



Norwegian University of  
Science and Technology

# Automatic Analysis of Scam Emails

**Vegard Fagerland**

Master of Science in Telematics - Communication Networks and Networked

Submission date: July 2017

Supervisor: Maria Bartnes, IIK

Co-supervisor: Erlend Andreas Gjære, SINTEF

Norwegian University of Science and Technology

Department of Information Security and Communication Technology





**NTNU – Trondheim**  
Norwegian University of  
Science and Technology

# Automatic Analysis of Scam Emails

**Vegard Fagerland**

Submission date: July 2017  
Responsible professor: Maria Bartnes, NTNU, SINTEF  
Supervisor: Erlend Andreas Gjære, SINTEF

Norwegian University of Science and Technology  
Department of Telematics





**NTNU – Trondheim**  
Norwegian University of  
Science and Technology

**Title:** Automatic Analysis of Scam Emails  
**Student:** Vegard Fagerland

**Problem description:**

Electronic mail, which is abbreviated email, is a major security concern. Even though numerous security features can be implemented, malicious emails will probably get through to end-users' inboxes. Email is commonly used to distribute malware, phishing and, scams – tricking the user into clicking on malicious links, opening infected attachments, and submitting personal information among others. Spam filters and scanning of email attachments continue to evolve. However, so do the malicious actors and their methods and it becomes harder and harder to identify malicious emails and act accordingly. As of today, it depends on the security awareness and scepticism of the end-users. There is potential for handling such emails in a smarter way after they have reached the users' inbox. This could reduce the success rate of attacks and protect users from unfortunate consequences.

The purpose of this project is to identify characteristics and search for multiple denominators by the means of analysis on a large collection of possibly malicious emails. These emails have all passed standard security mechanisms, and could be viewed with less scepticism and as secure to open. This will be the basis for developing an analysis engine for scam emails which can assess each email and further decide on proper actions to be taken.

**Responsible professor:** Maria Bartnes, NTNU, SINTEF  
**Supervisor:** Erlend Andreas Gjære, SINTEF



## Abstract

Email and email security have been the main topics of this master thesis. The thesis considers how an organization works with email security and security culture, the email specifications, threat agents, vulnerabilities and attacks distributed via email. Several technological security features are standard in email systems nowadays. Technological evolution and development give better solutions for filtering and rejecting malicious email. However, new vulnerabilities are exploited and new attacks take place. Some email containing malware, phishing, or scam will probably get through to end-users' inboxes. The only truly effective protection found is to promote email security and make email users aware of the potential threats.

Today there are no good solutions for dealing with email that have passed these technical security measures. As a part of the organizations work to improve security culture, a functionality for users to report suspicious emails has been developed. This enables users to directly report suspicious emails to IT security personnel by a simple click of a button. However, as of now, it is up to the IT security personnel to manually perform analysis on the reported emails. This takes time, and the amount of reported emails increases every week.

To improve email security and reduce time spent on manual analysis there is potential for handling such email in a smarter way. One solution is to automate the process of analysing the suspicious emails reported. This automation tool could help IT security personnel reduce risks and provide information to other users so that measures could be taken to stop malicious email.

Results from data analyses and hypotheses testing show that it would be beneficial with better information to the users, and to implement some added functionality to the reporting of suspicious emails. This would better achieve the intention of having users report suspicious email as a part of the email security work. An automated system for extracting and parsing reported emails can be used for alerting users and informing system administrators. Further, with few modifications, this system could be used with data from the reported emails to proactively block or filter future emails before they reach end-users' inboxes.





## Sammendrag

Hovedtemaene i denne master oppgaven har vært epost og epostsikkerhet. Oppgaven ser på hvordan en organisasjon arbeider med epostsikkerhet og sikkerhetskultur, de ulike epostspesifikasjonene, trusselaktører, sårbarheter og angrep via epost. Tekniske sikkerhetsløsninger er standard i dagens epostsystemer. Likevel oppstår nye sårbarheter, disse utnyttes og nye angrep forekommer. Epost som inneholder skadelig programvare, forsøk på phishing eller svindel vil trolig passere de tekniske løsningene og ende opp i sluttbrukernes innbokser. Den eneste virkelige effektive beskyttelsen mot trusler gjennom epost er å fremme epostsikkerhet og gjøre epostbrukerne oppmerksomme på de potensielle truslene.

I dag finnes det ingen gode metoder for å håndtere epost som har passert de tekniske sikkerhetsmekanismene. Som en del av organisasjonens arbeid med sikkerhetskultur er det utviklet en funksjonalitet for å rapportere mistenkelige eposter. Dette gjør det mulig for epostbrukerne å rapportere mistenkelig eposter direkte til IT sikkerhetspersonell ved hjelp av et enkelt museklikk. Per i dag er det opp til dette personellet å manuelt håndtere og analysere de rapporterte epostene. Dette er tidkrevende og mengden rapporterte eposter øker hver uke.

Det ligger et potensiale i å håndtere slik epost på en smartere måte. Dette vil kunne forbedre epostsikkerhet og redusere tiden brukt på manuell analyse. En mulig løsning er å automatisere behandlingen av de mistenkelige epostene som er blitt rapportert. Dette kan bidra i arbeidet med epostsikkerhet, hjelpe IT sikkerhetspersonell og muligens stoppe skadelige epost.

Resultatene fra dataanalyser og hypotesetesting viser at både informasjon til brukere og en tilleggsfunksjonalitet i rapporteringsverktøyet vil kunne gi mulige forbedringer i sikkerhetsarbeidet. Dette vil styrke hensikten ved å få epostbrukere til å rapportere mistenkelig epost som en del av sikkerhetsarbeidet. Et automatisert system for å hente ut data og gjøre analyse av rapporterte epostmeldinger kan brukes til å varsle epostbrukere og informere systemadministratorer. Ved hjelp av få modifikasjoner kan data fra de rapporterte epostene brukes proaktivt for å blokkere eller filtrere fremtidige epost før de når frem til sluttbrukernes innboks.



## Preface and Acknowledgements

This master thesis is a completion of the authors specialization in Information Security and Master of Science degree in Telematics at the Norwegian University of Science and Technology (NTNU).

Honestly, when I started, I did not fully understand how vast of a topic email and email security actually is. Still, the work with this master thesis has been very interesting and challenging. One of the biggest motivational reasons during this work has been the general attention email security and attacks via email have had in the media and day-to-day life the past year.

I would like to thank my professor Maria Bartnes and my supervisor research scientist Erlend Andreas Gjære at The Foundation for Scientific and Industrial Research (SINTEF). I have learned a lot through meetings and discussions on how an organization face email threats and how they view email security work and security culture.

Finally, I want to thank my fiancé and family for their support and understanding throughout this process.

Vegard Fagerland  
Trondheim, Norway  
July 2017



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	3
1.2 Scope . . . . .	4
1.3 Research questions . . . . .	4
1.4 Outline . . . . .	5
<b>2 Background</b>	<b>7</b>
2.1 Email . . . . .	7
2.1.1 Email message format . . . . .	9
2.2 Technical security measures . . . . .	11
2.2.1 Sender Policy Framework . . . . .	12
2.2.2 DomainKeys Identified Mail . . . . .	13
2.2.3 Domain Message Authentication Reporting & Conformance . . . . .	15
2.2.4 Spam filters . . . . .	16
2.2.5 Anti-malware protection, malicious attachments and URLs . . . . .	17
2.3 Human factors . . . . .	18
<b>3 Method</b>	<b>19</b>
3.1 Qualitative method, literature and case study . . . . .	19
3.2 Quantitative method, data collection and testing . . . . .	20
<b>4 Design and implementation</b>	<b>23</b>
4.1 Retrieving and parsing emails . . . . .	23
4.2 Environment for visualizing data . . . . .	28
4.3 Obtaining measurable data from the dataset . . . . .	30
<b>5 Questionnaire on suspicious emails from the dataset</b>	<b>33</b>
5.1 Overall results . . . . .	33
5.2 Individual questionnaire responses and results . . . . .	35

<b>6</b>	<b>Data analyses and hypotheses testing</b>	<b>47</b>
6.1	General data and statistics from dataset . . . . .	47
6.1.1	Outer-email, which is generated when users report suspicious emails . . . . .	47
6.1.2	Inner-email, the suspicious emails being reported by users . .	49
6.2	Comparison of different message header fields . . . . .	53
6.3	Hypotheses testing . . . . .	57
6.3.1	Hypothesis 1, Strict enforcement of SCL . . . . .	57
6.3.2	Hypothesis 2, Strict enforcement of email authentication . . .	57
6.3.3	Hypothesis 3, Notify IT security personnel based on reported emails . . . . .	58
6.3.4	Hypothesis 4, Block or deliver emails to spam folder based on reported emails . . . . .	60
<b>7</b>	<b>Discussion and suggested solutions</b>	<b>65</b>
7.1	How is the functionality for reporting email used . . . . .	67
7.1.1	Information about the functionality, and expanding the functionality for reporting emails . . . . .	68
7.2	How could the reported emails be used more efficiently . . . . .	70
7.2.1	Temporarily block or filter emails for some time . . . . .	71
7.2.2	Functionality to alert users when receiving emails . . . . .	71
<b>8</b>	<b>Conclusion and further work</b>	<b>73</b>
8.1	Conclusion . . . . .	73
8.2	Further work . . . . .	74
8.2.1	Virtual sandbox environment . . . . .	74
	<b>References</b>	<b>77</b>
	<b>Appendices</b>	
<b>A</b>	<b>X-MS-Exchange-Organization-SCL</b>	<b>85</b>
<b>B</b>	<b>Received-SPF</b>	<b>87</b>
<b>C</b>	<b>DKIM-signature</b>	<b>89</b>
<b>D</b>	<b>Authentication-Results</b>	<b>91</b>
D.1	Authentication-Results SPF . . . . .	92
D.2	Authentication-Results DKIM . . . . .	93
D.3	Authentication-Results DMARC . . . . .	94
<b>E</b>	<b>X-Forefront-Antispam-Report</b>	<b>95</b>
E.1	X-Forefront-Antispam SCL . . . . .	95

