanced Search field. ACM Digital

Library does not offer to search all the metadata at the same time, nor does it support fully combining the search terms into a proper query. Therefore, I decided to only search from the title field and leave Google Scholar to catch the rest of the papers from ACM that the search missed. Searching ACM for all fields (including the full text) resulted in so many papers (404 results in 28th of September 2016) that I did not have time to go through all of them. I limited the search to ACM Full-text Collection.

With ScienceDirect, I decided to go the other way because of lack of results with only the abstract, the keywords and the title -fields. I included all the fields, i.e. also the full text, for the *encryption* constraint. Expert Search of ScienceDirect, books were excluded and Computer Science was selected as the field of study. The search engine of ScienceDirect required me to put curly brackets around all the terms to be treated case-sensitive. Some of the terms, e.g. "detection methods" yielded better results if I used the wild-cards instead of typing the terms separately. With this search, I was able to bring down the results to a manageable level while keeping the most relevant articles in the result set. Table 4 presents the final search expressions.

During the pilot search, I encountered a drawback with Google Scholar. I could not limit the search in a clever way to get only the most relevant studies. Therefore, I decided to drop it because of a lack of time for the selection process. The final amount was 170 results. There were too many, and data extraction would have taken too much time. Studies presented in Petersen, Vakkalanka, and Kuzniarz (2015) do not use Google Scholar as one of their sources. Kaijanaho (2015), Brereton et al. (2007) and Kitchenham et al. (2009) recommend using Google Scholar. I discuss the credibility of the study in Section 8.1.

6.3.4 Screening process

Table 6 shows the overall, the exclusive contributions and the specificity of each source. There was only one overlap between sources (i.e. [S3]). Thus, all sources are valuable for the study. See Tables 9 and 7 for details. I did not calculate the sensitivity (see Section 2.1) of the study, because I cannot objectively estimate the complete set of all relevant studies.

The selection process consisted of screening the results for inclusion using the inclusion

Table 4. Summary of the search terms by database

Database	Search term
IEEE Xplore	(("DoS" OR "DDoS" OR "denial of service" OR "denial-of-service") AND ("detection method" OR "detection methods" OR "detecting" OR "detection" OR "detect") AND ("encrypted" OR "encryption" OR "encrypting" OR "SSL" OR "TLS" OR "HTTPS"))
ACM DL	+acmdlTitle:(DoS "denial of service" denial-of-service DDoS) AND +acmdlTitle:(detecting detection detect "detection method" "detection methods") AND +(SSL encryption encrypt encrypted TLS HTTPS) "filter": owners.owner=HOSTED
ScienceDirect	TITLE-ABSTR-KEY({DDoS}) OR TITLE-ABSTR-KEY({denial of service}) OR ({denial-of-service}) OR TITLE-ABSTR-KEY({DoS}) AND tak(detect*) AND ALL(encrypt*) OR ALL({SSL}) OR ALL({TLS}) OR ALL({HTTPS}) [Journals(Computer Science)]
Scopus	(TITLE-ABS-KEY (dos OR ddos OR denial-of-service OR denial of service) AND TITLE-ABS-KEY (detection method OR detecting OR detect OR detection methods) AND TITLE-ABS-KEY (ssl OR encrypted OR https OR tls OR encryption OR encrypting))
Google Scholar	intitle:"DoS" OR intitle:"DDoS" OR intitle:"denial of service" OR intitle:"denial-of-service" AND intitle:detection* AND encrypt* OR "SSL" OR "TLS" OR "HTTPS"

Table 5. Search results and paper yield per database

Database	Date	Results	Studies after IC	Studies after EC
IEEE Xplore	13.10.2016	59	24	4
ACM DL	13.10.2016	78	15	4
ScienceDirect	15.10.2016	66	24	3
Scopus	15.10.2016	22	11	4
Google Scholar	(29.09.2016)	(170)	0	0
Total		225 (395)	74	15

Table 6. Evaluation metrics from each data source

	IEEE	**ACM DL**	**ScienceDirect**	**Scopus**	All
Overall contrib.	4	4	3	4	15
Overall contrib. %	~26.7	~26.7	20.0	~26.7	100.0
Exclusive contrib.	3	4	3	3	13
Exclusive contrib. %	20.0	~26.7	20.0	20.0	~86.7
Specificity %	~6.9	~5.1	~4.5	~18.2	~6.67

Table 7. Overlap matrix for each of the data source

	IEEE	**ACM DL**	**ScienceDirect**	**Scopus**
IEEE		0	0	1
ACM DL	0		0	0
ScienceDirect	0	0		0
Scopus	1	0	0	

criteria (IC), and after the initial yield, a further exclusion criterion (EC) was used to drop papers which were not usable in this study, even if they included all the search terms and discussed a detection method. During the initial sweep, I evaluated the title, keywords and abstract of the papers before I made a decision to include the paper for the next phase. For the purpose of getting all the possible candidates, the inclusion criteria were chosen to be quite gentle. Once I had downloaded the paper, the second round of exclusion would look to the possibility of detecting encrypted traffic. In the exclusion phase, also the full text was evaluated. I checked the conclusions and the results chapters of the papers first to determine if the studies were relevant for this study.

Inclusion criteria (IC):

- IC1: Papers that present a DDoS detection method
- IC2: Papers which were published until 2016 September
- IC2: Papers written in English
- IC4: Papers which I can access in full text (JYU has access or the article is free)
- IC5: Papers where the detection method could be applied to the TCP/IP network

I evaluated the initial search results based on the title and abstract and dropped papers which did not precisely fit the study. I also included some articles that only hypothesized about a detection method or the paper discussed an entirely different networking scheme. I did that to get the most accurate results and save time in this phase. The choice, however, led to more papers in the next step. If I was not able to determine the eligibility of an article for inclusion only by looking at the abstract, keywords and title, I added it to the next stage. I confirmed that abstracts vary in the DDoS detection literature, such as Kitchenham and Charters (2007) pointed out regarding software engineering.

Exclusion criteria (EC):

- EC1: Papers which do not present results, but merely hypothesize of a detection method
- EC2: Papers which are already included from another source
- EC3: Papers which present a detection method based on packet payload analysis
- EC4: Gray literature, lecture notes and books

Applying the EC to the downloaded studies after the IC was a long and difficult task, since I did not want to exclude a paper without concluding fully that it was not suitable for my use. This process evolved during the pilot extraction as I introduced stricter criteria. I understood that I need only papers that study the TCP/IP and Internet-based DDoS attacks. Then the number of papers to be included shrunk greatly. The extra papers that were talking about DDoS attacks and included the word *encryption* or other combinations studied MANETs (Mobile Ad-hoc Networks) or WSNs (Wireless Sensor Networks).

I decided to exclude DoS papers that were about MANETs and WSNs as they were not directly applicable to the detection methods on the Internet. There was also one paper about Opportunistic Networks (OpNet), where also the method are not compatible. This decision saved time a lot, since after reviewing few of the papers, I discovered that the evaluation methods were usually related to rate caps and simple methods. This is because of the nature of the network nodes, where computationally expensive operations are slow and costly because of the limited battery power of the nodes. Therefore, the potential methods presented about encrypted traffic, might not be entirely applicable. For instance, Sedjelmaci and Senouci (2014, 3640) note that energy consumption is a valuable asset for WSN nodes. Thus they cannot rely on long and complex computations. Madhavi (2008, 7) also propose that MANET nodes have limited resources as well.

I conducted a random retest according to Kitchenham and Charters (2007, 20) recommendation. This happened after the data extraction had already finished. I did it to see if I had missed something and some study should have been included. This was a validation step to ensure that the findings are valid for myself and the reader. The process was that I took all the 74 papers after the inclusion criteria (see Appendix B and Table 5) and generated seven random numbers with a pseudo-random generator. For the purposes of this study, a pseudo-random generator was enough, as the purpose was to have a random set of papers to re-evaluate, i.e. I could not choose myself which papers I would retest.

I chose seven because Kitchenham and Charters (2007) do not give an example, as the case varies and I wanted to have papers from each source. Once I had papers from each source, I stopped generating numbers. I believe that five would have been already enough, and therefore seven is more than sufficient. I marked the papers that were chosen by the random

number generator and retested to the Appendix B with a circle. I read the paper and compared the decision to the previous one. No decision changed. For these papers, I also marked the page number of the paper, where the decision is based on. There was no retest done before the inclusion criteria, i.e. directly from the search results, other than comparing the results with the pilot phase, which were similar.

The majority of the papers did not specify the features to be extracted from the network traffic in the abstract, conclusions or the results. I needed to read almost the whole paper to discover that the whether the method was using the payload of the packet or not (i.e. would it be eligible for encrypted traffic analysis). Most of the papers that investigated this phenomenon wrote about it clearly. However, there are papers that I decided to include because of the nature of the method and the applicability to the DDoS attack detection from encrypted traffic, based on the payload feature question. I tell more on these papers in the results section of the study and how they are related to the overall picture of the study. In the next section, I explain the quality measure to distinguish these papers.

6.3.5 Evaluation of quality

Quality criteria (QC), score 0 or 1:

- QC1: Do the authors acknowledge the efficacy of the method for encrypted traffic?

Quality assurance of the papers in mapping studies is not the main focus. The idea is not to judge the papers, but to get an image of the research area with all its research. Because of that reason, I took a quality measure into account, because I wanted include papers that might extend the view of the research area. These papers did not explicitly specify their method to be applicable in encrypted networks. Therefore, there are two kinds of studies included, the ones that acknowledge the efficacy of their method to encrypted traffic or directly study the phenomenon and the ones that do not take payload inspection into the equation, but do not acknowledge applicability to encrypted traffic. The method is capable of detecting a DDoS attack over SSL/TLS with the assumption that when the payload is not inspected, but only the packet header or other information derived from packet flows.

6.3.6 Data extraction and mapping process

Table 8. Form for data extraction

Name	Type
ID	Integer
Title	Text
Authors	Text
Year	Integer
Venue	Enum(journal, conference)
Detection method	Text
Strategy	Text
Example data	Text
Sample attack(s)	Text

I performed the data extraction phase in the order of publication starting from the most recent. I arranged the final list of papers included in the study in Excel and added the data fields from the protocol (see Table 8). I kept the initial research question in mind when reading and analyzing the final set. The detection method is a simple term e.g. the name of the algorithm, program or application used in the detection. Both the training and testing phases were taken into account in determining the underlying method of detection.

The strategy measure was based on the earlier classification of the detection methods based on the taxonomy in Figures 8 and 7. Example data is a single name of the data, where it was mentioned. Sample attacks were also included if they were mentioned. In most of the cases, it was not clear from the example data what kind of attacks were in the data. For this reason, this information was not included. At the time of data extraction, I was not sure how I would use the example data other than note the problems related to generated sets.

During the exclusion phase, I decided to include few studies that showed an example of a study which could be an encrypted traffic detection method but did not specifically mention that. I wanted to differentiate them from the primary studies that were deliberately researching DDoS attacks in encrypted traffic. For that reason, I added a letter *i* and a quality score

for the studies in the Excel sheet.

As a result of the data extraction, I had a table of all the information. This information was divided into several tables for brevity and lack of space. I did not want to make one large table with all the data, but rather smaller chunks of the information always referring to the correct table when discussing the findings. These are the Tables 9, 10, 11 and 12.

The number of publications per year, in Figure 9 was counted by hand from the publication year of the paper. Sometimes that was difficult, as both the submission and acceptance dates were given. I chose the publication year of the paper to be the final publication year of the journal or conference the paper was presented.

The final list includes 14 publications from four different sources. In the next section, I present the results of the mapping study. In Section 8.1, I discuss to the validity of the mapping study.

6.4 DDoS attack detection methods in encrypted network traffic

6.4.1 The results of the mapping study

Table 9 exhibits the selected studies, with the source and a quality score. The quality measure was implemented because these studies did not mention doing research on detection methods of encrypted network traffic, but since they do not apply packet payload features in the anomaly detection schemes, I included them to see the full picture of the research as it is now. The primary studies that were about DDoS detection in encrypted traffic are on the list, but the ones that do not mention or acknowledge the viability of the detection method of encrypted traffic are marked with the letter *i* and the simple quality score is *0*.

The study was concentrated in scientific literature, and the qualified studies are from peer-reviewed journals, conferences, and workshops. From the distributions of selected studies by their forum (see Figure 10) of publication, can be seen that the research concentrates in conference publications. All the journals are peer-reviewed, checked from Ulrichsweb:

Table 9. Selected studies

ID	Authors	Title	Source	Quality
S1	Eliseev and Gurina (2016)	Algorithms for network server anomaly behavior detection without traffic content inspection	ACM	1
S2	Zolotukhin et al. (2016b)	Weighted Fuzzy Clustering for Online Detection of Application DDoS Attacks in Encrypted Network Traffic	Scopus	1
S3	Zolotukhin et al. (2016a)	Increasing Web Service Availability by Detecting Application-Layer DDoS Attacks in Encrypted Traffic	IEEE, Scopus	1
S4	Zolotukhin et al. (2015)	Data Mining Approach for Detection of DDoS Attacks Utilizing SSL/TLS Protocol	Scopus	1
S5	Petiz et al. (2014)	Detecting DDoS Attacks at the Source Using Multiscaling Analysis	IEEE	1
S6	Wang et al. (2015)	DDoS attack protection in the era of cloud computing and Software-Defined Networking	ScienceDirect	1
S7	Hoeve (2013)	Detecting Intrusions in Encrypted Control Traffic	ACM	1
S8	Amoli and Hämäläinen (2013)	A Real Time Unsupervised NIDS for Detecting Unknown and Encrypted Network Attacks in High Speed Network	IEEE	1
S9i	Das, Sharma, and Bhattacharyya (2011)	Detection of HTTP Flooding Attacks in Multiple Scenarios	ACM	0
S10i	Shiaeles et al. (2012)	Real time DDoS detection using fuzzy estimators	ScienceDirect	0
S11	Chen, Chen, and Delis (2007)	An Inline Detection and Prevention Framework for Distributed Denial of Service Attacks	Scopus	1
S12i	Lee et al. (2008)	DDoS attack detection method using cluster analysis	ScienceDirect	0
S13i	Caulkins, Lee, and Wang (2005)	A Dynamic Data Mining Technique for Intrusion Detection Systems	ACM	0
S14	Abimbola, Shi, and Merabti (2003)	NetHost-Sensor: A Novel Concept in Intrusion Detection Systems	IEEE	0

Elvisier Computers & Security[16], Elvisier Computer Networks [17], Elvisier Expert Systems with Applications[18] and Oxford The Computer Journal[19]. See the full list of selected studies with venues and journals in Table 10.