E.2	X-Forefront-Antispam SPF . . . . .	96
<b>F</b>	<b>Comparison of results from email headers</b>	<b>99</b>
F.1	X-MS-Exchange-Organization-SCL and Received-SPF . . . . .	100
F.2	X-Forefront-Antispam-Report SCL and X-MS-Exchange-Organization-SCL . . . . .	101
F.3	X-Forefront-Antispam-Report SPF and Received-SPF . . . . .	102
F.4	X-Forefront-Antispam-Report SPF and Authentication-results SPF	103
F.5	Authentication-Results SPF and Received-SPF . . . . .	104





# List of Figures

2.1	This figure shows an example of the email message format with its three parts. The first part is an example of the email header. The second part shows some of the message header fields. While the third part, the message body, can be seen after the empty line of the message header fields. . . . .	11
2.2	Flow of Sender Policy Framework. . . . .	13
2.3	Flow of DKIM framework. . . . .	15
2.4	Email statistics 2014-2016 [1]. . . . .	16
4.1	Example of the email structure for the emails that are parsed. . . . .	24
4.2	Example of email structure where the message/rfc822 only have text, no attachments. . . . .	25
4.3	Shows the targeted domain where users report suspicious email, these emails are so sent to the email server of the test domain. . . . .	25
4.4	Shows the basic design of the environment for retrieving, parsing email headers and storing them to the database. . . . .	26
4.5	Flowchart parsing emails and saving parsed data to the database. . . . .	28
4.6	Pagemap, showing different trust zones and user inputs. . . . .	29
4.7	Complete test environment. Reporting domain on the left side, test domain and email server in the middle and test environment on the right side. . . . .	30
4.8	Basic setup of the testing on the dataset. . . . .	31
5.1	Total answers from the questionnaire. . . . .	34
5.2	Questionnaire, email number 1. . . . .	35
5.3	Questionnaire, email number 2. . . . .	36
5.4	Questionnaire, email number 3. . . . .	37
5.5	Questionnaire, email number 4. . . . .	38
5.6	Questionnaire, email number 5. . . . .	39
5.7	Questionnaire, email number 6. . . . .	40
5.8	Questionnaire, email number 7. . . . .	41
5.9	Questionnaire, email number 8. . . . .	42

5.10	Questionnaire, email number 9. . . . .	43
5.11	Questionnaire, email number 10. . . . .	44
5.12	Questionnaire, email number 11. . . . .	45
5.13	Questionnaire, email number 12. . . . .	46
6.1	Weekday and time when the emails were reported. . . . .	48
6.2	Weekday and time when the emails were received. . . . .	50
6.3	X-Forefront-Antispam-Report SCL and X-MS-Exchange-Organization-SCL. . . . .	55
6.4	Authentication-Results SPF and Received-SPF. . . . .	56
7.1	Adding functionality to email reporting, example 1. . . . .	69
7.2	Adding functionality to email reporting, example 2. . . . .	70
7.3	Example of alerting users about a reported email address with a pop-up window. . . . .	72
7.4	Example of alerting users in their email client. . . . .	72
8.1	Flowchart of the environment with sandbox. . . . .	75
A.1	X-MS-Exchange-Organization-SCL . . . . .	86
B.1	Received-SPF status . . . . .	88
D.1	Authentication-Results SPF status . . . . .	92
D.2	Authentication-Results DKIM status . . . . .	93
D.3	Authentication-Results DMARC status . . . . .	94
E.1	X-Forefront-Antispam-Report SCL status . . . . .	96
E.2	X-Forefront-Antispam SPF status . . . . .	97

# List of Tables

2.1	Email message header fields as specified in RFC 5322 [2]. . . . .	10
6.1	Complementary data to figure 6.1. . . . .	48
6.2	Complementary data to figure 6.2. . . . .	50
6.3	Top reported sending domains where SCL is -1,0,1,2,3,4 or No Value. . .	52
6.4	Top reported sending domains where SCL value is 5, 6, 7, 8 or 9. . . . .	53
6.5	Complementary table to figure 6.3. X-Forefront-Antispam-Report SCL is shown horizontally, X-MS-Exchange-Organization-SCL is shown vertically in the table. . . . .	54
6.6	Complementary table to figure 6.4. . . . .	56
6.7	Top reported <i>bulk-emails</i> , more than 10 reported. . . . .	59
6.8	Showing domain names, from emails reported with SCL value -1, as white-listed. . . . .	59
6.9	Showing domain names, from emails reported with no SCL or Recieved-SPF message header. . . . .	60
6.10	Complementary results to test 4.1, only showing a selection of senders with more than 25 emails. Usernames, part of the email address, identifying people by name are anonymized. . . . .	61
6.11	Complementary results to test 4.2, only showing a selection of senders with more than 15 emails. Usernames, part of the email address, identifying people by name are anonymized. . . . .	63
A.1	Complementary data to graph in figure A.1 . . . . .	85
A.2	Complementary data to graph in figure A.1 . . . . .	86
B.1	Complementary data to graph in figure B.1 . . . . .	87
B.2	Complementary data to graph in figure B.1 . . . . .	88
C.1	DKIM-signature header data . . . . .	89
C.2	Top domains in DKIM-signature header field, more than 18 received . .	89
D.1	Complementary data to figure D.1, D.2 and D.3 . . . . .	91

D.2	Complementary data to figure D.1 . . . . .	91
D.3	Complementary data to figure D.2 . . . . .	91
D.4	Complementary data to figure D.3 . . . . .	91
D.5	Complementary data to figure D.1 . . . . .	92
D.6	Complementary data to figure D.2 . . . . .	93
D.7	Complementary data to figure D.3 . . . . .	94
E.1	Complementary data to figures E.1 and E.2. . . . .	95
E.2	Complementary data to figure E.1 . . . . .	95
E.3	Complementary data to figure E.2 . . . . .	96
F.1	X-MS-Exchange-Organization-SCL and Received-SPF . . . . .	100
F.2	X-Forefront-Antispam-Report SCL and X-MS-Exchange-Organization-SCL	101
F.3	X-Forefront-Antispam-Report SPF and Received-SPF . . . . .	102
F.4	X-Forefront-Antispam-Report SPF and Authentication-results SPF . . .	103
F.5	Authentication-Results SPF and Received-SPF . . . . .	104

# Chapter 1

## Introduction

The topic of this report is how an organization's employees could be used to report suspicious emails in order to strengthen email security against the continuous flood of potential malicious email. They can do so by reporting suspicious emails, so that IT security personnel can perform analyses, discover email trends and patterns. Peoples risk perception and security awareness could help detect suspicious email and alert the IT security personnel. One way to increase this is through better training and attention to security. In many cases there are certain observations that only humans can make where technical solutions fail for detecting some type of malware, scams, phishing, and social engineering attempts [3]. When suspicious emails have been detected and reported to IT security personnel the emails are manually analysed. This is time consuming and the amount of reported email increases every day. Consequently, there is potential for handling such email in a more automated manner. This could give the IT security personnel another tool in handling malicious emails and a potential for earlier warning. The latest attacks, phishing emails with some types of ransomware [4], should be a serious reminder about the potential damage that can be caused when both technical solutions and human awareness fail.

Today, twenty-six years after the Internet<sup>1</sup> took its first baby-steps, digitalization is probably the biggest technological development since the nineteenth century and the industrial evolution. The digital revolution has brought massive social, economic and technological changes at an enormous pace. In 2016, Norway is in the top five on the list of the worlds most digitized countries [5]. Over ninety-six percent of the Norwegian population are online [6]. The number is even higher for Norwegian enterprises with at least hundred employees or more. As much as ninety-eight percent of these enterprises have some sort of Internet connection [7]. These numbers reflect on how important and how extensive the use of ICT<sup>2</sup> has become.

---

<sup>1</sup>from the predecessor of ARPANET in the 80s, to the start of World Wide Web in 1991 and the increase of use through the mid-90s and today

<sup>2</sup>Information and communication technology

One of the advantages of digitalization is that it has increased the efficiency of communication and given numerous options to how we communicate. Instant messaging, video chat, social media, and email are just some of the tools that can be used as opposed to the regular mail and the old telephone system. Email, in contrast to the other means of communication, has existed since long before the use of the Internet started picking up speed. But the growth in the number of email users and the volume of email sent can be credited to the success and widespread use of the Internet. Almost half of the world's population today have one or multiple email accounts [8]. When looking at both business and consumer email users the use is expected to increase even further the coming years [9]. Even though instant messaging has become popular, there are so many different platforms that are not inter-operable, e.g. Facebook Messenger does not allow you to send instant messages to Skype and the other way around. Email is available on several different platforms, e.g. email clients from popular email providers as Gmail and Outlook. By using email, users can send and receive email to one person or a group of people without any difficulties. Adding to this, email is used heavily by businesses in marketing, and advertisement as well as an internal and external means of communication. These are all some of the reasons to why this old-age technology will continue to be relevant in the future.

Due to the extensive use of ICT and the continuously technological development, Norwegian society is becoming increasingly vulnerable to attacks from different threat agents with varying motives [10]. Some of these agents are foreign states trying to access digital infrastructure to retrieve information about advanced technology and research. Lately, foreign states have also allegedly manipulated elections and tried to alter public opinion. Other actors have different motives and look for financial opportunities to enrich themselves. Information security, and more specifically email security, has been brought to the public's attention through publications and media attention. Despite this, it is estimated that over ten percent [11] of Norwegian computers are infected with some sort of malicious software, also referred to as malware. The most common way these attacks are conducted is through digital attacks, at times targeted attacks, using email with some type of malicious attachment or URL<sup>3</sup> links [13]. This is done to trick the recipient to open attachments or falsified URL links which could give the attacker illegitimate access to parts of the ICT system. Today this is the common choice of method and it has shown to be highly effective. The survey conducted in 2016 about Norwegian Computer Crime and Data Breach states that some of the major contributing factors to information security incidents have been human error and lack of security awareness [14]. This can be seen in relation to another survey from 2016 on Norwegian Cyber Security Culture

---

<sup>3</sup>A URL (Uniform Resource Locator) provides a way to locate a resource on the web, the hypertext system that operates over the internet. The URL contains the name of the protocol to be used to access the resource and a resource name. The first part of a URL identifies what protocol to use. The second part identifies the IP address or domain name where the resource is located [12].

where participants answered on how much risk they associate with using email. Over fifty percent answered that they were not worried, eighteen percent answered that they were worried [15].