16. Ulrichsweb ISBN 1872-6208
17. Ulrichsweb ISBN 1389-1286
18. Ulrichsweb ISBN 0957-4174
19. Ulrichsweb ISBN 0010-4620

Table 10. Included studies and their publication forums

Study	Venue	Refereed
[S1] Eliseev and Gurina (2016)	9th Intl. Conference on Security of Information Networks 2016	-
[S2] Zolotukhin et al. (2016b)	16th Intl. Conference NEW2AN 2016 & 9th Conference ruSMART 2016	-
[S3] Zolotukhin et al. (2016a)	IEEE 23rd Intl. Conference on Telecommunications	-
[S4] Zolotukhin et al. (2015)	15th Intl. Conference NEW2AN 2015 & 8th Conference ruSMART 2015	-
[S5] Petiz et al. (2014)	16th Intl. Telecommunications Network Strategy and Planning Symposium 2014	-
[S6] Wang et al. (2015)	Computer Networks 81(2015) 308-319	Yes
[S7] Hoeve (2013)	1st ACM Workshop on Smart Energy Grid Security 2013	-
[S8] Amoli and Hämäläinen (2013)	IEEE Intl. Workshop on Measurements and Networking 2013	-
[S9i] Das, Sharma, and Bhattacharyya (2011)	Intl. Conference Communication, Computing and Security 2011	-
[S10i] Shiaeles et al. (2012)	Computers & Security 31 (2012) 782-790	Yes
[S11] Chen, Chen, and Delis (2007)	Oxford The Computer Journal 50(1) (2007) 7-40	Yes
[S12i] Lee et al. (2008)	Expert Systems and Applications 34 (2008) 1659-1665	Yes
[S13i] Caulkins, Lee, and Wang (2005)	43rd ACM Annual Southeast Regional Conference Vol. 2	-
[S14] Abimbola, Shi, and Merabti (2003)	IEEE 8th Intl. Symposium on Computers and Communications 2003	-

As I explained in Section 6.3.5, the quality score was implemented to gain a better knowledge of the available detection methods and the research area as a whole. Figure 9 illustrates all the included studies and their publication years in a bar chart. The studies with the full

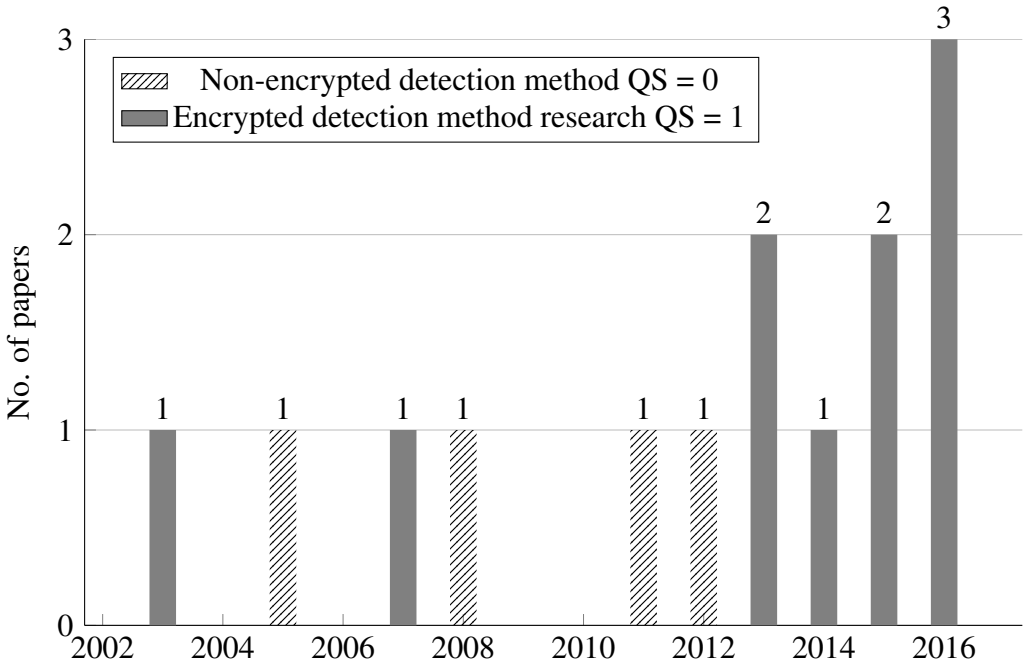

Figure 9. Selected papers published by year

quality score have a solid color and the ones that may be applicable to encrypted network traffic (i.e. QS = 0), have a diagonal line pattern. The first publication that consciously addressed DDoS attacks in encrypted traffic is from 2003. Toward the end of the figure, in the past years, the amount of research increases. This increase in interest is in line with the fact that Arbor Networks (2013, 25) reported an increase in application layer attacks using encrypted traffic in 2012. The need for research in the field is increasing.

In Figure 10 display the venues of publication of the papers in a pie chart. Such as in computer science in general, the selected publications have concentrated in conference proceedings and workshops. The sample is not large enough to make an assumptions on the actual overall distribution of publication venues in the DDoS attack research. However, the distribution gives and an idea of where current research is published.

The strategy (or classification of the method) by Mirkovic and Reiher (2004) and extended by Patcha and Park (2007) are used to evaluate the current status of the DDoS detection in encrypted network traffic. Based on the taxonomy in Figure 8, the comparison to the selected articles on the detection methods in encrypted network traffic can be made, and possible new

Table 11. Detection methods in encrypted networks from included studies

Study	Detection method	Strategy	Features
[S1]	Correlation functions & MLP	Statistical analysis & Classification	Server response rate metrics
[S2]	Fuzzy c-means	Fuzzy clustering	Statistics and data from packet headers
[S3]	Single-linkage, K-means, fuzzy c-means, SOM, DBSCAN & SAE	Classification (NN) & clustering	Statistics and data from packet headers
[S4]	DBSCAN, K-means, k-NN, SOM, SVDD	Clustering	Packet header statistics
[S5]	Multiscaling Analysis	Statistical analysis	Number of packets & average energy per timescale
[S6]	Probabilistic inference graphical model	Bayesian networks	Chow-Liu algorithm for feature decision
[S7]	Edit distance -based searching	Statistical analysis & clustering	time, size and direction of the packet
[S8]	DBSCAN	Statistical analysis & clustering	Packet header and flow data in different resolutions
[S11]	Signatures & stateful protocol analysis	Signature & stateful protocol analysis	TCP, UDP and ICMP packet headers and statistics as well as payload
[S14]	Snort signatures	Signature & system call sequence analysis	packet payload

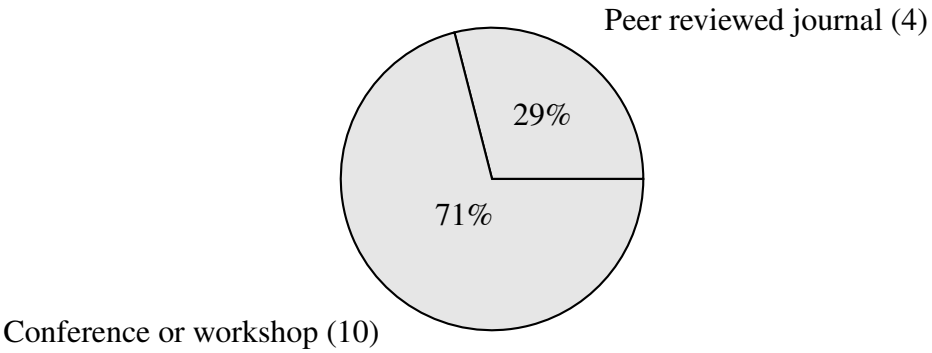

Figure 10. The publication venue distribution of the included studies approximately

Table 12. Applicable methods from non-encrypted research in included studies

Study	**Detection method**	**Strategy**	**Features**
[S9i]	Statistical analysis, pattern disagreement and projected clustering	Statistical analysis and clustering	TCP header data & packet rate per interval
[S10i]	Fuzzy estimator	Statistical analysis	Mean time between network packets
[S12i]	Hierarchical clustering	Clustering	TCP header information & number of packets
[S13i]	Classification tree	Classification	TCP header data

areas of research could be revealed. In the next section, I present the summaries of each method, and how it classifies according to the previously mentioned taxonomy of detection methods. See the distribution of methods in various categories in the bubble plot in Figure 11 after the explanations of methods in the papers.

6.4.2 Summaries of the included studies

[S1] Eliseev and Gurina (2016) use correlation functions of data block size & number of packets per time unit observed from the webserver. They use long time intervals, i.e. three weeks of real data to train. They propose two algorithms. The first looks at the Pearson correlation coefficient between cross-correlation functions in a similar time interval in the current and training sets. The second algorithm uses a multilayer perceptron (MLP) with Levenberg-Marquardt algorithm to train and test the current cross-correlation functions. A threshold for the reconstruction error is set to determine an anomalous function. They say that these algorithms can be easily implemented as a lightweight DDoS HIDS in IoT devices. The method uses both statistical analysis and classification.

[S2] Zolotukhin et al. (2016b) propose a method for detecting DDoS attacks in encrypted network traffic in both offline and online case using fuzzy c-means clustering algorithm. In the method, they train the system with flow information such as conversation length, packet velocity, packet size averages, and flags. They build feature vectors form the information by also normalizing the values with min-max normalization. They have two different versions of the algorithm: an online and an offline version. The tests of the method are conducted using the Realistic Global Cyber Environment (RGCE), where the attacks can be simulated as realistically as possible. Slowloris, SSLsqueeze, and some advanced DDoS attacks were tested in the system and they found that the trivial cases such as Slowloris and SSLsqueeze were detected nearly 100% of the time, whereas the advanced DDoS attacks had only 70% accuracy when keeping the false positives to the minimum. Categorical classification of this method is clustering.

[S3] Zolotukhin et al. (2016a) study the application layer DDoS attacks in encrypted network traffic employing hierarchical, centroid- and density-based clustering algorithms and a

stacked auto-encoder (SAE). The features for clustering come from the packet header information and conversation to the server by each user. The conversations are mended together from two different flows with matching sources and destinations. After this, statistics such as the velocity of packets, extent, flags and the number of encrypted messages are extracted into tuples for clustering. The tuples are normalized with max-min normalization. Using the clustering methods described in Table 11, the most common DDoS attack types can are detected by comparing the incoming flows to the clusters. In each of the type, a different deviation measure is used. For example, for centroid-based algorithms, a threshold is set for the maximum distance of the vector from the normal traffic cluster centers. The common DDoS attack types are Slowloris and Slow POST and a more advanced DDoS attack mimicked the behavior of the users in a web service. This attack was detected by combining conversations from the same source together, calculating the approximate similarity form each cluster by percentages and applying the stacked auto-encoder. A reconstruction error value of the SAE is the anomaly measure. The methods presented in this paper put the paper in both classification and clustering categories.

[S4] Zolotukhin et al. (2015) present a clustering-based anomaly-based detection method using DBSCAN (density-based spatial clustering of applications with noise) and comparing the algorithm with others such as SOM (Self-Organizing Map), K-means, k-Nearest Neighbors and Support Vector Data Description (SVDD). The features for training and testing data use only packet header statistics such as the averages of packet sizes or TTL (Time to live), TCP flag appearance averages, no name a few. The feature vectors are min-max normalized. If the pairwise distance from the nearest cluster member is more than the maximal pairwise distance for that cluster, it is labeled as an anomaly. The method is categorized in clustering methods because of the various clustering algorithms used in the detection.

[S5] Petiz et al. (2014) propose a statistical analysis detection method in the source network that uses a multiple scale traffic analysis. The statistics used as the features are statistics of the packets flows. Thus they conclude that this method is also applicable to encrypted traffic. The detection method is based on the premise that DDoS attacks have a pseudo-periodicity fingerprint in the traffic. By calculating an average energy for packets per second from multiple time intervals, the anomalous traffic should have higher energy in one interval

length. This paper is an example of a purely statistical analysis of network metrics.

[S6] Wang et al. (2015) develop a complete NIDS with detection and mitigation modules for software-defined networks (SDN). Their detection method is based on probabilistic inference graphical model that updates itself all the time in order to fight a data-shift issue, unlike traditional Bayesian networks. The data shift issue assumes that the training data and real attacks imitate the same statistical frequency (Wang et al. 2015, 313). The features are not preselected by the researchers but by a Chow-Liu algorithm. They are selected from flows or packet headers. After applying the algorithm, commonly the relevant variables have been linked to the Chow-Liu tree. These are chosen for the analysis of the graphical model. The graphical model is an adaptation from Bayesian networks. Thus the category for this paper is in BN.

[S7] Hoeve (2013) explore an intrusion detection method for encrypted control traffic. A packet series search and comparison using edit distance is the measure of the difference between the series. The method uses time, size and direction of the packet to form the feature vector. Traffic consists of series of these vectors. The training phase is done by a clustering the series into clusters. The next phase searches for series with approximate string matching and edit distance. The series which are over a set threshold, are malicious. This method uses both statistical methods and clustering. Thus these are the categories.

[S8] Amoli and Hämäläinen (2013) have designed an NIDS to work with large amounts of data. The method first employs an algorithm that uses statistical analysis to detect variations in the flows. If an anomaly is detected, the second phase with DBSCAN starts. The outliers from the final set of clusters are flagged as anomalous and a potential DDoS attack. Thresholds for the DBSCAN, the minimum size of the clusters is set to 5% of the number of flows and the maximum distance between vectors shall be fixed to the average Mahalanobis distance of the vectors. From the anomalous traffic, the starts to pinpoint the attacker from C&C traffic patterns. Because of the two different phases, the category of this paper is in both statistical analysis and clustering.

[S9i] Das, Sharma, and Bhattacharyya (2011) have developed a three-phased method for detecting DDoS flooding attacks. The first phase uses a simple threshold value for the number

of HTTP requests per interval. The second takes advantage of parallel time interval request rates and computes a pattern disagreement value. The maximum of this value during a time with no known attacks is considered as the threshold for anomalous traffic. The third method uses packet header data and projected clustering with Oracles SQL queries. They create an index to determine the type of the cluster either malicious or normal. The first two are online and the last is an offline detection method. This paper belongs to both statistical analysis and clustering-based groups.

[S10i] Shiaeles et al. (2012) propose a detection method that uses the packets arrival times in small time windows. It is assumed that DDoS attacks mean packet arrival does not follow the Poisson distribution. An α-cuts fuzzy estimator is used to derive a single fuzzy value for the mean arrival times in the earlier time window. Then the current mean time is compared to the value. If the value is less than the fuzzy value, an alarm is raised. If it is more, the traffic is considered to be normal. They note that flash crowd events might be flagged as DDoS attacks using this method. I chose to include this paper, as the method does not require any payload inspection and could be utilized as a detection method in encrypted network traffic. This method is based on a statistical analysis.