## 1.1 Motivation

The goal with email security is to not set limitations on how, when, from where or to whom people communicate within an organization or with externals. Email security and the security measures are about facilitating and supporting use of email, so that it can be used in a secure way without unfortunate consequences. The motivation behind considering this topic area comes from the fact that the use of email in and outside business communication is still the most common way of communicating [16]. In addition, The Norwegian National Security Authority (NSM) reports that, the use of email is the most common method of attack registered in the successful targeted attacks where businesses have been affected. Technical solutions continue to evolve, and consumers buy and use equipment with better security features, updated software and operating systems [13]. Nevertheless, new vulnerabilities will most likely be discovered and can be exploited. At the same time threat agents and their modus operandi<sup>4</sup> will change and adapt. Addressing and educating users on email threats is motivating. Particularly, when this is done to make users feel some sort of ownership and a presence of accountability towards secure use of email.

A malicious attacker, for whatever reason, trying to gain access to information systems, will only need to be successful once to cause potential damage. Hence, working with email security is a formidable task. Author Salman Rushdie is stated to have said [17] – *Working with security is an ungrateful job. Security is after all the art to make sure certain things do not happen. For when such things do not happen, there will always be some who argue that the security measures were excessive and unnecessary.* This statement, whether true or not, has become somewhat famous and popular when it comes to working with security and security culture. For sure, it also applies to email security. Email security, and more generally information security, is about protecting networks, computers, programs, data, and people from threats of attacks, damage, or financial losses. It is not all technology, which could be hard to grasp and in some ways contradictory. When thinking of email security, most people think about the technical side of it. Meaning that it is important to have technical solutions and security measures implemented. This is true, it is important, but it is also important to look at the end-users, on how they interact and use the systems, their security awareness and risk perception. Email security should in most scenarios not set restrictions and limitations. The aim of email security is to keep functionality and at the same time to be a secure means of communication.

---

<sup>4</sup>method or mode of operation

## 1.2 Scope

The scope of this report is how an organization has set email security on the agenda through a security culture initiative. They have implemented functionality so that users can report suspicious and potential malicious emails. In more details, the report looks at what type of emails these users report, and how these reported emails could be used in an automated manner to better email security and alert IT security personnel. The scope is not on technical security measures like encryption, firewalls, intrusion detection system, spam-filters and malware protection. However, some will be addressed in short along with some of their vulnerabilities. There should be no doubt about the importance of these and that they are implemented correctly and working. Unfortunately they are not sufficient alone. The Norwegian National Security Authority (NSM) have published several guides [18],[19] explaining what measures should be taken and implemented to avoid information security attacks. NSM have also published a guide [19] on basic measures for secure transfer of email between email clients. These guidelines are primarily addressing system and administrator level and what technical security measures should be in place. One guideline which could be more suited for the basic email user to reduce the security risk is Recognizing and Avoiding Email Scams by United States Computer Emergency Readiness Team<sup>5</sup> [15]. It highlights specific threats and how users can recognize and avoid these.

## 1.3 Research questions

Based on the problem description earlier presented in this report, the main research questions are as stated:

1. How is email used and what are the threats? How can these threats be mitigated within the organization?
2. How is the functionality for reporting suspicious emails used in the organization today?
3. How could the reported emails be used more efficiently in the email security work?

---

<sup>5</sup>US-CERT address security breach and denial-of-service incidents, providing alerts and incident-handling and avoidance guidelines. CERT also conducts an ongoing public awareness campaign and engages in research aimed at improving security systems [20].



## 1.4 Outline

The master thesis is structured as follows:

- **Chapter 1** is an introduction to, motivation for, and scope of the thesis work.
- **Chapter 2** introduce relevant background information on topic of the thesis.
- **Chapter 3** is a presentation on the choice of methods.
- **Chapter 4** shows how email was collected and parsed, documentation on design, implementation and decisions for the environment and for the data analyses in this thesis.
- **Chapter 5** presents the questionnaire conducted and its results.
- **Chapter 6** presents the data analyses, the hypotheses testing and results.
- **Chapter 7** discusses results from chapter 5 and 6, and presents suggested solutions based on the gathered data from the questionnaire and the hypotheses testing.
- **Chapter 8** gives a conclusion on the thesis work, and suggestions for future work with this topic.



# Chapter 2

## Background

The following chapter presents an introduction to email, the email message format and the major security concern it represents. It is followed by examples of technical solutions which all try to overcome some of the challenges concerning email security. These examples raise questions related to other methods for how one can reduce the risks related to email. Because, as proven time after time, technical solutions are not sufficient by themselves. Next, the chapter brings attention to how human factors can contribute in preventing security breaches through emails. This creates the foundation for the remainder of the report.

### 2.1 Email

Email is a method of exchanging digital messages. Originally, email was transmitted directly from one user's device to another's just like instant messaging. Email systems are now based on a store-and-forward model in which email server accept, forward, deliver and store messages on behalf of users [21]. The key parts of an email system are the email client, the email server and the protocols that makes sending and receiving of email possible. The email message was initially standardized as early as in 1982 with RFC<sup>1</sup> 822 [23] by the Internet Engineering Task Force<sup>2</sup> (IETF). It was later superseded by RFC 2822 [25] in 2001, and the latest RFC 5322 [26] from 2008. These three are the main RFCs, but minor updates have been published in between and the latest with RFC 6854 [27] in 2013. However, they only specify the syntax for email messages. RFC 821 [28], 2821 [29] and the latest 5321 [2] for the Simple Mail Transfer Protocol (SMTP) specifies the sending of email to email servers and the forwarding of email messages between email servers. RFC 3501 [30]

---

<sup>1</sup>A Request for Comments (RFC) is a formal document from the Internet Engineering Task Force (IETF) that is the result of committee drafting and subsequent review by interested parties [22].

<sup>2</sup>Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet [24].

is another RFC which allows a client to access and handle email messages on a server. There are also several extensions published to ensure that email has met the requirements and functionality of modern use such as HTML and Multi-Purpose Internet Mail Extensions (MIME). MIME is specified in six different RFCs, among RFC2045 [31] and RFC2046 [32], which describe mechanisms for the transmission of data attachments by using email.

Email is based on old technology and there are few security mechanisms implemented as found in the RFCs. They mainly address availability, and reliability of the email system. This makes it vulnerable to several possible attacks. One example of the lack of integrity and email authentication can be found in the address field *from* and *to* in the email format and the SMTP protocol which in the specifications are not the same. This vulnerability makes it possible for an attacker to make the email appear to be from a legitimate address. A strength, but at the same time a weakness, with email is that you can send almost any type of data as an attachment. The convenience and anonymity of email, along with the capability it provides for easily contacting thousands adds to the vulnerabilities.

Email can be used in attacks to steal information or to plant software that can later be used in exploitation. Email can also be more directly used in attacks, like phishing or spear-phishing, to trick a recipient to disclose information. There are fail-safe solutions on how to secure email and avoid being compromised. No matter how many technical security features implemented, it will never give hundred percent security. The common threats when using email are the described in National Institute of Standards and Technology's (NIST) Guidelines on Electronic Mail Security [33].

**Malware.** *Increasingly, attackers are taking advantage of email to deliver a variety of attacks against organizations from malware, or malicious software, that include viruses, worms, Trojan horses, and spy-ware. These attacks, if successful, may give the malicious entity control over workstations and servers, which can then be exploited to change privileges, gain access to sensitive information, monitor users' activities, and perform other malicious actions [33].*

**Spam, and phishing.** *Unsolicited commercial email, commonly referred to as spam, is the sending of unwanted bulk commercial email messages. Such messages can disrupt user productivity, utilize IT resources excessively, and be used as a distribution mechanism for malware. Related to spam is phishing, which refers to the use of deceptive computer-based means to trick individuals into responding to the email and disclosing sensitive information. Compromised email systems are often used to deliver spam messages and conduct phishing attacks using an otherwise trusted email address [33].*

**Social engineering.** *Rather than hack into a system, an attacker can use e-mail to gather sensitive information from an organization's users or get users to perform actions that further an attack. A common social engineering attack is email spoofing, in which one person or program successfully masquerades as another by falsifying the sender information shown in emails to hide the true origin [33].*

### 2.1.1 Email message format

The structure of an email message format has three basic parts as specified in the RFCs. The first part is the header, which is a set of lines containing information about how the message was transported. This part is explained closer in RFC 2821 [29]. These lines consist of the sender's address, the recipient's address, timestamps and information about the different *hops* showing when the message was sent by intermediary servers to the mail transfer agents (MTA)<sup>3</sup>. It begins with a *Received: from* line and is added for every time it passes through an intermediary server. From this header, one can see the exact path taken by the email. Where the email originated, the path through the Internet and the email's destination before being delivered to the end-user. It also shows how much time each server spent processing that actual email. The second part is the message. This part contains information about the message in several message header fields. It must include at least three headers. The *from* header with the sender's email address. The *to* header with the recipient's header. And the *date* header which indicates the date and time of when the email was sent. There are also several other different message header fields as described in the RFCs, which can be seen in table 2.1. The header fields *optional-field* is worth mentioning. These are non-standardized fields that starts with an X-, and are used by mail user agents<sup>4</sup>. The third and last part is the body, also referred to as the message body. It contains the actual message, separated from the message email headers by a line break. This part can consist of multiple parts depending on what type of content and formats it holds. An example can be seen in figure 2.1.

---

<sup>3</sup>A message transfer agent (MTA) is a software application used within an Internet message handling system (MHS). It is responsible for transferring and routing an electronic mail message from the sender's computer to the recipient's computer. The basic platform for an MTA is an exchange system with client/server architecture [34].

<sup>4</sup>A mail user agent (MUA) is a program that allows you to receive and send e-mail messages; it's usually just called an e-mail program [35].