[S11] Chen, Chen, and Delis (2007) have developed an NIDS, called *DDoS Container*, that uses several detection methods in succession to detect DDoS attacks from network traffic. They consciously acknowledge that their method does not fully comply with encrypted network traffic, but say that the behavioral analysis of the stateful inspection does also catch flows that are encrypted. As stated in Table 11, the method combines both stateful protocol analysis and signature-based payload inspection. The system is placed in the network, and it has been placed in a segment where all the traffic flows through between two switches, presumably before or in the DMZ (Demilitarized zone). The system comprises of multiples phases of detection, whose names are Protocol Decoder, Behavior Police, Session Correlator, Message Sequencer, Traffic Distinguisher, and Traffic Arbitrator. The first three take care of the stateful protocol analysis and the latter three of the more careful packet inspection using signatures. At the beginning only the header information of the packet is taken into account, and therefore it is possible to detect malicious flows with abnormal behavior also in encrypted traffic. Thus, this paper is in both stateful protocol analysis and signature-based

detection classes.

[S12i] Lee et al. (2008) propose a hierarchical clustering-based detection method that uses various entropy values and other metrics calculated from the packet header information as the features. The vectors are normalized by standard deviation before clustering. Euclidean distance is used as the measure of similarity. The method is purely a clustering-based method.

[S13i] Caulkins, Lee, and Wang (2005) use a decision tree to detect DDoS attacks. The learning phase was done in a supervised manner from the known attacks of DARPA1999 IDEVAL dataset. Only the TCP packet header information was taken into account. The decision tree classifies connections into either intrusive or normal classes. This is a categorically classification-based method.

[S14] Abimbola, Shi, and Merabti (2003) discuss the difficulty of signature-based systems and encrypted network traffic. They propose a host-based IDS where they can access the payload of the encrypted traffic. The HIDS lies right after the application layer and detects DDoS attacks using signatures for network traffic packet payload and analyzes the system calls of the target application. The, categorically signature and system call analysis -based, paper is the first included in the mapping study that notifies the difficulties of analyzing encrypted network traffic and consciously researches the field.

6.4.3 Datasets and sample DDoS attacks

Table 13 depicts the included papers with their datasets and their sample DDoS attacks. The purpose of this section is to discuss the efficacy of the methods based on the facts about the datasets and sample DDoS attacks. Research on the DDoS detection in encrypted network traffic is conducted in many ways. For example, running the experiments on a non-encrypted dataset (generated or real), generating totally encrypted dataset or observing real traffic data. I illustrate some of the problems in intrusion detection research.

Studies S1, S9i, S10i and S13i used real network traffic in their investigations from their universities or other sources. These kind of datasets are difficult for the simple reason that they are unlabeled data when reconstruction and validation of the method becomes an issue. Other than S1 also used other datasets to verify their findings, and Eliseev and Gurina (2016),

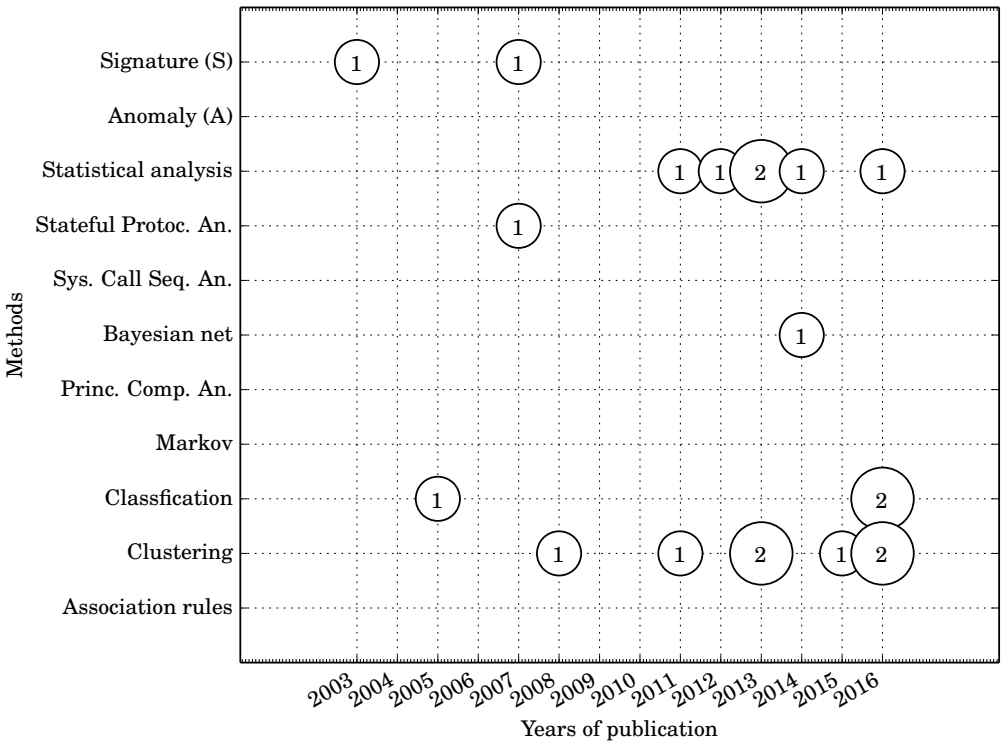

Figure 11. Detection methods by class in a bubble plot over the years

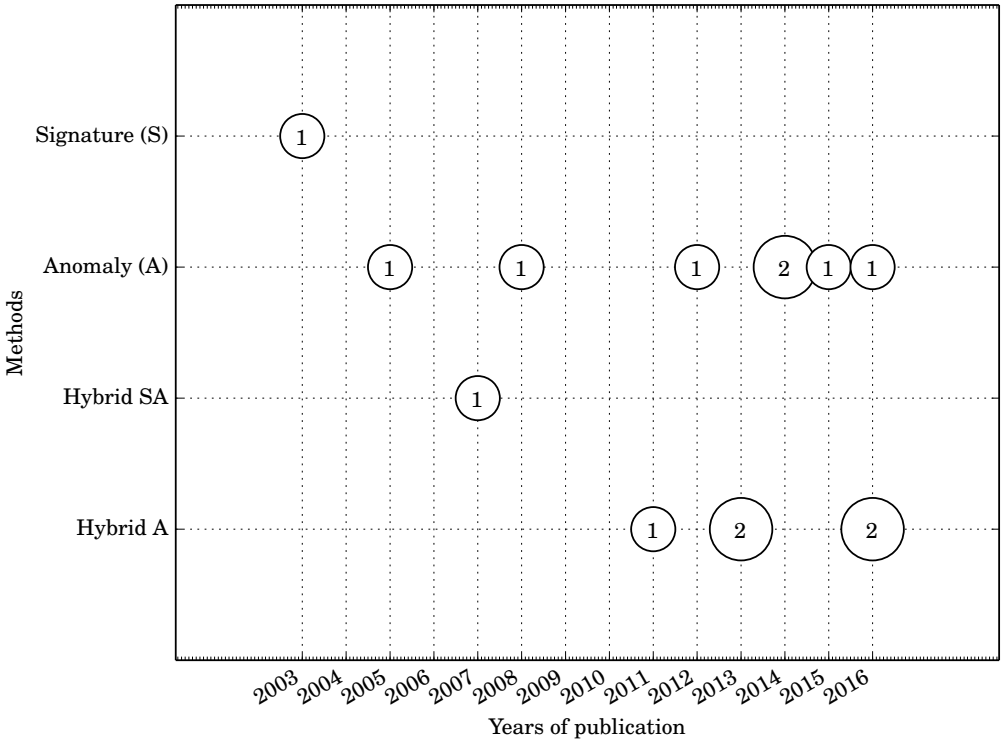

Figure 12. Detection methods classified in hybrid classes over the years

Table 13. Sample DDoS attacks and datasets used in included studies

Study	Data/simulation	Sample DoS
[S1]	Real server traffic	-
[S2]	RGCE	HTTPS Slowloris, SSLsqueeze and an advanced human-like DDoS
[S3]	RGCE	HTTPS Slowloris, Slow POST and an advanced DDoS
[S4]	RGCE	Simple HTTPS flood
[S5]	Lab net generated	Hping3 TCP SYN flood through SSH & VPN tunnels
[S6]	Traffic w/ Amazon EC2 & private lab cloud & UNB ISCX dataset gen.	HTTP Slowloris & HTTP flood & other intrusions (Shiravi et al. 2012, 369.)
[S7]	Lab generated data (PLC simul.)	-
[S8]	(no tests)	-
[S9i]	KDD'99, LBNL & real university network	ICMP, SYN flood (Tavallaee et al. 2009, 3.)
[S10i]	LLS_DDOS_1.0 DARPA2000 & real university network generated	Hping & BlackEnergy Bot & DARPA-attacks
[S11]	Lab test net	Mainly floods: TFN2K, Stacheldracht, Trinoo and Mstream
[S12i]	2000 DARPA IDSSD	Mstream TCP & UDP flood & others
[S13i]	1999 DARPA IDEVAL & university net data	Several attacks incl. ICMP, SYN floods (Mahoney and Chan 2003, 7-8.)
[S14]	Lab net gen. data	W32.DoS and IGMP Nuke & others

in S1, tell that their method needs more investigation and can only work as a lightweight IDS. Such as Shiravi et al. (2012, 372) point out, most of the publicly available datasets are anonymized, because of privacy reasons, to the point when they are not real anymore. Therefore, real is better. Applying the detection methods to a university network has its upside of being natural, but reproducing of the study is difficult given that the data is not available and probably was not anonymized before the tests. Not anonymizing the data makes the detection method be as close as a live NIDS as possible.

Kokkonen et al. (2015) have developed the RGCE used by Zolotukhin et al. (2015) S4 and later studies from the same research group in S2 & S3. It is a Realistic Global Cyber Environment meant of various simulations of Internet situations. The environment acts such as a real Internet where the geographic locations have been simulated and the normal traffic in the network is generated. The system mimics the real behavior of the Internet as well as possible, but all the data is generated, and the environment is completely offline. Within the control system, most common DDoS attacks are also simulated by a packet generation and replay program.

Detection results from bigger networks tend to be applicable in a small network, but the outcomes are seldom transferable from smaller to bigger (Sommer and Paxson 2010, 312). Nevertheless, researchers are forced to create small laboratory settings to test their hypotheses. Papers S5, S6, S7 S11, S14 are using this approach to test their methods, making it the most popular in this sample of studies. Also, studies that use the RGCE are still lab networks, albeit in a bigger scale. Especially when studying DDoS attacks, aspects that are prevalent in real life (e.g. the locations and variance in IP addresses) are difficult to replicate. The traffic generated in a lab environment has the problem, such as in publicly available generated datasets, that they are pseudo-random and do not represent the real world perfectly.

Generating truly anonymized datasets is difficult as the underlying relationships of individual data points in the traffic are dependent on each other. Semantically speaking these relationships have to be both kept and anonymized to keep the legitimacy of the data. (Coull et al. 2009, 232.) Based on the sequence of packets, their flags and direction it is possible to determine operating systems, programs, and web applications that the people have been using, not even looking at e.g. the HTTP packet payload.

For now, KDD'99 is the most widely used dataset for intrusion anomaly detection research. Searching for research from Google Scholar with "KDD 99 DDoS detection" with the limit for 2016 gives 122 hits (27th of November 2016). These papers may use the dataset as their only one or if it is one of many datasets for comparison. Nevertheless, the KDD'99 is still in use. The dataset has been crafted from the DARPA'98 Lincoln Lab simulation traffic, which in turn is generated synthetic traffic. The data is labeled. This set has been studied and criticized widely and further improved sets have been proposed. (Tavallaee et al. 2009, 1-2.). It has been demonstrated that the DARPA-dataset contains artifacts that skew the results (Sommer and Paxson 2010, 309). It is clear that these datasets are not up to date anymore, the Table 13 also shows that the prevalence of these sets at least in an encrypted DDoS attack research is not an issue. The extension studies that did not mention encrypted traffic in their research (studies marked with *i*) are the ones that use these datasets in their research. S9i use KDD (Knowledge Discovery and Data Mining) '99 and LBNL (Lawrence Berkeley National Laboratory) datasets. DARPA-datasets are IDSSD (Intrusion Detection Scenario Specific Dataset) and IDEVAL (Intrusion Detection Evaluation) dataset, used by S10i, S12i and S13i.

Interestingly, Shiravi et al. (2012) aim to solve the issues of real data versus generated data and the problems with privacy issues by proposing a systematic and dynamic way of creating datasets with profiles. These profiles symbolize the behavior of the traffic and the attacks. Based on these profiles, the dataset can be created for many protocols, volumes and situations. (Shiravi et al. 2012, 372-373.) This is how S6 has generated its data for the laboratory environment.

6.5 Answer to the research question 1

This section answers to the first research question about the existing detection methods of DDoS attacks in encrypted networks, based on the literature in the previous section.

RQ1: What methods for detecting encrypted DDoS attacks are presented in the scientific literature?

As can be seen in Figure 11, methods that exist in the scientific literature currently are con-

centrated in a clustering and a statistical analysis -based methods. I found several methods including statistical analysis with correlation functions, multiscaling analysis and edit distance -based searching. Classification-based methods with neural networks such as Single-linkage and MLP were included. The clustering methods included fuzzy c-means, K-means, and DBSCAN. One method with probabilistic inference graphical model was observed. Stateful protocol analysis and system call sequence analysis were also observed. Even signature-based model that tackled encrypted traffic implemented to the application layer was noted.

As expected, majority of the research is focused on non-encrypted traffic and these were the few papers that I was able to find using the systematic mapping study method. As I was conducting the mapping study, I came across four methods that I included from this pool of research which did not take encrypted traffic knowingly into account. These methods could be applied to encrypted network traffic, but they did not test or mention the eligibility of their method. They could also be included as a possible new area for testing. These were pattern disagreement analysis, fuzzy estimators and classification trees.

6.6 Summary

This chapter explained the difference detection methods of DDoS attacks and especially anomaly-based techniques in encrypted traffic. The author searched the literature for detection methods and backed the findings by conducting a systematic mapping study. In summary, encrypted traffic can be detected by signature when the access to the payload is granted, but current research focuses on anomaly detection methods because they can be deployed in routers as well. The chapter answered the research question about what methods are found from the encrypted DDoS detection literature. In the next chapter, a clustering anomaly detection method is presented.