**Table 2.1:** Email message header fields as specified in RFC 5322 [2].

Field	Min number	Max number	Notes
trace	0	unlimited	Block prepended
resent-date	0*	unlimited*	One per block, required if other resent fields are present
resent-from	0	unlimited*	One per block
resent-sender	0*	unlimited*	One per block, MUST occur with multi-address resent-from
resent-to	0	unlimited*	One per block
resent-cc	0	unlimited*	One per block
resent-bcc	0	unlimited*	One per block
resent-msg-id	0	unlimited*	One per block
orig-date	1	1	
from	1	1	See sender
sender	0*	1	MUST occur with multi-address from
reply-to	0	1	
to	0	1	
cc	0	1	
bcc	0	1	
message-id	0*	1	SHOULD be present
in-reply-to	0*	1	SHOULD occur in some replies
references	0*	1	SHOULD occur in some replies
subject	0	1	
comments	0	unlimited	
keywords	0	unlimited	
optional-field	0	unlimited	

```

1 Received: from VE1EUR03FT006.eop-EUR03.prod.protection.outlook.com
2 (2a01:111:f400:7e09::209) by DB5PR06CA0042.outlook.office365.com
3 (2a01:111:e400:52c2::52) with Microsoft SMTP Server (version=TLS1_2,
4 cipher=TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P256) id 15.1.1199.15 via
5 Frontend Transport; Wed, 21 Jun 2017 07:16:13 +0000
6 Received: from omr-a010e.mx.aol.com (204.29.186.54) by
7 VE1EUR03FT006.mail.protection.outlook.com (10.152.18.116) with Microsoft SMTP
8 Server (version=TLS1_0, cipher=TLS_RSA_WITH_AES_256_CBC_SHA) id 15.1.1178.14
9 via Frontend Transport; Wed, 21 Jun 2017 07:16:12 +0000
10 Received: from 206.123.152.28 by webprd-a30.mail.aol.com (10.72.52.207) with HT
11 Wed, 21 Jun 2017 03:16:10 -0400
12 From: =?utf-8?B?7GFycyBTw7hydW0=? <fnktnko@aol.com>
13 To:
14 Subject: Sats.
15 Thread-Topic: Sats.
16 Thread-Index: AQH56LST5gre5kFVnk5SR64fq+Zi5A==
17 Date: Wed, 21 Jun 2017 07:16:10 +0000
18 Message-ID: <15cc9822ca8-148-38f0@webprd-a30.mail.aol.com>
19 Content-Language: nb-NO
20 X-MS-Exchange-Organization-AuthAs: Anonymous
21 X-MS-Exchange-Organization-AuthSource: AM5EUR03FT064.eop-EUR03.prod.protection.
22 X-MS-Exchange-Organization-Network-Message-Id: 70bbfda0-f264-4428-93b1-08d4b87:
23 X-MS-Exchange-Organization-SCL: 1
24 X-MS-Exchange-Organization-OriginalServerIpAddress: 25.152.17.53
25 received-spf: Pass (protection.outlook.com: domain of aol.com designates
26 204.29.186.54 as permitted sender) receiver=protection.outlook.com;
27 client-ip=204.29.186.54; helo=omr-a010e.mx.aol.com;
28 dkim-signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=mx.aol.com;
29 s=20150623; t=1498029371; bh=vATUPPKVZxsvdHrMuZ6R6+vbzET0q7yhYI/
30 h=From:To:Subject:Message-Id:Date:MIIME-Version:Content-Type;
31 b=ag+10bV0XVyo0E441VXNDV/LQk+cX5qm/swsEfyAtx89ruwMn7mH5a0qA1MK4W
32 AQjMYG3vLNsXT5KFezm14LkgBDW/KWHMMjMlPAz3Kh2ZAhoXdDdsLbmAjcxtMs6eo
33 2CUS4Xpk8dN7QWu+v7SfmxLXSGIWCbnX0959e5r0=
34 X-maller: JAS STD
35 X-MS-PublicTrafficType: Email
36 X-Microsoft-Exchange-Diagnostics: 1;DB5PR06MB1736;27:0PyyKvI4Ls5aaUXQxHK
+bATT5pcmbi7lH4TAGKC90Jo/nRdx11vhVfLI7w3cDN0oIxAE6LXLKQSVSDbuh07x6Ecr5XeeII/
BBvrlt4nszGBNkXLA482yL/65YNo49A1vsvxCV5G15Q55Ho5CJzXYpQ==
37 Content-Type: multipart/alternative;
38 boundary="._000_15cc9822ca814838f0webprda30mailaolcom_"
39 MIME-Version: 1.0
40
41 --._000_15cc9822ca814838f0webprda30mailaolcom_
42 X-Microsoft-Exchange-Diagnostics: 1;DB5PR06MB1736;27:0PyyKvI4Ls5aaUXQxHK
+bATT5pcmbi7lH4TAGKC90Jo/nRdx11vhVfLI7w3cDN0oIxAE6LXLKQSVSDbuh07x6Ecr5XeeII/
BBvrlt4nszGBNkXLA482yL/65YNo49A1vsvxCV5G15Q55Ho5CJzXYpQ==
43 Content-Type: text/plain; charset="utf-8"
44 Content-Transfer-Encoding: base64
45
46 TFyaxQsDQnckh2Y5BlciB2YXN1dGFrdXJzZW4gZm9yIEV1cm8gaSBkYnYk/DQoNcG0KTGFycyBT
47 w7hydW0uDUQpTZw5kdCBmcEgbluIGlQaG9uZS4NCg0KDQo=
48
49 --._000_15cc9822ca814838f0webprda30mailaolcom_
50 X-Microsoft-Exchange-Diagnostics: 1;DB5PR06MB1736;27:0PyyKvI4Ls5aaUXQxHK
+bATT5pcmbi7lH4TAGKC90Jo/nRdx11vhVfLI7w3cDN0oIxAE6LXLKQSVSDbuh07x6Ecr5XeeII/
BBvrlt4nszGBNkXLA482yL/65YNo49A1vsvxCV5G15Q55Ho5CJzXYpQ==
51 Content-Type: text/html; charset="utf-8"
52 Content-ID: <1EF40B2E574E374B8ED245A78C3365F9@eurprd06.prod.outlook.com>
53 Content-Transfer-Encoding: base64

```

**Figure 2.1:** This figure shows an example of the email format with its three parts. The first part is an example of the email header. The second part shows some of the message header fields. While the third part, the message body, can be seen after the empty line of the message header fields.

## 2.2 Technical security measures

There are several technical solutions continuously evolving trying to stop attacks via email. Email systems have solutions like firewalls, spam-filters, malware scanning and detection implemented on the actual email server, client hosts or in the network. In reality some spam email or email with malicious content will pass these technical security measures. These are often based on a reactive approach, meaning that they are updated according to earlier detected and identified malicious activity, email

addresses or content, attachments, URLs and malware signatures.

### 2.2.1 Sender Policy Framework

Sender Policy Framework (SPF) is one among multiple sender authentication protocols. SPF is a technical method designed to protect against forgery of email sender identities [36], known as *email spoofing*. Even though SPF has been around since 2006, it became a proposed standard in 2014 following the publications of RFC 7208 [37] and RFC 7372. For this to work the system administrator of a domain needs to publish a SPF record. This is a list of authorized hosts allowed to use their domain name in the Domain Name System<sup>5</sup> (DNS). An example of a SPF flow can be seen in figure 2.2.

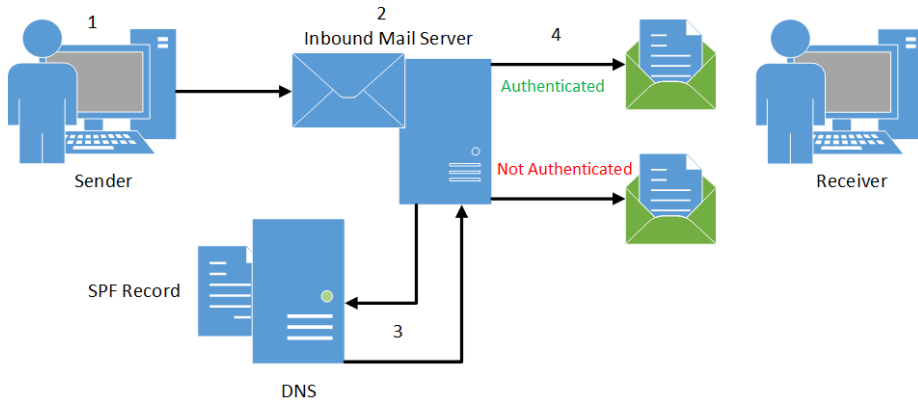
- (1), The flow starts when the sender tries to send an email to the receiver.
- (2), The inbound mail server receives the email, and obtains the name of the domain which it was sent from.
- (3), Now, the inbound email server uses this information to perform a DNS lookup to check if the SPF record for that domain. If the sending IP address in the email matches any one of the outbound addresses included in the SPF record, the email is authenticated and delivered. If no address match is possible, authentication fails.
- (4), Once the email is authenticated or the authentication has failed, the inbound email server can process the email based on the specific rules of that email system and domain [39]. A value will be added to the *Received-SPF* email message header based on the SPF check and if it passes or not. This value could be pass, fail, softfail, neutral, none, permerror or temperror which are specified and explained in RFC 7208 [37]. These are the possible results of the DNS lookup and SPF record check or SPF query. However, it is up to the inbound email server what actions are to be taken based on the evaluation results. For example, a set of rules could be to block all email that does not pass SPF, but deliver email that pass SPF to the end-users' inboxes.

The use of SPF has grown rapidly. Statistics from Google [40] in 2016 show that 9.8 % of incoming emails are authenticated by SPF, while 85.9 % are authenticated by SPF and DomainKeys Identified Email (DKIM). Nevertheless, as the technical community has provided a method for avoiding sender address forgery, threat agents have also

---

<sup>5</sup>The domain name system (DNS) is the way that internet domain names are located and translated into internet protocol (IP) addresses. The domain name system maps the name people use to locate a website to the IP address that a computer uses to locate a website [38].