7 Simulation experiment with clustering

This chapter investigates the detection of DDoS attacks in encrypted network traffic in more detail using K-means++ clustering method. It includes experiments and a comparison of results with previous research. At the end of the chapter, the second research question is answered.

7.1 Theoretical setting and implementation of the detection method

In the paper Zolotukhin et al. (2015) propose a DDoS detection method for encrypted network traffic which uses K-means cluster method. This method was not able to handle large files as the K-means needs to have all the vectors in memory to start clustering. As a solution, I suggested that we would use streaming K-means with K-means++ seeding. Mikhail made the changes to the code. The following description explains the method.

7.1.1 Feature selection and anomaly detection

In this process, traffic is considered as flows of a small interval. A conversation between two hosts is a merger of two flows, where in the first flow traffic goes from one host to another, and in the next flow it comes back. There exist source and destination IPs and ports of both hosts for each of the conversations. (Zolotukhin 2016, 1.)

For these conversations, a system calculates the next parameters, which form the feature vectors, such as in Zolotukhin et al. (2016b):

1. The length of conversation in time
2. The sum of individual packets per 1 second
3. The sum of bytes in the same time
4. Packet size metrics: max, min, avg
5. The TCP window metric: max, min, avg
6. % of packets with TCP flags
7. % of encrypted packets with attributes such as “handshake” or “alert” (Zolotukhin

2016)

The method utilizes max-min normalization strategy with the range from 0 to 1 to normalize the scales of these features. Based on these feature vectors, the system creates a baseline model for the traffic in the training phase. Attacks are hypothetically distinct from this model and can be detected as anomalies. The model is a set of clusters, where each of the clusters is a behavioral pattern in the normal traffic. Conversations which are alike should, therefore, be clustered in the same cluster. Attacks should not belong to a cluster. (Zolotukhin 2016, 1.)

For K-means, i.e. centroid clustering, the measure of membership in a cluster is the distance from the center point in the cluster in Euclidean space. That distance should not be greater than a value that must be defined for each cluster separately. This value is calculated for the normal traffic clusters by $T = \mu + \gamma\sigma$, where T is the value, μ is the average of distance for member vectors from the centroid, γ is a tuning variable to find the perfect measure in the testing phase, and σ is the standard deviation of the distances of normal traffic vectors from the centroid. (Zolotukhin 2016, 2.)

7.1.2 K-means algorithm

K-means algorithm is commonly referred as Lloyd's algorithm; that was suggested in 1957 by him. The idea is to find the midpoint (or the mean) for k number of clusters by assigning points to midpoints and then recalculating the midpoint within the cluster until the midpoints do not change. (Braverman et al. 2011, 2.)

As described by Arthur and Vassilvitskii (2007, 1028), the K-means algorithm is as follows:

1. Initiate feature vector space $X = \{x_1, x_2, x_3, ..., x_n\}$, where each x_i is a feature vector
2. Select k starting centroids $B = \{c_1, c_2, c_3, ..., c_{k-1}, c_k\}$ at random from X
3. For $\forall i \in \{1, 2, 3, ..., k\}$:

    - Add vector $x \in X$ to belong to cluster G_i if the distance between x and $c_i \in B$ is less than the distance between any other cluster centroid $c_j \in B$ (where $j \neq i$), for all $x \in X$.

4. For $\forall i \in \{1,2,3,...,k\}$:
    - Reassign centroid c_i for cluster C_i so that $c_i = \frac{\sum_{x \in G_i} x}{|G_i|}$ where $|G_i|$ is the number of vectors x in cluster G_i and the new centroid is the mean of all the features
5. Repeat steps 3 and 4 until centroids c_i for clusters G_i do not change anymore

In practice, this method is sure to find the local optimum and all the feature vectors need to be in memory to use this method (Arthur and Vassilvitskii 2007, 1028). In a normal K-means clustering algorithm, all data is considered and clustered. Streaming clustering comes in handy if the aim is to detect anomalies in real time traffic. To dynamically change the cluster centers resulted by the new conversations arriving at each stage, the old ones need to be discarded changing the makeup of the model.

7.1.3 K-means++ and K-means#

Arthur and Vassilvitskii (2007, 1029) propose an addition to the random selection of the first cluster centroids, thus calling their clustering algorithm K-means++ approximation. The algorithm is as follows, and it replaces the point 2 in the normal K-means:

1. Select the first centroid $c_1 \in B$ uniformly from X randomly
2. Select the following centroids $c_i = x' \in X$ according to probability $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$
3. Repeat 2 $k-1$ times (i.e. k centroids exist)
4. Move to point 3 in the K-means algorithm (Lloyd's algorithm)

In the K-means++ algorithm, the starting centroids are as far away from each other as possible, given that the first selection dictates how the next rounds go. Function $D(x)$ is the distance to the nearest centroid that has been selected for vector x. In the algorithm, for each vector, the distance to its nearest centroid is divided by the sum of all distances for nearest centroids for vectors in X. The vector x' is chosen, if it has the highest probability, and then the distances are calculated again, repeating the step 2 until the desired number of clusters has been reached.

Ailon, Jaiswal, and Monteleoni (2009, 4) propose and improvement to the K-means++ by instead of selecting k centroids per round, they select $3 \cdot logk$ centroids resulting in a better

running time overall. They call it K-means#. Using these algorithms, they also propose a single linear scan streaming divide & conquer strategy for large datasets.

7.1.4 Analyzing traffic stream and detecting outliers

Using this strategy, it is possible to divide the stream of packets, i.e. the traffic flows, into pieces and select new centroids. In this method, the pieces are dictated by τ successive time windows t. The length of that piece depends on the amount of memory available. In this way, the system can go through a large file or stream network traffic directly from the network interface in an online mode, creating a model for testing. (Zolotukhin 2016, 2.)

The baseline model consists of the cluster centroids, the number of vectors corresponding to each cluster, the sum of all raw value of all features in each vector, largest feature attributes, and smallest feature attributes. Now, the distance of the new training dataset conversation and its closest centroid is evaluated. If this is greater than threshold $T = \mu + \gamma\sigma$ for that cluster presented above, the conversation is labeled as intrusive. (Zolotukhin 2016, 3.)

7.1.5 Implementation

The IDS uses the same code base proposed in Zolotukhin et al. (2015), but the streaming and seeding clustering algorithms replace the existing clustering methods. The implementation is done in Python and can run in both online (i.e. listening to the network interface directly) or in offline where the traffic is read from a PCAP-file.

In the network simulation phase, the IP addresses of the attackers were noted and the final statistics of the detection method are calculated based on the known attacker IP addresses. If all the malicious connections were from only the attacking machines, the classifier would be perfect. In the next section, the results of the simulations are presented.

7.2 Experimental setup

In this section I describe the setup to run the experiment and detect DDoS attacks in encrypted traffic. I created datasets in a virtual network environment built for this purpose by

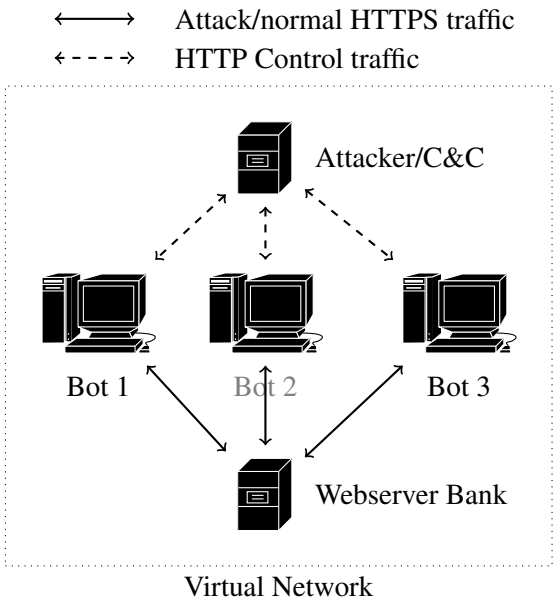

Figure 13. Virtual network simulation architecture

capturing the traffic in a PCAP-file.

7.2.1 Setup of the botnet environment

I created a botnet to generate traffic, both legitimate and DDoS attacks. It is a network of virtual Ubuntu machines running in Oracle VirtualBox[1]. The network consists of three bots, a webserver, an attacker C&C server. The three bots connect to the C&C server via a Java-application running on each one of them. When the C&C server sends a command, the bots start attacking the webserver or sending normal traffic by accessing the website periodically. All the traffic to the webserver is encrypted with the SSL/TLS protocol, and thus cannot be eavesdropped from anywhere else from the network. The webserver has a program (`tcpdump`) running to capture all the connection in the network. This C&C and botnet are implemented by using a botnet environment meant for educational purposes, called FrankenB.

FrankenB is not supposed to be sophisticated botnet infrastructure, but it can do what is needed in this research. The bots themselves are a simple Java programs that initiate the

1. www.virtualbox.org

connection with the C&C server at random intervals. The communication is done by HTTP POST requests and the return from the server are the commands that the bots should do. That can either be "wait for a command," "send normal traffic" or "launch a slow HTTPS GET DDoS attack." The bots send their traffic to the webserver over an SSL/TLS encrypted channel, thus making communication immune to an HTTP signature detection. (Bégin 2011, 5-7.)

In FrankenB, the C&C server URI is hard-coded in the application, which is fine for these purposes (Bégin 2011, 11.). This research ignores the C&C traffic altogether. This botnet does not stand a chance if a security expert would try to detect such a botnet and its control traffic. Setting up this botnet and running the attacks was enough for these simulations. The hypothesis is that the attacks would be easy to detect with clustering anomaly detection methods from the botnet traffic trace.

7.2.2 Running attacks in the network

The botnet was configured to send HTTPS GET & POST traffic to a bank website. The traffic is configured to look as human as possible, considering that it is generated by a piece of software. The purpose is to create normal traffic in the network for training. The bots generated normal page accesses and bank transactions on the online banking website, including logging in, making a transaction and logging out. This is seen as the normal state in the network in this simulation.

The bots were also configured to send a Slowloris (HTTPS GET) and slow HTTPS POST DoS collectively, creating a denial-of-service attack to the website. Under the hood, the Slowloris opens a connection and uses a for-loop to send one piece of the request in each second. That piece is a single HTTP header every time. From the point of view of the server, there is nothing wrong with the request, as it might come from a slow connection.

I configured R-U-Dead-Yet[2], also known as RUDY, to use HTTPS and attack the login page of the bank server. RUDY is the HTTPS POST DoS attack in the simulation. A slow POST DoS sends a piece of the request, e.g. a username-field of the login page, slowly every

2. sourceforge.net/projects/r-u-dead-yet

second. RUDY by default sends a single letter "A."

One of the bots sent malicious traffic, and the other two sent normal traffic at any one time, to ensure that the DoS would be disguised within the normal traffic. The Slow HTTPS POST was configured not to consume the whole connection pool.

Listing 7.1. Apache Status: Webserver under normal load

```
Server load: 0.79 0.75 0.46
Total accesses: 4636 - Total
   Traffic: 5.3 MB
CPU Usage: u1.48 s4.51 cu0 cs0 -
   .00653%
CPU load .0505 requests/sec - 61
   B/second - 1209 B/request
5 requests currently being
   processed, 42 idle workers
....C.___._...___._......____.__
........_..._..._.....K_..__...
   ......
____.............._._..._._.___.__
......_..K..__..._._..... W...K_
..........._....
```

Listing 7.2. Apache Status: Webserver during the Slow HTTPS attack

```
Server load: 1.30 0.91 0.57
Total accesses: 5062 - Total
   Traffic: 6.1 MB
CPU Usage: u1.75 s4.94 cu0 cs0 -
   .00728%
CPU load .0551 requests/sec - 70
   B/second - 1272 B/request
150 requests currently being
   processed, 0 idle workers
WWWCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCWCCCCCCCCCCCCCCCCCCCCC
CCCCCC
```

Apache Status (`http://localhost/server-status`) page of the webserver shows how the attack affects the server. Listing 7.1 presents what a normal state looks like. Listing 7.2 shows how the slow HTTPS GET DoS has filled the whole default 150 connection pool. Almost all the connections are in the "C" state, which means that the server tries to close them. The volume of traffic in the network does not change much, but the effect on the server's ability to answer is practically gone.

7.2.3 Sniffing traffic and generating the datasets

During the experiments, the webserver was recording the network traffic with `tcpdump` command below, creating datasets for testing. In the command, `-nn` means no hostname or

port resolving, `-S` sets absolute sequence numbers on and `-s` defines snap length, which is normally 262144 bytes. Setting it to zero captures all data per package.

```
sudo tcpdump -i eth0 -nnS -s0 -w botnet_normal.pcap
```

The detection method could also work in an online mode because of the capabilities discussed in the previous section, but in this case, I only looked at the offline behavior of the detection method. The offline simulations were done by evaluation a PCAP-trace file from the network traffic happening during the attacks in the virtual network. There were three cases that I needed to record. First, the training phase of the method needs normal traffic to create a model of the normal state. The second phase was to record a PCAP-file when the slow HTTPS GET DoS attacks running. The third dataset was created during the slow HTTPS POST denial-of-service attack. I had three PCAP-files to evaluate with the detection method now.

To compare the results with an older dataset, I chose the DARPA1999 dataset. I identified Wednesday 7th of April 1999 inside PCAP-data [3] to contain DoS attacks on the port 80. The data does not contain encrypted HTTP traffic on the port 433. The same metrics can be evaluated from the traffic, even though the port is different. According to the identification scoring truth list, there were two different kinds of attacks during that day to the specified port: a Back DoS and an Apache2 Header DoS. The attackers were 152.204.242.193, 194.27.251.21 and 172.16.117.52. I used Monday the 1st of March inside PCAP as the training data [4]. I used only a part of the DARPA1999 dataset and only two kinds of DoS attacks in the dataset.

As I explained, I do not take the C&C traffic into account in this test, and I removed it from the PCAP-file. Thus, the datasets contained only HTTPS traffic and TCP handshakes. DARPA1999 set contains much traffic with different protocols. I was interested in the trivial DoS attacks, and thus I removed everything else than the HTTP traffic. The modifications to the datasets were done with Wireshark.

3. www.ll.mit.edu/ideval/data/1999/testing/week5/index.html
4. www.ll.mit.edu/ideval/data/1999/training/week1/index.html

7.3 Results of the experiments

In this section, I display the results of the simulations in ROC-curves and comparison tables.