**Figure 2.2:** Flow of Sender Policy Framework.

evolved. Some have started using own domains with SPF, even lookalike domains to spoof the brands, or third parties which have SPF implemented, to circumvent SPF. This will lead to SPF pass, and the email will be delivered to end-users' inbox. SPF does not solve all spoofing or phishing problems. This should be emphasized to not give a false sense of security. If strictly enforced, the cost of spamming and email address forgery would go up [41]. An issue with SPF, which applies to those who forward emails, is that it will not work as SPF is intended [42]. DNS have been mentioned as key in the SPF flow. The inbound email server checks the domain from the email against the SPF record. If DNS goes down, email would not be the biggest concern, but SPF would not work. A SPF check would not pass or worse a clear SPF fail would result in temporary or permanent error. For now, SPF is primarily used as an authentication protocol to prevent email-spoofing of own domain when sending and receiving emails. It lets organizations take responsibility for emails in transit which claim to be from their domains. Users will receive a warning in the email client if emails received are claiming to be from a user in the same domain. This warning is telling the user that *This sender failed our fraud detection checks and may not be who they appear to be.* and it provides a link with useful information on email spoofing [43]. However, to strict enforcement of SPF at the receiving side could lead to legitimate email being blocked.

### 2.2.2 DomainKeys Identified Mail

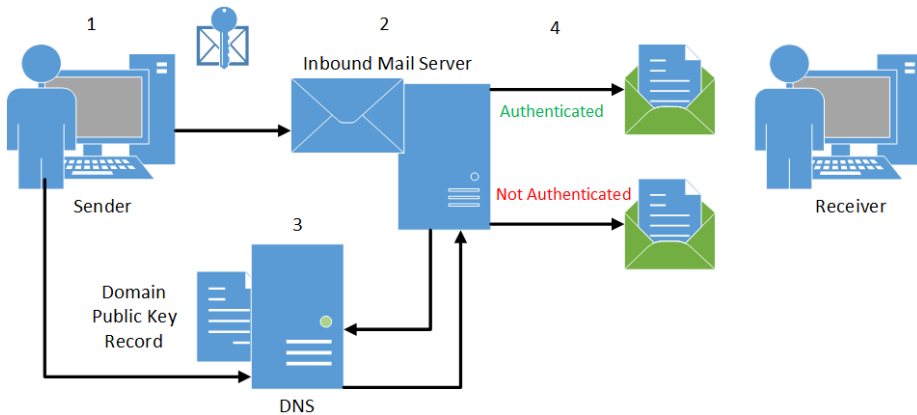
DKIM is another email authentication protocol method designed to protect against email spoofing. It does so with the use of public-key cryptography<sup>6</sup> in a more

<sup>6</sup>system that uses two keys – a public key published to DNS and known to everyone, and a private or secret key known only to the domain of the sender of the message

technical method than what SPF does with its SPF records. DKIM is a synthesized and enhanced version of Yahoo!'s DomainKeys and Cisco's Identified Internet Mail specifications. It is a result of a collaboration in the industry during 2005, to develop an open-standard e-mail authentication specification [44]. DKIM became a proposed standard in 2007 with RFC 4871 [45] and later succeeded by RFC 6376 [46] in 2011. For DKIM to work the system administrator of a domain needs to publish DNS records, a policy record and a public key record. The policy record tells the receiving email servers if the sender domain name uses DKIM. If the policy record is published, and the domain name use DKIM, a public key record will give receiving domains the public key of the sender domain in order to verify the signature of the email. An example of a DKIM flow can be seen in figure 2.3.

- (1), The flow starts when the sender tries to send an email to the receiver. The sender email platform creates a hash of the parts of the email to be signed. The hash is then encrypted by the sender domains private key [47].
- (2), The inbound mail server receives the email, and sees that it has a DKIM signature.
- (3), Now, the inbound email server uses this information to perform a DNS query to find the public key for that domain. This public key has prior to (1) been published to DNS. This public key is the only match for the private key used for signing the email, and it enables the inbound email server to decrypt the DKIM signature back to its original hash [47]. The inbound email server takes the elements of the email signed by DKIM and generates its own hash. At last the inbound email server verifies its calculated hash against the decrypted hash from the DKIM signature. If they match, the email is authenticated. If not, something has gone wrong, and the inbound email server cannot authenticate the email coming from that specific domain.
- (4), Depending on the inbound email server policies, the email will be processed and delivered to user inbox, or it will be discarded. A value will be added to the *DKIM-signature* email message header based on the DKIM test. This value could be pass, fail, none, policy, neutral, permerror or temperror [45].

The use of DKIM has also grown rapidly along with the use of SPF. Statistics from Google [40] in 2016 show that 1.7 % of incoming emails are authenticated by DKIM alone, while 85.9 % are authenticated by SPF and DKIM. DKIM is more difficult to implement than SPF, so fewer senders have adopted it. DKIM has difficulties with using mailing lists which leads to problem with authentication. If an email is modified in transit, example with mailing lists which would change one of the email message header fields, the inbound email server will calculate a different hash



**Figure 2.3:** Flow of DKIM framework.

with the sender domain public key than the hash encrypted in the original email. This lack of functionality is one of the reasons system administrators choose not to implement DKIM. As threat agents evolve, malicious emails have evolved and pass the DKIM authentication even though the cost has increased.

### 2.2.3 Domain Message Authentication Reporting & Conformance

Domain Message Authentication Reporting & Conformance (DMARC) is an email authentication, policy, and reporting protocol which builds on SPF and DKIM protocols. It works by publishing policies for recipient handling of email authentication failures, and reporting from receivers to senders, to improve and monitor protection of the domain from fraudulent email [48]. DMARC, specified in RFC 7489 [49], is in the process of being adopted by the IETF. DMARC ensures that legitimate email is properly authenticating against established DKIM and SPF standards, and that fraudulent activity appearing to come from domains under the organization's control is blocked [47]. DMARC is depending on a proper implementation of SPF and DKIM. To pass DMARC a message must pass SPF authentication and SPF alignment and/or DKIM authentication and DKIM alignment [47]. For SPF, the message must pass the SPF check and the domain name in the *From:* header must match the domain name used to validate SPF (must exactly match for strict alignment, or may be a sub-domain for relaxed alignment - which is the default). For DKIM, the message must pass the DKIM check and the domain name of the valid signature must align with the domain name in the *From:* header (must exactly match for strict alignment, or must be a sub-domain for relaxed alignment) [50]. Even if SPF and DKIM pass authentication, DMARC will still fail if the *From:* headers are not aligned. Depending

on the policies set by system administrator, emails that do not pass DMARC can be quarantined. Since DMARC builds on SPF and DKIM, with its challenges for passing authentication, the possibility of legitimate email being blocked will still be there. This could be the reason some choose not to implement DMARC. SPF, DKIM and DMARC on top of these two, will secure your email identity. However, it brakes the functionality of using mailing lists. Because, mailing lists change messages by adding headers or content [51]. There are possible ways of improving these email authentication protocols, but as of now they are not set into operation [52], [53] and [54].

### 2.2.4 Spam filters

One of the continuous problem areas with email is the amount of spam, also known as unsolicited email, users receive daily. Spam is not necessarily dangerous, but can annoy users. The statistics in figure 2.4 shows the share of global spam volume as percentage of total e-mail traffic as of December 2016, sorted by month. As of December 2014, spam messages accounted for 66.41 percent of e-mail traffic worldwide. This share decreased to 61.33 percent in the most recently reported period [1].

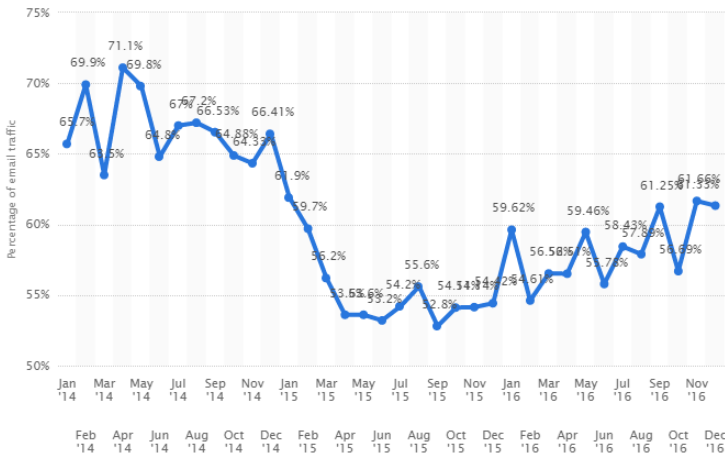


Figure 2.4: Email statistics 2014-2016 [1].

Email spam filters, or anti-spam software, process incoming emails according to specific criteria. This can vary from one spam filter to another, meaning that one type of spam could pass one spam filter, and be blocked by another. These different criteria range from the sender email and IP address to specific content, message characteristics, and format of the email. Some spam filters even add email message headers to the email based on the results from the spam filters called anti-spam stamps. One example of this is Microsoft's anti-spam stamp, X-MS-Exchange-Organization-

SCL<sup>7</sup> [55], and the anti-spam message headers X-Forefront-Antispam-Report<sup>8</sup> and Authentication-results<sup>9</sup> [56]. Results from these different anti-spam message headers and values can then be used for filtering purposes.

The spam confidence level stamp and its rating is assigned each email that is received by domain email servers. SCL value range from 0 through 9, including -1. A value of 0 indicates an extremely low probability that the message is spam. While the highest value 9 indicates an extremely high probability that the message is spam [57]. SCL value of -1 indicates that the email has bypassed anti-spam scanning because the sender address or domain name is white-listed<sup>10</sup>. The SCL value can be used for email filtering. Emails with low SCL can be delivered to end-users' inboxes, while email with high SCL can either be blocked or quarantined and delivered to end-users' spam folders.

### 2.2.5 Anti-malware protection, malicious attachments and URLs

An added solution to the different possible types of spam filters, is anti-malware protection. This type of email filtering is primarily aiming to prevent emails with malicious attachments, virus, code or URLs from passing through to end-users' inboxes. Several of these email filtering types of software use sandboxing<sup>11</sup> technology to safely run email attachments before delivering the email to end-users' inboxes. If the email filtering software identifies attachments with, for instance running code or password protected files, the email will be blocked and not delivered. This also applies to URLs in emails. However, threat agents have started using other ways of avoiding these types of email filtering software. One method seen lately has been that threat agents using URLs from trusted domains like Google or Dropbox where the URLs have been linked to malicious code. Another, more sophisticated, way of operating is when the malicious code or malware is developed to know when it is in a virtual environment like sandboxing. And the code only runs and starts infecting when it is on the victim's computer. Anti-malware protection is important and removes a lot of different possible attacks. But, as in the latest ransomware attacks, anti-malware protection does not provide a hundred percent security.