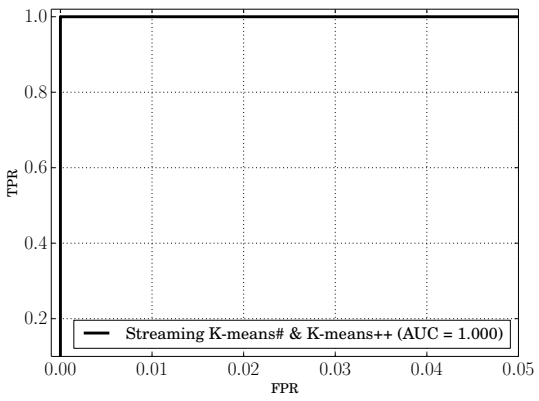

Figure 14. ROC DARPA'99 with K-means# & K-means++

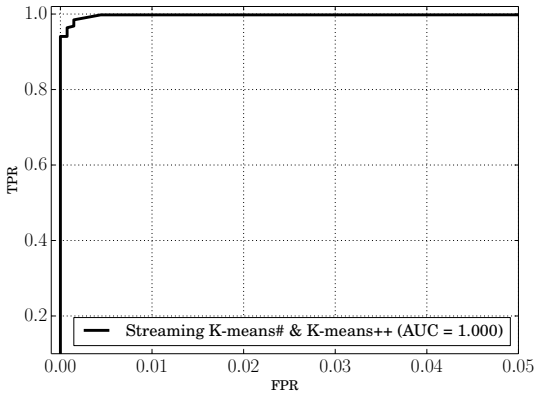

Figure 15. ROC slow HTTPS POST (RUDY) with K-means# & K-means++

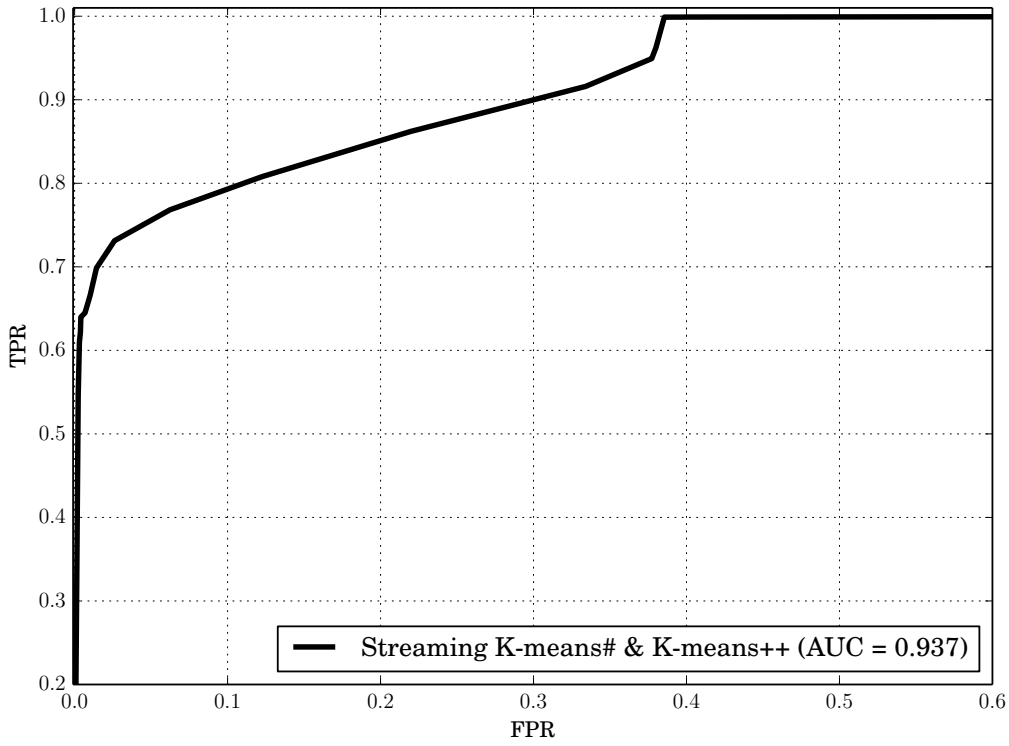

Figure 16. ROC Slowloris with K-means# & K-means++

The ROC-curves for the attacks in the virtual network can be seen in Figures 15 and 16. Both of the attacks were done in a similar manner, i.e. one bot attacking and the rest communicating with the server normally. The slow HTTPS POST attack was much easier to detect from the traffic. Since the slow HTTPS POST attack did not consume the full capacity of the webserver, the normal bots were able to continue normally. In the case of HTTPS GET, I configured the attacker to consume all available connection slots. This led to the normal bots to slow down their communication to the point that their traffic to the server was flagged partially anomalous, thus the high rate of false positives.

DARPA1999 DoS attacks are not very sophisticated in this case, and they differ from the normal traffic to the webserver a lot. Therefore, the K-means++ method turned out to be a perfect classifier, i.e. 100% TPR, and 0% FPR. These kind of trivial denial-of-service attacks are already easy to detect with the current methods. The ROC-curve for the DARPA1999 -dataset can be seen in Figure 14.

Mikhail Zolotukhin also made simulations with other DDoS attacks for comparison, and I am reporting them in this thesis. Also, for the same attacks, it is interesting to compare the results with other studies from the mapping study.

As I discussed in Section 6.2, an effective anomaly-based NIDS has a false positive rate of under 1% . Especially in high-speed traffic, the amount alarms may lead to unwanted pressure for the administrator to make decisions. In contrast, if the true positive rate is too low, the effectiveness of the method is questionable. For this reason, Tables 14 and 15 display the results of the simulations and the comparison results from other studies with less than or close to 1% false positive rate. The Slowloris comparison is interesting, and the difference may be caused by a configuration error as well. The method needs more research.

Table 14. K-means# & K-means++ detection accuracy comparison with other attacks

Experiment	Attack	Accuracy	FPR
My simulations with K-means++ & K-means#	Slowloris	70.222%	1.428%
	Slow POST	99.633%	0.43956%
	DARPA1999 dataset	100%	0%
Zolotukhin (2016)	Slowloris	99.77%	0.8%
	SSLsqueeze	99.89%	0.8%

Table 15. Detection accuracy comparison with other methods

Experiment	Method	Attack	Accuracy	FPR
Zolotukhin et al. (2016a)	DBSCAN	Slowloris	99.99%	0.0045%
	SOM	Slow POST	99.99%	0.0045%
	SOM	An advanced DDoS	96.81%	1.36%
Zolotukhin et al. (2016b)	Fuzzy c-means	Slowloris	99.11%	<1%
	Fuzzy c-means	SSLsqueeze	100%	<1%
	Fuzzy c-means	An advanced DDoS	71.84%	<1%

7.4 Answer to the research question 2

The second research question led to the investigation of one clustering method and its functioning. The second part of the question touches the very nature of the network traffic. The simulation experiment shows how an anomaly-based method works in theory and practice.

RQ2: How do DDoS attack anomaly detection methods work and what are the main issues regarding their performance?

Intrusion detection systems (IDS) usually either compare the traffic to already known patterns (i.e. signature-based model) or classify the traffic to either anomalous or normal traffic classes (i.e. anomaly-based model). As opposed to signature-based mode, the anomaly-based model can detect zero-day attacks, i.e. attacks that have never been seen before. Still,

an attacker who knows about this method can easily avoid detection by designing the attack in a clever, unseen way. On the other hand, the anomaly-based model also has false positives that it needs to tackle. If a normal user gets labeled as an attacker because of their abnormal surfing pattern, it is a problem. When designing anomaly-based IDS, these faults need to be taken into account. (Ingham and Somayaji 2007, 139.)

Anomaly detection works by building a model or relying upon a known feature or a set of features of the network traffic. These features may be the payload, packet statistics or the server load. There are several anomaly detection methods proposed for the DDoS attack detection in encrypted network traffic and many more in non-encrypted traffic. Many of the methods in the latter, could also be used for the former, however. The anomaly detection idea relies on the fact that attacks against the availability of the service are different from normal user behavior in some measure. The measure and the difference in that measure have to be found and analyzed. This notion is also the problem of anomaly detection.

The definition of anomalies in network traffic is problematic. Much of the ordinary traffic may be classified as anomalous because of the properties they have, e.g. Slashdot effect. The qualities of traffic on the Internet are manifold and new applications, protocols and websites come and go all the time. The ever-changing environment increases the difficulty to detect anomalous traffic, as the concept of "normal" evolves all the time. Therefore, the attackers are imitating the traffic patterns that happen naturally and hiding the attacks from the anomaly detectors. (Raghavan and Dawson 2011, 133.)

The complexity of the issues regarding the validity of methods, security of the network users and ever-changing strategies of DDoS attacks beg the question of how difficult this task of detection is. Sommer and Paxson (2010, 314) claim that intrusion anomaly detection is one of the most difficult tasks currently for machine learning and data mining to tackle. The aim of intrusion and the DDoS attack detection is to detect the outliers or problems instead of trying to make sense of the data and create predictions, where machine learning is considered to being good at. The variability of network traffic at various servers and points in the network, as well as the problem of detecting false positives, raise the difficulty.

7.5 Summary

The chapter presented a method and the simulation experiment, including the dataset creation for the DDoS attack detection. The chapter presented the results alongside previous similar results, and they were found to be similar. The second research question was answered.

8 Discussion

The aim of this research was to research the current state of detection methods of encrypted denial-of-service attacks by reviewing the literature and discuss the efficiency of anomaly-based methods, as mentioned in Section 1.2. I discuss the validity of the mapping study in Section 8.1 and the significance of the results of the simulations in Section 8.2. I also return to the results of the mapping study with the results of the simulations in Section 8.3.

8.1 A validity evaluation of the systematic mapping study

The validity section of the mapping study covers the areas highlighted in the updated guidelines by Petersen, Vakkalanka, and Kuzniarz (2015). These are descriptive validity, theoretical validity, generalizability, interpretive validity, and repeatability.

8.1.1 Descriptive validity

I had a data extraction form in Excel and a systematic process during the phase. The form was easy to understand, and I designed it the research question in mind. I took an analytic approach to the process and extracted all the data possible. However, I was the only one conducting the extraction of data alone, which may lead to imperfections and misinterpretation of the content.

According to Kitchenham and Charters (2007, 13), if a researcher is working alone, he or she should present the protocol of the study to the supervisor for inspection and comments. I took care of this part of the guidelines by sending the protocol to my supervisor after I had done the pilot study. I included comments from Mikhail regarding the terms. Otherwise, the protocol was understandable according to my supervisor, and I could proceed with the mapping study.

8.1.2 Theoretical validity

I did not research how the mapping studies are done in other fields but trusted that the mapping study guidelines in software engineering according to Petersen, Vakkalanka, and Kuzniarz (2015) are valid and up to date. Many of the ideas and either came from the papers by Petersen, Vakkalanka, and Kuzniarz (2015), Kitchenham and Charters (2007) or Kaijanaho (2015). Using several sources may have led to inconsistencies in the process. However, I believe that this is not the case, but rather I have combined the practices from each source to this study. I did my best to get the consensus these guidelines have and implement it to my thesis to the best of my knowledge.

I kept the process as transparent by disclosing the protocol (see Appendix A), process, problems (see sub-sections of Section 6.3) and the full list of papers after the inclusion criteria (see Appendix B). These are the papers that were taken into closer inspection but were left out of the study. There may be mistakes because I was the only one who read and evaluated these papers. I listed the exclusion reasons for each article carefully, and I am confident that they are correct. This practice was proposed by Petersen, Vakkalanka, and Kuzniarz (2015, 14). It is up to the reader to decide whether the exclusion reasons hold and he or she can verify the validity of the mapping study.

By using the databases and the search terms as mentioned earlier, I limited the search results intentionally. I found the paper by Zolotukhin et al. (2015) that was supposed to be included in the search. The lack of knowledge of possible other articles may have limited the results and is a minor limitation of the study. Identifying two or more candidate papers before the search must be done for future research.

Note that I also searched from Google Scholar and decided to leave the search results out, because I could not limit the result set to even under 100 results. Reading would have turned out to be too time-consuming for this time scale. With a quick look, most of the publications were already in the study. I also saw multiple results of M.Sc. theses and other technical documents that were included in the search results but would have been discarded in the exclusion phase. Nevertheless, this decision to limit the search might have caused some important papers to be missed. If I did the same study with more time and people, Google

Scholar, more databases, manual search and snowball search would be included in the search strategies to include more points of view.

Table 6 shows that all the databases were important as the overall contribution of each source was 20% or more. Leaving out either one of them would have lessened the accuracy of the results. The level of contribution from all the sources works as an assurance of the quality of the results.

None of the search phrases is especially specific in finding what I was looking for, and only that. I made a mistake in the search term formulation phase. I should have included "intrusions" and added some limits for certain terms such as the "WSN" and the "MANET" that I decided to leave out. I could have increased the specificity of the searches and reduced the amount of reading in the first inclusion phase greatly.

8.1.3 Generalizability

I performed the mapping study in most common and suggested databases of the computer science literature in DDoS attacks. See Section 2.1 for the suggested list and Section 6.3 for a list that I used. Because of time constraints, few studies may have been missed, reducing the generalizability to the whole research area. Classification of the DDoS attack detection methods gathered by Mirkovic and Reiher (2004) and Patcha and Park (2007) are closely related to the results that this mapping study found. Therefore, I argue that the results give an idea of the research field and are generalizable because the papers were found from various recommended sources.

8.1.4 Interpretive validity

Research bias is characterized by designing and implementing experiments that have the likelihood to produce a result that is favorable or otherwise more desirable to the researchers. Research bias happens in all phases of the process. Even if the design of the study is unbiased, the selection, data extraction, and the synthesis stages may distort the results. (Pannucci and Wilkins 2010, 1.)

Research bias may play a role in summarizing and synthesizing methods from the papers. Because of the number of methods, I explain the classification decision of all included methods in the text (see Section 6.4.2). This way, possible mistakes can be found or results confirmed. Methods presented in the papers were reported very differently. I had to extract the information with care, and I am confident that the classification decisions are correct.

8.1.5 Repeatability

To address the repeatability concerns of the study, as notes and an example for myself and anyone who is interesting in mapping studies, I recorded the whole process as precisely as I could. I put the protocol with all the changes as an Appendix A. The process is described in Chapter 6 starting from the classification schemes of the studies, search terms, the process itself and finally the synthesis. Following the process, with its imperfections, I argue that it is possible to reach the same conclusions.

8.1.6 Research bias and confidence in results

I minimized research bias by conducting a systematic mapping study and searching for possible keywords to expand the search. In contrast, if I had done the literature review in an unsystematic way or only a snowball search round, the results would have looked different. Using this method, I was able to avoid identical authors problem (Jalali and Wohlin 2012, 36).