---

<sup>7</sup>Spam Confidence Level (SCL), which displays a rating of the message based on its content.

<sup>8</sup>message header with results from Exchange Online Protection (EOP), which is a Microsoft email filtering service

<sup>9</sup>message header with the results of email authentication protocols SPF, DKIM and DMARC

<sup>10</sup>earlier marked as safe by users or system administrators

<sup>11</sup>a sandbox is a security mechanism for separating running programs. It is often used to execute untested or untrusted programs or code, possibly from unverified or untrusted parties without risking harm to the host machine or operating system.

### 2.3 Human factors

User awareness is important. Recognizing email scams beyond what technical solutions detect can only be done by human users. To protect from these scams users should understand what they are, what they look like, how they work and what they can do to avoid them [3]. Some indicators can often be found by looking at the message to see if it contains a mismatched address, the address contains a misleading domain name, contains poor spelling and grammar, it asks for personal information or if it contains an offer that seems too good to be true [58]. However, malicious emails and its contents evolve. Today, more and more emails are found to have good grammar, language and the emails look authentic. Still, security awareness often comes down to a user's risk perception. Risk perception refers to the judgment that people make about characteristics and severity of a risk [15]. Results from the study [15] conducted by NorSIS on Norwegian Cyber Security Culture show that security education does not play a significant role in how people perceive risks. However, people who are interested in ICT and technology are more confident that they can assess what is safe. What is problematic is that the study finds how people assess risk is mostly subjective based on individual experiences and recent past. This could effect security awareness and how general information security education is conducted.

Email users can and will most likely be fooled, but at the same time by using employees' awareness there are ways of reducing the risk. Email security is in some way a double-edged sword. Having a too strict set of rules can potential lead to emails with business opportunities being filtered and deleted. While a policy opening for everything would fill every inbox with legitimate including malicious emails. Both options are to the disadvantage of the end-user using email as means of communication. There needs to be a balance between functionality and security. There are vulnerabilities with email as a way of communication and this is not likely to change in the foreseeable future. No matter how many technical security measures are implemented, a fail-safe solution on how to secure email and avoid being compromised due to malicious email is difficult to achieve. Especially, when there should be a balance between user functionality and security. People can be fooled and human error can occur, but at the same time there is only so much technical solutions can achieve. Reading, understanding the content and identifying malicious emails that have passed through the security mechanisms can only be done and reported by people. To accomplish this, it is important to have an elevated level of risk perception and security awareness.

# Chapter 3

## Method

The following chapter presents the separate phases of work and choice of method. There are two main methods within the scientific method of research, qualitative and quantitative. These will be explained in the following sections.

### 3.1 Qualitative method, literature and case study

Two qualitative methods were chosen to be used simultaneously for the first phase of the thesis work. One of the chosen methods was a literature study to collect a significant amount of information and literature on the topic area. The aim of the literature study was to provide an overall picture of the topic and the problems addressed. In a literature study, there will always be room for error in the interpretation of the contents in relation to what the authors have tried to communicate. So, a continuous dialogue with the professor and supervisor has been important throughout the process and work to prevent any misunderstanding. Some of the information used in the preliminary work on email threats and risks have come from Norwegian sources. The Norwegian Intelligence Service's (NIS) annually report [10] with assessment on current security challenges and The Norwegian National Security Authority's (NSM) annually report [13] on ICT risks. The first report looks at the global security situation, and brings attention to possible threat agents in the scope of intelligence threats against Norway. The latter report highlights more general email risks, threat agents, most common vectors of attack, technological and human vulnerabilities and possible ways of mitigating them. Reports from The Norwegian Business and Industry Security Council's (NSR) Norwegian Computer Crime and Data Breach Survey 2016, and The Norwegian Centre for Information Security's (NorSIS) The Norwegian Cyber Security Culture have also been helpful. The first report [14] is an annually survey on information security, privacy, and cybercrime, while the latter report [15] is a survey on cyber security culture. A security blog which have been to great help is Terry Zinks Security Talk [59], which discusses many of the challenges email is facing.

The second method was to conduct a case study involving email and email security within an organization. This helped clarify the use of and interaction between an email system and its users. It has given a context to the topic and the addressed problem better than what is possible only through a literature study. The case study has given information on how important email security is to an organization and its business assets. This includes how the use of security technology is a vital part of how systems are implemented and used. However, and probably the most important thing in all of this, is the human factor. The case study has shown that an organization must make email security the least common multiple in all aspects from higher management to employee. Because, when the technical security measures implemented does not stop malicious attempts, it is up to the employee's risk perception and level of security thinking to raise alerts. The use of a case study has been a good strategy and given answers to questions that have been raised on how and why a solution could and should be implemented.

The qualitative methods conducted, provided knowledge and understanding regarding how email is used in an organization, on threat actors, risks, vulnerabilities and possible attacks. As well as how human risk perception and security thinking is vital to improve security towards email. The disadvantage of the qualitative method in relation to the quantitative method is that the first does not have measurable empirical data, making it difficult to generalize the results. However, a qualitative method was more suited to the start phase of the thesis.

### **3.2 Quantitative method, data collection and testing**

In the start of the second phase of the thesis two quantitative methods were chosen to be used simultaneously. This was to conduct data collection of reported emails for later to derive hypotheses and do testing. The first method was a questionnaire which looked at some of the reported emails found in the dataset. The purpose for this questionnaire was to have some statistical data on what users in the organization found suspicious on a set of different emails. This gave some understanding into the users' reasons and reasoning for reporting a specific email based on some of the email headers and content. A weakness with the questionnaire it that participants are unable to state whether they would have reported the email in question, if it had arrived in their own inbox. Another weakness is that the questionnaire should have asked for users email address in the responses. This could have later been used to compare these results to that of the data analyses and hypotheses testing on the complete dataset of reported emails. However, conducting an anonymous questionnaire could be perceived as less imposing than that of the opposite. It was more important to have some data from a wider range of sources, than to identify the sources and possibly have less data from the questionnaire. This was the primary



source of data for understanding why emails are reported, and the secondary source of data for the hypotheses testing.

The second method was data collecting from emails reported by users at the organization of the case study. This was the primary source of data for the data analyses and hypotheses testing. Quantitative methods raise questions. These can be used to derive possible hypotheses from the knowledge and observations gained from the qualitative methods in the first phase, and the data from the quantitative methods in the second phase. Key areas to consider in this phase was how data was going to be collected and visualized. The third and final phase was data analyses and hypotheses testing. An applicable feature with hypotheses testing is its iterative process. It allows for modification of the hypothesis if a test fails, or the option to completely discard the hypothesis and come up with something else. *Theories are proposed, and then experiments are designed and performed to test those theories. Based on the measured results, the theory is either rejected or confirmed* [60]. Doing so several times and with different criteria has resulted in measurable numerical data. This can then be used for suggesting solutions and further work with email security in an organization.



# Chapter 4

## Design and implementation

This chapter documents how the email dataset was retrieved and parsed. Next, it documents the design and implementation of a web based framework to visualize data. Finally, the chapter documents how data analyses and hypotheses testing were conducted during the thesis.

### 4.1 Retrieving and parsing emails

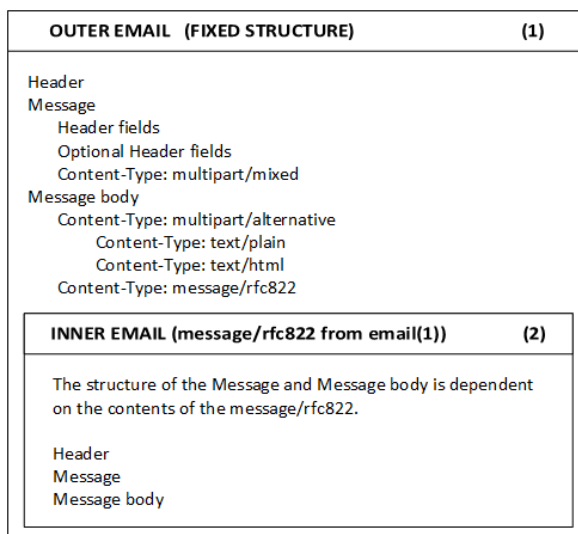
To retrieve and parse the emails that would later become the dataset, several steps were taken. The first step was to look at how and what type of data was generated when users report suspicious emails in their email client. Secondly, how could these emails be extracted from the organizations email system? The third and last step was to consider how the content of the emails could be parsed, which output format was suitable and how this data was going to be stored. The later was also important for how data analyses and hypotheses testing was conducted.

The main goal of conducting a case study was to see how email security within an organization use employees to recognize suspicious emails and at the same time boost security awareness. The organization has implemented a functionality in users' email clients for reporting suspicious email as a part of a security culture initiative called *OJ!*. This functionality allows users to report emails they find suspicious by a simple click of a button. The intention behind this functionality is to identify malicious emails that have passed the technical security measures and use this to alert other users from being tricked.

Using this functionality will report the suspicious email to a mailbox administrated by IT security personnel, and it will at the same time remove that suspicious email from end-users' inboxes. It works by creating a new email message, from now on referred to as the *outer-email*, and adds the suspicious email, *inner-email*, as an RFC

822<sup>1</sup> attachment. It does so to preserve the emails header and message header fields of interest. Basically, it is an email message within a new email message, both a .msg file extension. The .msg format is used for an Outlook Mail Message file<sup>2</sup>. This file extension usually entail message saved and created in Microsoft Outlook [61].

An example of how the structure of the *outer-email* and *inner-email* is shown in figure 4.1. The structure of the *outer-email* is mostly fixed, meaning that in most cases contains an RFC 822 attachment in the. In cases where the *inner-email* had contents filtered by the anti-malware software or similar, no RFC 822 attachment would be enclosed. The structure of the *inner-email* depends on its content, and example is shown in figure 4.2 where the *inner-email* only contains plain text and html.



**Figure 4.1:** Example of the email structure for the emails that are parsed.

As mentioned, the reported emails are sent to a mailbox administrated by IT security personnel. To retrieve these emails and automate the parsing a test domain with email service was set up as shown in figure 4.3. An email address to the test domain was added in the email reporting functionality to send the reported emails. Two rules were implemented for this email mailbox. The mailbox only accepted email coming from the reporting domain, the organization domain. And no sending or forwarding. These were set so that no other email would be accepted, or any of

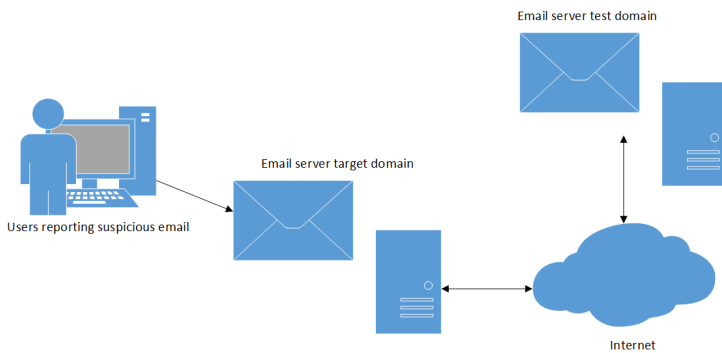
<sup>1</sup>RFC 822 is the ARPA standard for the format of Internet text messaging

<sup>2</sup>many programs are capable of opening outlook messages provided they are compatible with Outlook Mail Message through use of Microsoft's Messaging Applications Programming Interface (MAPI) [61].

INNER EMAIL (message/rfc822 from email(1))	(2)
Header	
Message	
Header fields	
Optional Header fields	
Content-Type: multipart/alternative	
Message body	
Content-Type: text/plain	
Content-Type: text/html	

**Figure 4.2:** Example of email structure where the message/rfc822 only have text, no attachments.

the reported emails were sent or forwarded by mistake. At a later point in time it became clear that the domain and email service did not support the .msg proprietary Microsoft Outlook file extension. It only supported the .eml file<sup>3</sup> extension which are stored email messages in plain text formats [62]. So, to automate the parsing of the reported emails, the .eml file extension needed to be used.

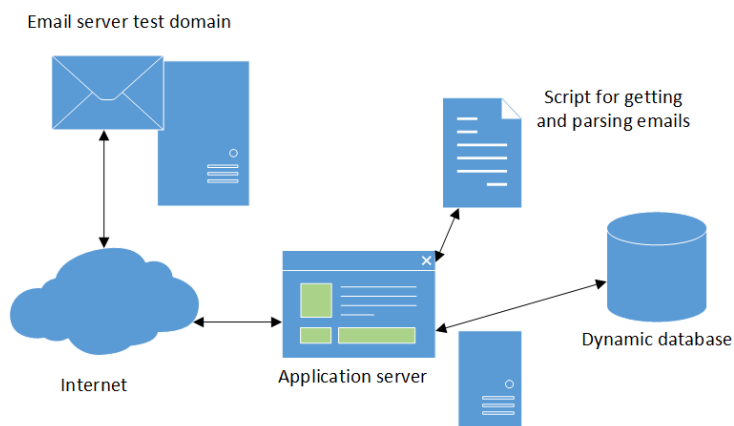


**Figure 4.3:** Shows the targeted domain where users report suspicious email, these emails are so sent to the email server of the test domain.

After the reported emails arrive at the email server of the test domain, and their content with header and message header fields are parsed, the metadata needed to be stored. The output data format of the parsing, the metadata from the reported emails, needed to be stored for later use and tests. This led to storing the data in a dynamic, NoSQL database, a more flexible database. This type of database was better suited for the web application framework used later and the output format from the parsing. This was done because of the varying content and different message

<sup>3</sup>is in compliance with the standards for electronic mail headers or otherwise known as RFC 822, EML files can be used with various applications, servers and email clients. This means that EML files can be viewed without restrictions by other operating systems and different browsers preferred by users [62].

header fields, making every email unique. The output format was chosen to be JavaScript Object Notation (JSON) which is an open and text-based data exchange format that provides a standardized data exchange format suited for web applications [63]. JSON uses nested key-value pairs, where the key describes data and the data is stored as the value [64], which worked nicely to save email and message header fields. In this scenario, each email header and message header field became keys and the data became their corresponding values. This made object traversal easy for testing purposes later in the thesis. Each email was parsed and then saved to a JSON document in the database. The database used was MongoDB, which is an open-source document-oriented database [65]. It suited storing email data as of the dynamic contents. A basic design of the environment can be seen in figure 4.4.



**Figure 4.4:** Shows the basic design of the environment for retrieving, parsing email headers and storing them to the database.

The script for retrieving and parsing emails shown in figure 4.4 has multiple tasks. It is written in python<sup>4</sup> do its simple and easy to use syntax. There are also a lot of sources and libraries including examples of parsing emails using python. The script runs either from the web application or direct from the command line. It could have been proactive to run the script when emails had been received to the mailbox or periodically at fixed times. The latter is better for an operational environment either as a Windows service<sup>5</sup> or a Cron<sup>6</sup> job. It could also have been set up to be alerted via IMAP<sup>7</sup> when emails arrive to the mailbox and then run the parsing script.

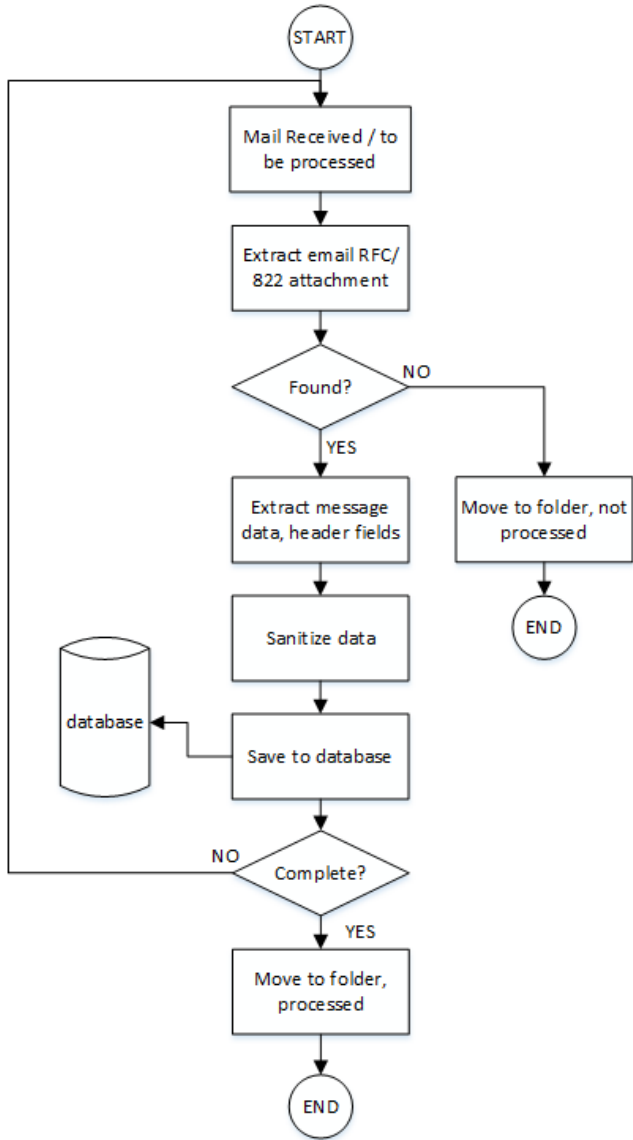
<sup>4</sup>object-oriented programming language

<sup>5</sup>applications that run in the background of the operating system

<sup>6</sup>Cron is a time-based job scheduler in Unix-systems

<sup>7</sup>standard email protocol which enables user to view and manipulate emails

A flowchart of how the script works is shown in figure 4.5. It initially starts by connecting and logging in to the email account on the test domain's email server using a subclass of the IMAP4 protocol client *imaplib* library. This subclass, `IMAP4_SSL`, connects over an SSL encrypted socket [66]. It then uses IMAP4 objects from the library to select the email folder of interest. Every reported email is sent to the same folder. The script continues to run if there are any emails found in the Inbox folder. It takes each email and sees if it contains an RFC 822 attachment. Emails without this attachment are not processed and moved to a different folder. The emails with correct attachment are processed using the python eml parser module for parsing .eml-files and returning various information found in the email [67]. Some email headers and message header fields from the *outer-email* and all this information from the *inner-email* is for each email first sanitized and secondly stored in JSON documents and saved to the MongoDB database. If the process completes, processed email will be moved to a folder, and the script terminates the IMAP4 by closing the connection and logging out.



**Figure 4.5:** Flowchart parsing emails and saving parsed data to the database.

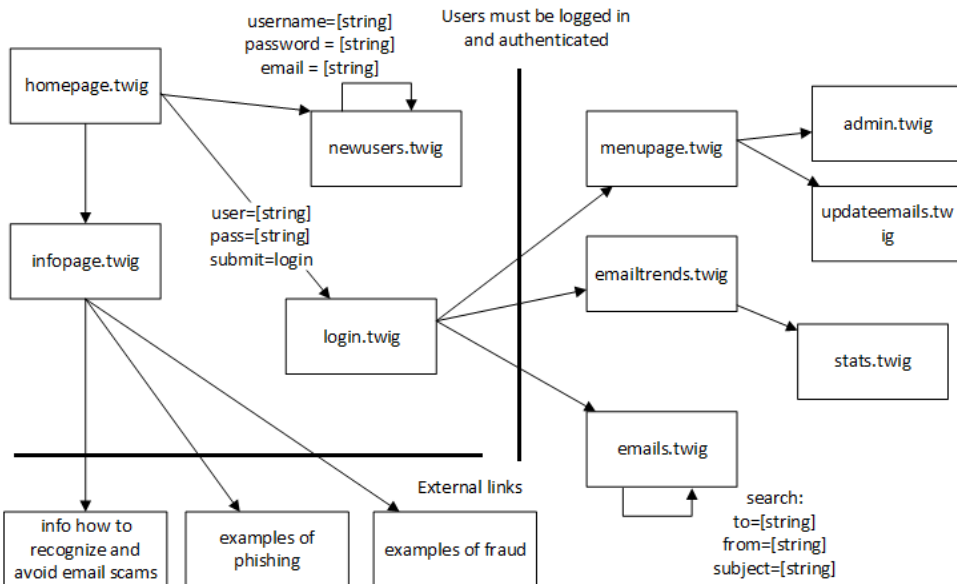
## 4.2 Environment for visualizing data

To visualize data a web based framework was set up. The thought behind this was to give system administrators and users the ability to have access, overview and statistics on the vast amount of reported email through a simple graphic user interface. The design of this framework was closely based on Ben Shneiderman’s principle on



information visualizations, *overview first, zoom and filter, then details-on-demand* [68]. Users of this web application are given an overview of the complete collection of reported emails via a page named *emails*. This page also enables the users to zoom and filter based on a search functionality. Statistics are also provided in the *emailtrends* page. If the users want more detailed information about specific emails they can go directly to the email by clicking the id of that email.