The results of the mapping study can be seen as skewed because four out of 14 (see Section 6.4.1) studies are written by my supervisors. The presumption that there are not many encrypted DDoS attack detection methods in the literature was the basis for the first research question. The wording of the question directly influenced the keywords and search queries. I asked for comments on the search terms from Mikhail Zolotukhin as an expert opinion.

I conducted the mapping study and reported the process in a neutral way. This particular research group is one of few, and that is the reason why the results tend to favor these particular papers. Since the results of the mapping study follow the assumption of few studies in the encrypted DDoS attack research (see Section 1.2), I argue that the results are correct.

8.2 Limitations of the simulations

The purpose of the simulation experiments was to explain the selected method in detail and detect application layer DDoS attacks in an SSL/TLS encrypted network traffic and discuss the accuracy issues related to anomaly-based methods. Nevertheless, the limitations of the setup are apparent. First, the size of the botnet is significantly smaller than nearly any network where such an NIDS would be implemented. As Sommer and Paxson (2010, 312) argue, smaller networks rarely produce results that are completely applicable to bigger networks. Second, the traffic that is considered normal is created by a bot program. Human traffic is never that predictable, and that is one of the reasons I got inconsistent results with the HTTPS GET attack when the normal traffic was mainly considered as anomalous as well. The inconsistency with the results happened because the connection times of the "normal" bots got longer as the connection pool filled. Third, I did not test the online capabilities of the method, i.e. the possibility to detect a live DDoS attack. Regardless of these clear limitations of the simulations, the results are in line with the previous research, and they give an idea of the problems of the DDoS detection research.

8.3 Discussion on the results of the thesis

As I said in the introduction, the numbers of application layer and encrypted DDoS attacks are growing. Research in this area is important for these reasons. The mapping study found that the majority of the research focuses on anomaly detection, which is logical considering the inability to read the payload.

Datasets vary greatly, and that tells about the difficulty of producing reliable datasets. DARPA1999 and KDD'99 datasets are still being used for comparison reasons. They, however, include only trivial flood attacks which are easily detectable with the current anomaly-based methods.

As I saw the number of times the DARPA-based datasets were used, I also downloaded a sample of it to see how the K-means++ method would do. The method detected all the malicious flows with ease. However, the proof of the accuracy of the method from those tests cannot be trusted considering the trivial attacks in the datasets. Also, the virtual network

datasets are not ideal, because the normal data is generated by a bot and it is too similar to a malicious attack when the server is not able to answer anymore. The slow HTTPS POST attack likely was easier to detect for that very reason. Because of the datasets I used, the only contribution of the simulations is the comparison with the DARPA-dataset and comparable results with the previous tests what Zolotukhin et al. (2015) made.

The main contribution of this thesis, therefore, is in the results of the mapping study. The results show that there are gaps in the research. Figure 11 shows that system call sequence analysis, Bayesian networks, PCA, Markov models, classification and association rules are underrepresented. These same methods are used in non-encrypted research.

9 Conclusion

DDoS attacks are becoming larger and more disguised. This thesis explored the research into DDoS attacks in encrypted network traffic because it was not clear how much and what kind of methods exist according to Zolotukhin et al. (2015). The method used for literature review was a systematic mapping study. To study the functioning of the anomaly detection methods, I experimented with a clustering-based method and conducted simulations for trivial application layer DDoS attacks that were created in a controlled virtual network environment.

Based on the results of the mapping study, I conclude that there exist only ten papers on the topic and four additional methods that can detect encrypted DDoS without experimenting with it. The methods presented have concentrated in statistical and clustering methods. The prevalence of statistical methods can be explained by the lack of access to the payload features to distinguish DDoS attacks from normal traffic. The methods use various metrics for detecting DDoS attacks because of the limitation, flow statistics and packet header information being the most prevalent ones. The identified gaps in research methods were system call sequence analysis, Bayesian networks, PCA, Markov models, classification, and association rules.

In the simulation experiment, I identified that the K-means++ clustering method detects with near 100% accuracy trivial application layer attacks despite a lower result for a slow HTTP GET attack. The accuracy was near 70% with low values of false positives. The slow HTTPS GET result shows, however, how the K-means++ clustering method classifies the legitimate traffic as anomalous because it is similar to the attack traffic. The reasons for the similarity also lie in the setup and normal traffic being also generated rather than real. The same concept applies when a more advanced DDoS attack is compared to a real human-generated traffic.

The limitations of the mapping study restrict the results. Even though the results show that there are gaps in the research compared to non-encrypted methods, there are resources (such as Web of Science, Google Scholar, snowball searching and gray literature) that could change this assumption. For this reason, a more thorough mapping study of anomaly-based DDoS

detection methods in encrypted network traffic would be an excellent contribution to the research area. Based on a comprehensive map, further systematic reviews could be conducted to draw conclusions on the state of the existing methods. As more and more traffic gets encrypted and DDoS attacks are changing to more advanced ones, this becomes even more important to study.

Bibliography

928 F.2d 504. 1991. *US v. Morris.* Visited on September 18, 2016. https://scholar.google.com/scholar_case?case=551386241451639668.

Abdi, Hervé, and Lynne J Williams. 2010. "Principal component analysis". *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (4): 433–459.

Abimbola, Abiola, Qi Shi, and Madjid Merabti. 2003. "Nethost-sensor: a novel concept in intrusion detection systems". In *Computers and Communication, 2003.(ISCC 2003). Proceedings. Eighth IEEE International Symposium on,* 232–237. IEEE.

Aiello, M., E. Cambiaso, M. Mongelli, and G. Papaleo. 2014. *An on-line intrusion detection approach to identify low-rate DoS attacks.* doi:10.1109/CCST.2014.6987039.

Ailon, Nir, Ragesh Jaiswal, and Claire Monteleoni. 2009. "Streaming k-means approximation". In *Advances in Neural Information Processing Systems,* 10–18.

Amoli, Payam Vahdani, and Timo Hämäläinen. 2013. "A real time unsupervised NIDS for detecting unknown and encrypted network attacks in high speed network". In *Measurements and Networking Proceedings (M&N), 2013 IEEE International Workshop on,* 149–154. IEEE.

Arbor Networks. 2011. *Worldwide infrastructure security report.* Volume VII. Arbor Networks Inc.

———. 2013. *Worldwide infrastructure security report.* Volume IX. Arbor Networks Inc.

———. 2015. *Worldwide infrastructure security report.* Volume X. Arbor Networks Inc.

———. 2016. *Worldwide infrastructure security report.* Volume XI. Arbor Networks Inc.

Arthur, David, and Sergei Vassilvitskii. 2007. "k-means++: The advantages of careful seeding". In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms,* 1027–1035. Society for Industrial and Applied Mathematics.

AsSadhan, Basil, José MF Moura, David Lapsley, Christine Jones, and W Timothy Strayer. 2009. "Detecting botnets using command and control traffic". In *Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on,* 156–162. IEEE.

Bégin, François. 2011. *BYOB: Build Your Own Botnet and learn how to mitigate the threat posed by botnets.* SANS Institute. https://www.sans.org/reading-room/whitepapers/covert/byob-build-botnet-33729.

Birrell, Andrew D. 1985. "Secure communication using remote procedure calls". *ACM Transactions on Computer Systems* 3 (1): 1–14. ISSN: 0734-2071. doi:10.1145/214451.214452.

Blank, Andrew G. 2006. *TCP/IP Foundations.* John Wiley & Sons.

Botnet Carna. 2013. "Internet census 2012—port scanning/0 using insecure embedded devices". http://internetcensus2012.bitbucket.org/paper.html.

Bradley, Andrew P. 1997. "The use of the area under the ROC curve in the evaluation of machine learning algorithms". *Pattern Recognition* 30 (7): 1145–1159. ISSN: 0031-3203. doi:10.1016/s0031-3203(96)00142-2.

Braverman, Vladimir, Adam Meyerson, Rafail Ostrovsky, Alan Roytman, Michael Shindler, and Brian Tagiku. 2011. "Streaming k-means on well-clusterable data". In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms,* 26–40. SIAM.

Brereton, Pearl, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. 2007. "Lessons from applying the systematic literature review process within the software engineering domain". *Journal of systems and software* 80 (4): 571–583.

Caulkins, Bruce D, Joohan Lee, and Morgan Wang. 2005. "A dynamic data mining technique for intrusion detection systems". In *Proceedings of the 43rd annual Southeast regional conference-Volume 2,* 148–153. ACM.

US-CERT. 2013. "Security Tip (ST04-015): Understanding Denial-of-Service Attacks". Visited on September 17, 2016. https://www.us-cert.gov/ncas/tips/ST04-015.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. 2009. "Anomaly detection: A survey". *ACM computing surveys (CSUR)* 41 (3): 15.

Chen, Lianping, Muhammad Ali Babar, and He Zhang. 2010. "Towards an evidence-based understanding of electronic data sources". In *In Proc. 14th International Conference on Evaluation and Assessment in Software Engineering (EASE).*

Chen, Zhongqiang, Zhongrong Chen, and Alex Delis. 2007. "An inline detection and prevention framework for distributed denial of service attacks". *The Computer Journal* 50 (1): 7–40.

Cherdantseva, Yulia, and Jeremy Hilton. 2013. "A reference model of information assurance & security". *2013 International Conference on Availability, Reliability and Security.* doi:10.1109/ares.2013.72.

Claise, Benoit. 2008. *RCF5101 Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information.* Technical report.

Coull, Scott E, Fabian Monrose, Michael K Reiter, and Michael Bailey. 2009. "The challenges of effectively anonymizing network data". In *Conference For Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology,* 230–236. IEEE.

da Silva, Carlo Marcelo Revoredo, Jose Lutiano Costa da Silva, Ricardo Batista Rodrigues, Leandro Marques do Nascimento, and Vinicius Cardoso Garcia. 2013. "Systematic mapping study on security threats in cloud computing". *(IJCSIS) International Journal of Computer Science and Information Security* 11 (3).

Das, Debasish, Utpal Sharma, and DK Bhattacharyya. 2011. "Detection of HTTP flooding attacks in multiple scenarios". In *Proceedings of the 2011 International Conference on Communication, Computing & Security,* 517–522. ACM.

Dupont, Benoît, Anne-Marie Côté, Claire Savine, and David Décary-Hétu. 2016. "The ecology of trust among hackers". *Global Crime* 17 (2): 129–151.

Durcekova, V., L. Schwartz, and N. Shahmehri. 2012. *Sophisticated Denial of Service attacks aimed at application layer.* doi:10.1109/ELEKTRO.2012.6225571.

Dybå, Tore, Torgeir Dingsøyr, and Geir Kjetil Hanssen. 2007. "Applying Systematic Reviews to Diverse Study Types: An Experience Report." In *ESEM,* 7:225–234.

Eliseev, Vladimir, and Anastasiya Gurina. 2016. "Algorithms for network server anomaly behavior detection without traffic content inspection". In *Proceedings of the 9th International Conference on Security of Information and Networks,* 67–71. ACM.

Fawcett, Tom. 2006. "An introduction to ROC analysis". *Pattern Recognition Letters* 27 (8): 861–874. ISSN: 0167-8655. doi:10.1016/j.patrec.2005.10.010.

Fellows, D., and D. Jones. 2001. "DOCSISTM cable modem technology". *IEEE Communications Magazine* 39 (3): 202–209. ISSN: 0163-6804. doi:10.1109/35.910608.

Ferguson, D., R. Clouston, and A. Talerico. 2003. *Method and apparatus for SNA/IP correlation with multiple DSW peer connections.* US Patent 6,571,272. https://www.google.com/patents/US6571272.

Friedman, Nir, Dan Geiger, and Moises Goldszmidt. 1997. "Bayesian network classifiers". *Machine learning* 29 (2-3): 131–163.

Fürnkranz, Johannes, Dragan Gamberger, and Nada Lavrač. 2012. *Foundations of rule learning*. Springer Science & Business Media.

Garber, Lee. 2000. "Denial-of-service attacks rip the Internet". *IEEE Computer* 33 (4): 12–17.

Grira, Nizar, Michel Crucianu, and Nozha Boujemaa. 2004. "Unsupervised and semi-supervised clustering: a brief survey". *A review of machine learning techniques for processing multimedia content* 1:9–16.

Gutiérrez, Sergio Armando, and John Willian Branch. 2014. "Application of Machine Learning Techniques to Distributed Denial of Service (DDoS) Attack Detection: A Systematic Literature Review." *Revista NOOS* 4.

Han, Jiawei, Jian Pei, and Micheline Kamber. 2011. *Data mining: concepts and techniques.* Elsevier.

Hankey, James A. 1989. ”Receiver operating Characteristic (ROC) Methodology: A State of the Art”. *Crit Rev Diagn Imaging 1989* 29 (3): 307–335.

Harris, B, and R Hunt. 1999. ”TCP/IP security threats and attack methods”. *Computer Communications* 22 (10): 885–897.

Hawkins, Simon, Hongxing He, Graham Williams, and Rohan Baxter. 2002. ”Outlier detection using replicator neural networks”. In *International Conference on Data Warehousing and Knowledge Discovery,* 170–180. Springer.

He, Yaobin, Haoyu Tan, Wuman Luo, Shengzhong Feng, and Jianping Fan. 2013. ”MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data”. *Frontiers of Computer Science* 8 (1): 83–99. doi:10.1007/s11704-013-3158-3.

Hearst, Marti A., Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. 1998. ”Support vector machines”. *IEEE Intelligent Systems and their Applications* 13 (4): 18–28.

Hoeve, Maarten. 2013. ”Detecting intrusions in encrypted control traffic”. In *Proceedings of the first ACM workshop on Smart energy grid security,* 23–28. ACM.

Hu, Wenjie, Yihua Liao, and V Rao Vemuri. 2003. ”Robust Support Vector Machines for Anomaly Detection in Computer Security.” In *ICMLA,* 168–174.

Ingham, Kenneth L, and Anil Somayaji. 2007. ”A methodology for designing accurate anomaly detection systems”. In *Proceedings of the 4th international IFIP/ACM Latin American conference on Networking,* 139–143. ACM.

ISO/IEC 27000. 2016. *ISO/IEC 27000:2016(en) Information technology - Security techniques - Information security management systems - Overview and vocabulary.* Visited on August 2, 2016. https://www.iso.org/obp/ui/#iso:std:iso-iec:27000:ed-4:v1:en:term:2.21.

ISO/IEC 27002. 2013. *ISO/IEC 27002:2013(en) Information technology - Security techniques - Code of practice for information security controls.* Visited on August 2, 2016. https://www.iso.org/obp/ui/#iso:std:iso-iec:27002:ed-2:v1:en.

ISO/IEC 27032. 2012. *ISO/IEC 27032:2012(en) Information technology - Security techniques - Guidelines for cybersecurity.* Visited on August 2, 2016. https://www.iso.org/obp/ui/#iso:std:iso-iec:27032:ed-1:v1:en.

Jaffee, Larry. 2016. "Waiting for DDoS". Visited on September 17, 2016. http://www.scmagazine.com/waiting-for-ddos/article/523247/.

Jalali, Samireh, and Claes Wohlin. 2012. "Systematic literature studies: database searches vs. backward snowballing". In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement,* 29–38. ACM.

Jarvinen, PH. 2000. "Research questions guiding selection of an appropriate research method". *ECIS 2000 Proceedings:*26.

Jeon, Byeungwoo, and David A Landgrebe. 1999. "Partially supervised classification using weighted unsupervised clustering". *IEEE Transactions on Geoscience and Remote Sensing* 37 (2): 1073–1079.

Kaijanaho, Antti-Juhani. 2015. "Evidence-based programming language design : a philosophical and methodological exploration". Visited on September 24, 2016. http://urn.fi/URN:ISBN:978-951-39-6388-0.

Kitchenham, Barbara, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. "Systematic literature reviews in software engineering–a systematic literature review". *Information and software technology* 51 (1): 7–15.

Kitchenham, Barbara, and Stuart Charters. 2007. *Guidelines for performing systematic literature reviews in software engineering.*

Kokkonen, Tero, Timo Hämäläinen, Marko Silokunnas, Jarmo Siltanen, Mikhail Zolotukhin, and Mikko Neijonen. 2015. "Analysis of Approaches to Internet Traffic Generation for Cyber Security Research and Exercise". In *Conference on Smart Spaces,* 254–267. Springer.

Krawetz, Neal. 2007. *Introduction to network security.* United States: Charles River Media. ISBN: 9781584506430.

Krebs, Brian. 2016. ”KrebsOnSecurity hit with record DDoS”. *KrebsOnSecurity.* Visited on October 31, 2016. https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/.

Latif, Rabia, Haider Abbas, and Saïd Assar. 2014. ”Distributed denial of service (DDoS) attack in cloud-assisted wireless body area networks: a systematic literature review”. *Journal of medical systems* 38 (11): 1–10.

Lee, Keunsoo, Juhyun Kim, Ki Hoon Kwon, Younggoo Han, and Sehun Kim. 2008. ”DDoS attack detection method using cluster analysis”. *Expert Systems with Applications* 34 (3): 1659–1665.

Levillain, Olivier, Arnaud Ebalard, Benjamin Morin, and Herve Debar. 2012. *One Year of SSL Internet Measurement.* Orlando, Florida, USA. doi:10.1145/2420950.2420953.

Lin, Shun-Chieh, and Shian-Shyong Tseng. 2004. ”Constructing detection knowledge for DDoS intrusion tolerance”. *Expert Systems with applications* 27 (3): 379–390.

Linge, N, M Hope, et al. 2007. ”Active router approach to defeating denial-of-service attacks in networks”. *IET communications* 1 (1): 55–63.

Madhavi, SaniKommu. 2008. ”An intrusion detection system in mobile adhoc networks”. In *Information Security and Assurance, 2008. ISA 2008. International Conference on,* 7–14. IEEE.

Mahoney, Matthew V, and Philip K Chan. 2003. ”An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection”. In *International Workshop on Recent Advances in Intrusion Detection,* 220–237. Springer.

Mansfield-Devine, Steve. 2011. ”Anonymous: serious threat or mere annoyance?” *Network Security* 2011 (1): 4–10.

Mirkovic, Jelena, and Peter Reiher. 2004. ”A Taxonomy of DDoS Attack and DDoS Defense Mechanisms”. *SIGCOMM Comput. Commun. Rev.* (New York, NY, USA) 34 (2): 39–53. doi:10.1145/997150.997156.

Mitchell, Tom M. 1997. ”Machine learning. 1997”. *Burr Ridge, IL: McGraw Hill* 45:37.

Mouli, Varsha R, and KP Jevitha. 2016. ”Web Services Attacks and Security-A Systematic Literature Review”. *Procedia Computer Science* 93:870–877.

Nagaratna, M, V Kamakshi Prasad, and S Tanuz Kumar. 2009. ”Detecting and preventing IP-spoofed DDOS attacks by Encrypted Marking Based Detection and Filtering (EMDAF)”. In *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on,* 753–755. IEEE.

”HE 153/2006”. 2006. Oikeusministeriö. Visited on September 17, 2016. `http://www.finlex.fi/fi/esitykset/he/2006/20060153`.

Pa, Yin Minn Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. 2015. ”IoTPOT: Analysing the Rise of IoT Compromises”. In *9th USENIX Workshop on Offensive Technologies (WOOT 15).* Washington, D.C.: USENIX Association.

Pannucci, Christopher J, and Edwin G Wilkins. 2010. ”Identifying and avoiding bias in research”. *Plastic and reconstructive surgery* 126 (2): 619.

Patcha, Animesh, and Jung-Min Park. 2007. ”An overview of anomaly detection techniques: Existing solutions and latest technological trends”. *Computer networks* 51 (12): 3448–3470.

Petersen, Kai, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. ”Systematic mapping studies in software engineering”. In *12th international conference on evaluation and assessment in software engineering,* volume 17. 1. sn.

Petersen, Kai, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. ”Guidelines for conducting systematic mapping studies in software engineering: An update”. *Information and Software Technology* 64:1–18. ISSN: 0950-5849. doi:`10.1016/j.infsof.2015.03.007`.

Petiz, Ivo, Paulo Salvador, António Nogueira, and Eduardo Rocha. 2014. ”Detecting DDoS attacks at the source using multiscaling analysis”. In *Telecommunications Network Strategy and Planning Symposium (Networks), 2014 16th International,* 1–5. IEEE.

Raghavan, S V, and E Dawson. 2011. *An investigation into the detection and mitigation of denial of service (DoS) attacks critical information infrastructure protection.* New Delhi: Springer India Pvt. ISBN: 9788132202776.

Rastegari, Samaneh, Philip Hingston, and Chiou-Peng Lam. 2015. ”Evolving statistical rulesets for network intrusion detection”. *Applied Soft Computing* 33:348–359.

RFC706. 1975. ”On the Junk Mail Problem”. Visited on September 18, 2016. `https://tools.ietf.org/html/rfc706`.

Saltzer, J.H., and M.D. Schroeder. 1975. ”The protection of information in computer systems”. *Proceedings of the IEEE* 63 (9): 1278–1308. ISSN: 0018-9219. doi:`10.1109/proc.1975.9939`.

Schneider, Fred. 1999. *Trust in Cyberspace.* Volume ISBN 0-309-06558-5. Washington, D.C.: The National Academies Press. doi:`10.17226/6161`.

Sedjelmaci, Hichem, and Sidi Mohammed Senouci. 2014. ”A lightweight hybrid security framework for wireless sensor networks”. In *2014 IEEE International Conference on Communications (ICC),* 3636–3641. IEEE.

Shanthi, K, and D Seenivasan. 2015. ”Detection of botnet by analyzing network traffic flow characteristics using open source tools”. In *Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on,* 1–5. IEEE.

Shiaeles, Stavros N, Vasilios Katos, Alexandros S Karakos, and Basil K Papadopoulos. 2012. ”Real time DDoS detection using fuzzy estimators”. *Computers & Security* 31 (6): 782–790.

Shiravi, Ali, Hadi Shiravi, Mahbod Tavallaee, and Ali A Ghorbani. 2012. ”Toward developing a systematic approach to generate benchmark datasets for intrusion detection”. *Computers & Security* 31 (3): 357–374.

Silva, SéRgio SC, Rodrigo MP Silva, Raquel CG Pinto, and Ronaldo M Salles. 2013. ”Botnets: A survey”. *Computer Networks* 57 (2): 378–403.

Sommer, Robin, and Vern Paxson. 2010. ”Outside the closed world: On using machine learning for network intrusion detection”. In *2010 IEEE symposium on security and privacy,* 305–316. IEEE.

Sourav, Kumar, and Debi Prasad Mishra. 2012. ”DDoS detection and defense: client termination approach”. In *Proceedings of the CUBE International Information Technology Conference,* 749–752. ACM.

Spackman, Kent A. 1989. ”Signal Detection Theory: Valuable Tools for Evaluating Inductive Learning”. In *Proceedings of the Sixth International Workshop on Machine Learning,* 160–163. Ithaca, New York, USA: Morgan Kaufmann Publishers Inc. ISBN: 1-55860-036-1.

Stern, Henry. 2009. ”The rise and fall of reactor Mailer”. *Proc. MIT Spam Conference 2009.* Visited on July 3, 2016. http://projects.csail.mit.edu/spamconf/SC2009/Henry_Stern/.

Sun, Jimeng, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos. 2005. ”Relevance search and anomaly detection in bipartite graphs”. *ACM SIGKDD Explorations Newsletter* 7 (2): 48–55.

Tama, Bayu Adhi, and Kyung-Hyune Rhee. 2015. ”Data Mining Techniques in DoS/DDoS Attack Detection: A Literature Review”. *International Information Institute (Tokyo). Information* 18 (8): 3739–3747.

Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali-A Ghorbani. 2009. ”A detailed analysis of the KDD CUP 99 data set”. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009.*

Tirthani, Neha, and Ganesan R. 2013. ”Data security in cloud architecture based on Diffie Hellman and elliptical curve Cryptography”. *IACR Cryptology ePrint Archive, 2014* 49. doi:10.1.1.644.4623.

van Erkel, Arian R, and Peter M.Th Pattynama. 1998. ”Receiver operating characteristic (ROC) analysis: Basic principles and applications in radiology”. *European Journal of Radiology* 27 (2): 88–94. doi:10.1016/S0720-048X(97)00157-5.

Viaene, Stijn, Richard Derrig, and Guido Dedene. 2002. ”Boosting naive Bayes for claim fraud diagnosis”. In *International Conference on Data Warehousing and Knowledge Discovery,* 202–211. Springer.

von Solms, Rossouw, and Johan van Niekerk. 2013. ”From information security to cyber security”. *Computers & Security* 38:97–102. ISSN: 0167-4048. doi:10.1016/j.cose.2013.04.004.

Wang, Bing, Yao Zheng, Wenjing Lou, and Y Thomas Hou. 2015. ”DDoS attack protection in the era of cloud computing and software-defined networking”. *Computer Networks* 81:308–319.

Whitman, Michael E., and Herbert J. Mattord. 2011. *Principles of information security.* 4th. Cengage Learning. ISBN: 978-1-111-13821-9.

Zapata, Belén Cruz, José Luis Fernández Alemán, and Ambrosio Toval. 2015. ”Security in cloud computing: A mapping study”. *COMSIS Computer Science and Information Systems* 12 (1): 161–184.

Zelkowitz, Marvin V, and Dolores R. Wallace. 1998. ”Experimental models for validating technology”. *Computer* 31 (5): 23–31.

Zolotukhin, Mikhail. 2016. *Detection of trivial DDoS attacks with streaming k-means.* Unpublished simulations.

Zolotukhin, Mikhail, Timo Hämäläinen, Tero Kokkonen, Antti Nieme ä, and Jarmo Siltanen. 2015. ”Internet of Things, Smart Spaces, and Next Generation Networks and Systems: 15th International Conference, NEW2AN 2015, and 8th Conference, ruSMART 2015, St. Petersburg, Russia, August 26-28, 2015, Proceedings”. Edited by Sergey Balandin, Sergey Andreev, and Yevgeni Koucheryavy. (Cham):274–285. doi:10.1007/978-3-319-23126-6_25.

Zolotukhin, Mikhail, Timo Hämäläinen, Tero Kokkonen, and Jarmo Siltanen. 2016a. ”Increasing web service availability by detecting application-layer DDoS attacks in encrypted traffic”. In *Telecommunications (ICT), 2016 23rd International Conference on,* 1–6. IEEE.

———. 2016b. ”Weighted Fuzzy Clustering for Online Detection of Application DDoS Attacks in Encrypted Network Traffic”. In *International Conference on Next Generation Wired/Wireless Networking,* 326–338. Springer.

Appendices

A Systematic mapping study protocol

Version: 16.10.13

1 Change record

Table 16. Change record

Date	Changes
23.9.2016	The first draft
26.9.2016	Add extraction features from each study
3.10.2016	Change time spam of the search, change the outlook of the protocol to Markdown style in Git, refine the inclusion and exclusion criteria based on the initial pilot search, drop the PICO method because the guidelines say that only P and I can be used in mapping studies
5.10.2016	Change EC
6.10.2016	Change EC slightly to exclude non-encrypted protocols
8.10.2016	Fix a typo in RQ
12.10.2016	Add change record, scope, expert opinion, roles, synthesis, data ext. method, limitations, reporting and schedule. Fix IC and EC, data collection and quality criteria
13.10.2016	Add the expert opinion

2 Scope of the study

- There are no mapping studies for DDoS detection methods in encrypted traffic
- Previous systematic literature reviews do not address this part, therefore an overview of the research is needed to understand the current state of the field of study

2.1 Search strategy

Electronic online databases are searched. Because of a lack of time, no manual search or snowball methods are done.

- IEEE Xplore (ieeexplore.ieee.org)
- ACM Digital Library (dl.acm.org)
- Google Scholar (scholar.google.com)
- Scopus (www.scopus.com)
- ScienceDirect (www.sciencedirect.com)

I ruled out a manual search because of the amount of time available. A large set of online databases was chosen because I expected to find only a few methods.

2.2 Publication date limit

- Search time span: until 2016 September

2.3 Research questions

- RQ1: What methods for detecting encrypted DDoS attacks are presented in the scientific literature?

2.4 Search terms

- DDoS attacks
- encrypted network traffic
- detection methods

2.5 Search term formulation

- "DoS" OR "DDoS" OR "denial of service" OR "denial-of-service"
- "detection method" OR "detection methods" OR "detecting" OR "detection" OR "detect"
- "encrypted" OR "SSL" OR "TLS" OR "HTTPS"

2.6 Expert opinions

- I contacted Mikhail Zolotukhin to get comments about the search terms. He noted that they seem the only option.

2.7 Thesaurus check

- I checked Thesaurus.com, MOT Collins Compact Thesaurus, Oxford Thesaurus of British English:
  - (1) detect (verb)
  - (2) encrypt (verb)

The word "attack" was dropped since the term denial-of-service implies an attack in the literature. If this would not be the case, the paper could easily be rejected. The word "network traffic" was also dropped for the same reason. The main body of literature focuses on DDoS attacks coming from the network, not caused by other means such as physical threats.

Also the terms "Secure Socket Layer" and "Transport Layer Security" were considered, but the paper which would include these would also include the abbreviations. The term "scientific literature" in the RQ1 limits the search to the literature which has been published in a journal, conference proceedings, a technical paper or theses. It rules out e.g. patents and gray literature. The thesaurus check did not yield additional terms, but I added the various forms of "detect" and "encrypt."

2.8 Article that should be found

- Zolotukhin, M., Hämäläinen, T., Kokkonen, T., Niemelä, A., and Siltanen, J. (2015). Data Mining Approach for Detection of DDoS Attacks Utilizing SSL/TLS Protocol. Conference on Smart Spaces.

3 Roles

- One person (the author) conducts the entire mapping study

4 Screening process

4.1 Inclusion criteria

- IC1: Papers that present a DDoS detection method
- IC2: Papers which were published until 2016 September
- IC2: Papers written in English
- IC4: Papers which I can access in full text (JYU has access or the article is free)
- IC5: Papers where the detection method could be applied to the TCP/IP network

I decided to exclude DoS papers that were about MANETs or WSNs as they were not directly applicable to the detection methods in the Internet

4.2 Exclusion criteria

- EC1: Papers which do not present results, but merely hypothesize of a detection method
- EC2: Papers which are already included from another source
- EC3: Papers which present a detection method based on packet payload analysis
- EC4: Gray literature, lecture notes, and books

5 Data extraction and mapping

5.1 Quality criteria

- Do the authors acknowledge the efficacy of the method for encrypted traffic? 1 or 0

The quality criteria were implemented to include detection methods that are not based on payload inspection, and they could work as detection methods for encrypted traffic.

5.2 Data to be collected

- Title of the paper
- Authors of the paper
- Source: conference/journal
- Year of publication
- Name of the method for detection (algorithm, tool or set of steps)
- Strategy/category of detection method

- Sample DDoS attack used in the paper
- Example dataset used in the research
- Quality score of the paper

If several methods are presented in one paper, both methods should be listed separately in the results.

5.3 Data collection method

- An Excel spreadsheet is used to collect all the information
- In case there is an issue, the information has to be double checked

6 Synthesis

- The years of publication are reported in a bar chart
- All the strategies and the methods are listed in a table
- Example data and sample DDoS attacks are shown in a table
- The methods, publication year and number of studies are presented in a bubble plot
- The tables, graphs, and plot are analyzed for a conclusion about what are the detection methods in the scientific literature to detect DDoS attacks in encrypted traffic

7 Limitations of the study

- The author is alone with the search, inclusion and data extraction phases which might lead to mistakes
- In the search phase, a random sample has to be re-evaluated to reduce errors and researcher bias
- Only one example paper is the basis of the research and might cause skewed results

8 Reporting

- The study is a chapter in a master's thesis

9 Schedule

- Start of the mapping study 23.09.2016
- Pilot search 29.09.2016

- Pilot inclusion and extraction 03.10.2016 - 09.10.2016
- Query for comments 12-13.10.2016
- Search 13-14.10.2016
- Inclusion and extraction 15-19.10.2016
- Synthesis and reporting 19-23.10.2016

B Excluded studies after inclusion criteria

Tables 17, 18, 19 and 20 show the full list of studies after inclusion criteria, and their reasons for exclusion (based on the exclusion criteria). The studies that were included in the mapping study are marked with a dot character (•) and their font style is *italics*. The studies included in the random retest of the screening phase (see Section 6.3.4) are marked with a circle character (○).

Table 17. Scopus studies after inclusion criteria

	Authors	Title	Exclusion reason
1	• *Zolotukhin et al. (2016)*	*Increasing Web Service Availability by Detecting Application-Layer DDoS Attacks in Encrypted Traffic*	
2	• *Zolotukhin et al. (2016)*	*Weighted Fuzzy Clustering for Online Detection of Application DDoS Attacks in Encrypted Network Traffic*	
3	• *Zolotukhin et al. (2015)*	*Data Mining Approach for Detection of DDoS Attacks Utilizing SSL/TLS Protocol*	
4	Goel et al. (2013)	Wireless LAN (WLAN) Spoofing Detection Methods - Analysis and the victim Silent case	WLAN spoofing detection
5	Jaber et al. (2013)	Highly Effective Filtration and Prevention Framework for Secure Incoming VoIP Calls	Ourmon and Snort rules. SIP packet payload
6	Ramesh et al. (2012)	Wireless Sensor Network Security: Real-Time Detection and Prevention of Attacks	DDoS in WSN
7	○ Chen et al. (2012)	Detecting SIP flooding attacks on IP Multimedia Subsystem (IMS)	SIP packet payload (p. 157)
8	Hantehzadeh et al. (2012)	Statistical analysis of self-similar Session Initiation Protocol (SIP) messages for anomaly detection	SIP packet payload
9	Son et al. (2010)	Detecting Anomaly Traffic using Flow Data in the real VoIP network	SIP packet payload
10	• *Chen et al. (2007)*	*An Inline Detection and Prevention Framework for Distributed Denial of Service Attacks*	
11	Yan et al. (2004)	Defending Against Traffic Analysis Attacks with Link Padding for Bursty Traffics	Prevention system on Traffic Analysis, not DDoS

Table 18. ACM Digital Library studies after inclusion criteria

	Authors	Title	Exclusion reason
1	• *V. Eliseev and A. Gurina (2016)*	*Algorithms for network server anomaly behavior detection without traffic content inspection*	
2	J. Brynielsson and R. Sharma (2015)	Detectability of Low-Rate HTTP Server DoS Attacks using Spectral Analysis	HTTP Apache log analysis, not encrypted traffic
3	• *Hoeve, M. (2013)*	*Detecting Intrusions in Encrypted Control Traffic*	
4	Bansal et al. (2012)	Detection of NDP Based Attacks using MLD	Not encrypted IPv6. DoS detection based on a threshold
5	K. Sourav and D.P. Mishra (2012)	DDoS Detection and Defense: Client Termination Approach	HTTP packet payload detection
6	Benton et al. (2011)	SignatureCheck A Protocol to Detect Man-In-The-Middle Attack in SSL	Context correct, but no DDoS
7	W. Wang and X. Zhang (2011)	High-speed Web Attack Detection through Extracting Exemplars from HTTP Traffic	Packet payload analysis
8	• *Das et al. (2011)*	*Detection of HTTP Flooding Attacks in Multiple Scenarios*	
9	Y. Lee and Y. Lee (2011)	Detecting DDoS Attacks with Hadoop	Requires payload inspection to filter non-HTTP GET packets out
10	○ Jamdagni et al. (2010)	Intrusion Detection Using GSAD Model for HTTP Traffic on Web Services	HTTP payload features (p. 1194)
11	Wang et al. (2009)	A General Framework for Adaptive and Online Detection of Web attacks	Does not specify the nature of the traffic nor the features extracted for the HTTP traffic
12	○ Kloft et al. (2008)	Automatic Feature Selection for Anomaly Detection	Payload features (p. 74)
13	K. L. Ingham and A. Somayaji (2007)	A Methodology for Designing Accurate Anomaly Detection Systems	Packet payload analysis
14	• *Caulkins et al. (2005)*	*A Dynamic Data Mining Technique for Intrusion Detection Systems*	
15	M. V. Mahoney (2003)	Network traffic anomaly detection based on packet bytes	Payload analysis

Table 19. IEEE included studies after inclusion criteria

	Authors	Title	Exclusion reason
1	• *Zolotukhin et al. (2016)*	*Increasing Web Service Availability by Detecting Application-Layer DDoS Attacks in Encrypted Traffic*	
2	Nagpal et al. (2016)	DDoS Tools: Classification, Analysis and Comparison	DDoS tools, not detection
3	Agarwal et al. (2015)	Detection of De-authentication DoS attacks in Wi-Fi Networks: A Machine Learning Approach	Needs to analyze payload for Deauth-packets
4	H. Sedjelmaci and S. M. Senouci (2014)	A Lightweight Hybrid Security Framework for Wireless Sensor Networks	WSN
5	Soryal et al. (2014)	Combating Insider Attacks in IEEE 802.11 Wireless Networks with Broadcast Encryption	MANET, RTS/CTS handshake exploit DoS
6	• *Petiz et al. (2014)*	*Detecting DDoS Attacks at the Source Using Multiscaling Analysis*	
7	• *P. V. Amoli and T. Hämäläinen (2013)*	*A Real Time Unsupervised NIDS for Detecting Unknown and Encrypted Network Attacks in High Speed Network*	
8	Liying et al. (2013)	A SECURE MECHANISM FOR NETWORKED CONTROL SYSTEMS BASED ON TRUETIME	Only models DDoS attacks in a mathematical model
9	Agarwal et al. (2013)	Detection of De-authentication Denial of Service attack in 802.11 networks	Needs to analyze payload for Deauth-packets
10	○ Doh et al. (2013)	Secure Aggregation and Attack Detection for Smart Grid System	Not encr. DDoS (p. 274)
11	Taylor and Fokum (2013)	Securing Wireless Sensor Networks from Denial-of-Service Attacks using Artificial Intelligence and the CLIPS Expert System Tool	DDoS in WSN
12	D. Dudek (2012)	Collaborative Detection of Traffic Anomalies using First Order Markov Chains	DDoS in WSN
13	Chen et al. (2012)	Detecting SIP flooding attacks on IP Multimedia Subsystem (IMS)	SIP packet payload
14	Hantehzadeh et al. (2012)	Statistical analysis of self-similar Session Initiation Protocol (SIP) messages for anomaly detection	SIP packet payload
15	○ Ramesh et al. (2011)	Wireless Sensor Network Security: Real-Time Detection and Prevention of Attacks	DDoS in WSN (p. 783)
16	H. Son, Y. Lee (2010)	Detecting Anomaly Traffic using Flow Data in the real VoIP network	SIP packet payload
17	J. H. Sarker and H. T. Mouftah (2010)	Throughput and Stability Improvements of Slotted ALOHA Tasks289 Wireless Networks under the Random Packet Destruction DoS Attack	ALOHA ad-hoc network DDoS
18	Nagaratna et al. (2009)	Detecting and Preventing IP-spoofed DDoS Attacks by Encrypted Marking based Detection And Filtering (EMDAF)	No proof that the method works
19	Zhao et al (2009)	Exception Triggered DoS Attacks on Wireless Networks	Signature Snort based
20	S. Madhavi (2008)	AN INTRUSION DETECTION SYSTEM IN MOBILE ADHOC NETWORKS	DDoS in MANET
21	El-Moussa et al (2007)	Active router approach to defeating denial-of-service attacks in networks	Detection method: signature
22	Padmanabhuni et al. (2006)	Preventing Service Oriented Denial of Service (PreSODoS): A Proposed Approach	Payload analysis of the XML body
23	D. S. Phatak (2005)	Spread–Identity mechanisms for DOS resilience and Security	Resilience architecture for WLAN, not DDoS detection
24	• *Abimbola et al. (2003)*	*NetHost-Sensor: A Novel Concept in Intrusion Detection Systems*	

Table 20. ScienceDirect studies after inclusion criteria

	Authors	Title	Exclusion reason
1	M. Tarao, T. Okamoto (2016)	Toward an Artificial Immune Server against Cyber Attacks: Enhancement of Protection against DoS attacks	Packet payload
2	S. Laskar, D. Mishra (2016)	Qualified Vector Match and Merge Algorithm (QVMMA) for DDoS Prevention and Mitigation	Not encrypted, features good
3	Prakash et al. (2016)	Detection and Mitigation of Denial of Service Attacks Using Stratified Architecture	No proof provided
4	Tas et al. (2015)	Novel session initiation protocol-based distributed denial-of-service attacks and effective defense strategies	Efficiency in CPU usage percentage
5	Carlin et al. (2015)	Defense for Distributed Denial of Service Attacks in Cloud Computing	Literature review
6	Patil et al. (2015)	Comparative analysis of the Prevention Techniques of Denial of Service Attacks in Wireless Sensor Network	DDoS attacks in WSN
7	○ Darwish et al. (2015)	A cloud-based secure authentication (CSA) protocol suite for defense against Denial of Service (DoS) attacks	CSA is a DDoS prevention system by design (p. 97)
8	Fichera et al. (2014)	OPERETTA: An OPEnflow-based REmedy to mitigate TCP SYNFLOOD Attacks against web servers	SYN flood, not encrypted
9	Fachkha et al. (2014)	Inferring distributed reflection denial of service attacks from darknet	DNS amp., not encrypted DoS
10	Rastegari et al. (2013)	Evolving statistical rulesets for network intrusion detection	Not about encrypted DDoS
11	Saied et al. (2014)	Detection of known and unknown DDoS attacks using Artificial Neural Networks	They: not for encrypted
12	• *Wang et al. (2014)*	*DDoS attack protection in the era of cloud computing and Software-Defined Networking*	
13	Alajeely et al. (2014)	Catabolism attack and Anabolism defense: A novel attack and traceback mechanism in Opportunistic Networks	OPNET
14	Tsiatsikas et al. (2014)	An efficient and easily deployable method for dealing with DoS in SIP services	SIP packet payload
15	Vissers et al. (2013)	DDoS defense system for web services in a cloud environment	XML payload
16	○ Shamshirband et al. (2013)	Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks	DDoS in WSN (p. 228)
17	• *Shiaeles et al. (2011)*	*Real time DDoS detection using fuzzy estimators*	
18	Tariqa et al. (2011)	Collaborative Peer to Peer Defense Mechanism for DDoS Attacks	Collaboration architecture, not encrypted DoS
19	Roh et al. (2011)	A whitelist-based countermeasure scheme using a Bloom filter against SIP flooding attacks	SIP payload analysis
20	Ehlert et al. (2009)	Survey of network security systems to counter SIP-based denial-of-service attacks	Literature review
21	B. Li, L. Batten (2008)	Using mobile agents to recover from node and database compromise in path-based DoS attacks in wireless sensor networks	WSN
22	• *Lee et al. (2007)*	*DDoS attack detection method using cluster analysis*	
23	Agah et al. (2006)	Security enforcement in wireless sensor networks: A framework based on non-cooperative games	WSN
24	C. Douligeris, A. Mitrokotsa (2003)	DDoS attacks and defense mechanisms: classification and state-of-the-art	Taxonomy and review of methods