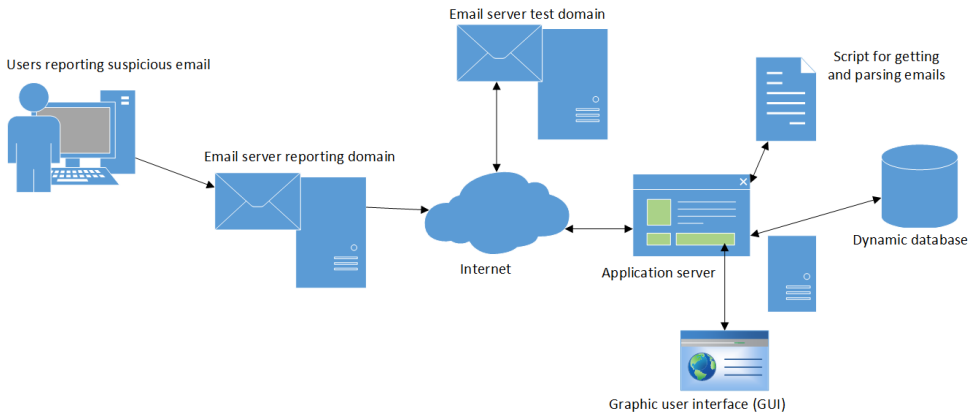
Figure 4.6 shows a page-map over the web application with its different trust zones and user inputs. This layout was used in the implementation of the web application following principles from the Open Web Application Security Project (OWASP) top 10 application security risks. Key areas were the different pages with user inputs to prevent both injections, database injections, and cross-site-scripting [69]. Countermeasures to meet these possible risks are proper validation of user input, and the use of prepared statements for the interaction between users and the search functionality for queries against the database.



**Figure 4.6:** Pagemap, showing different trust zones and user inputs.

The web application uses a Hypertext Pre-processor (PHP) micro framework called Slim that allows for quick and efficient writing of simple and powerful web applications [70]. It can easily be run with PHP's built in web-server. PHP is a widely-used open source general-purpose scripting language that is especially suited for web development where the main goal is on server-side scripting [71]. PHP code is

executed on the server side and generates twig templates, similar to HTML, which is then sent to the client. Twig is a flexible, fast and secure template engine for PHP [72]. These tools for building a web application are simple and efficient to use. The script for retrieving, parsing and saving data from the reported emails was run from the web application using JavaScript<sup>8</sup> to access the script on the server-side. The complete environment can be seen in figure 4.7. Emails are reported by users and ends up at the email server of the test domain. The script runs from either command line or web application to retrieve and parse email data before storing them to the database. Data objects are sorted in repositories on the server-side and controllers present data visually by the web application through a graphic user interface.



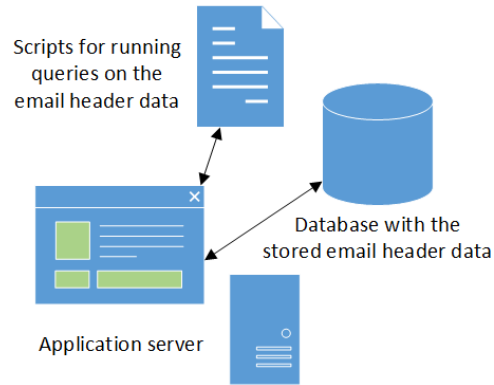
**Figure 4.7:** Complete test environment. Reporting domain on the left side, test domain and email server in the middle and test environment on the right side.

### 4.3 Obtaining measurable data from the dataset

Some testing needed to be done in order to obtain measurable data from the dataset. The main goal was to have a simple and efficient method to perform multiple tests on the dataset with the possibility of using different test criteria. This was achieved through several scripts, each run separately. It worked by connecting to the the MongoDB database using the Python PyMongo module, instead of having to work in the MongoDB command shell. PyMongo is a Python distribution containing tools for working with MongoDB, and the recommended way to work with MongoDB from Python [74]. After successfully connecting to MongoDB database and choosing the correct data collection (db = emails), the scrips run a cursor to traverse all the objects containing email data. After specifying the key, the email header or

<sup>8</sup>JavaScript (JS) is a lightweight interpreted or JIT-compiled programming language [73].

message header field, of interest the results from the script were printed directly to the command line. The basic setup of this testing can be seen in figure 4.8.



**Figure 4.8:** Basic setup of the testing on the dataset.



# Chapter 5

## Questionnaire on suspicious emails from the dataset

This chapter presents the questionnaire used to assess how users choose to report emails.

The questionnaire was based on twelve potential malicious emails found in the dataset. It covered the *human factor*. How an employee would recognize potential malicious emails (phishing, scam, fraud, malware) that have passed the technical security measures. Participants were given several options to choose from on why they would have reported the emails in question. These options were related to different headers, message body text, attachments and links found in the emails. The questionnaire was anonymous and conducted over two weeks with a total of 83 responses. Participants were recruited with the help of the professor through internal communications in the organization. The twelve emails had varying SCL values from no value, 1 and up to 9. The SCL value gives the system administrator an option for filtering emails. A low SCL value found in the message header indicates if the reported email would have passed the spam filter and delivered to end-users' inboxes. The emails with high SCL value would most likely be filtered as spam and delivered to the spam folder.

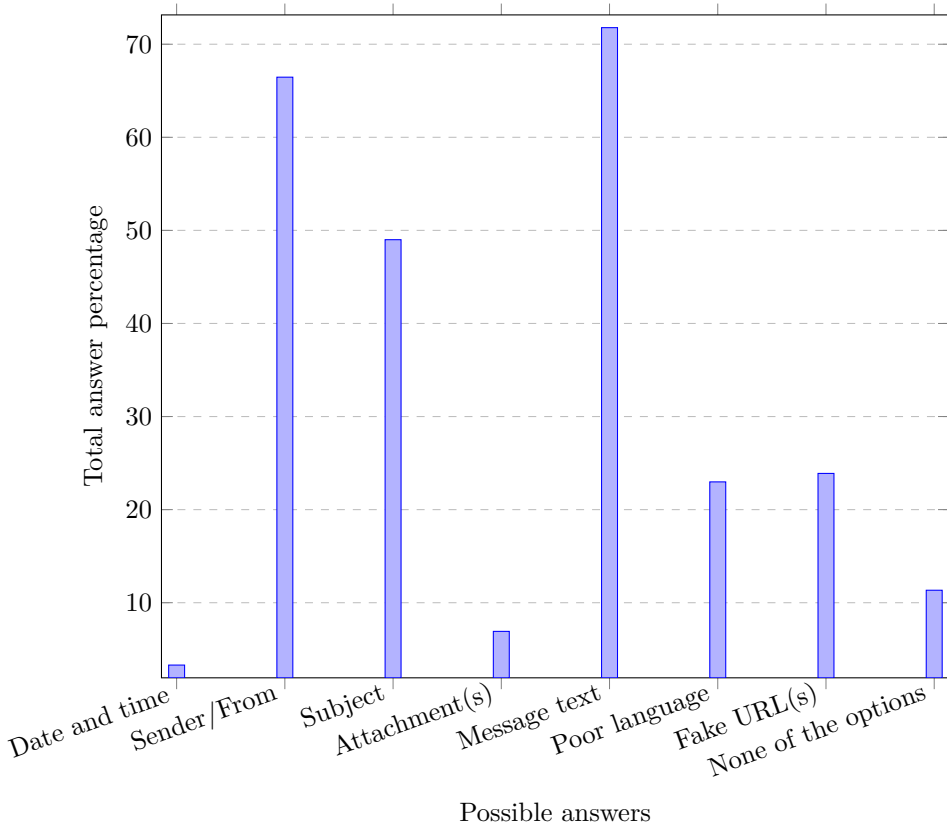
### 5.1 Overall results

The majority of respondents chose the following three options as the main reasons for reporting an email.

1. the message text in general
2. the sender or from header field, unknown or bogus email address and name
3. the subject of the email

All of the emails did not have an attachment or URL in its content. So, when looking at the responses individual for emails with attachments or URLs, one would see that these are also options respondents would choose for reporting an email. A general impression of the overall feedback is that the considerations taken on each email are

subjective. Whether the respondent would have chosen to report the email or not, is based on own experiences with similar scams, or if the content is of relevance. These are understandable factors. Nevertheless, it would be advantageous if they would also see it objectively and report emails they found dangerous which potentially could trick other users. Some of the respondents state to have different criteria for disregarding a message as a potential fraud and for reporting it – reserving the latter for those fewer instances where they think there is a real danger. Most of the time, they only take a quick look at the contents before discarding it as either spam or fraud. So, in these cases they have not really considered all the options that have been provided in the questionnaire. If in doubt as to whether they should react to the message beyond reading it, they then consider clickable links like fake URLs and attachments found by looking closer at the emails. **Results:** Percentage overall results, in descending order: Message text (71.78), Sender/From address (66.46), Subject (49.0), Fake URL(s) (23.89), Poor language (22.98), None of the options (11.34), Attachment(s) (6.92) and Date and time (3.3).



**Figure 5.1:** Total answers from the questionnaire.

## 5.2 Individual questionnaire responses and results

**Email number 1**, shown in figure 5.2, has only been reported once with SCL value 1. However, seven other reported emails found in the dataset have the same sender address, with slightly different subject text. All were received within two months. These have different SCL values, from 1, 5 and up to 7. This email is a typical business opportunity or sales scam email, and can be easily identified using open sources like Google. The sender domain used is a free email provider based in Asia known for being used in this type of scams. The email message text has poor language and lack of coherent text. However, there is no immediate danger, and it can be dismissed as spam. General comments from the responses highlights the email sender as unknown, the uninteresting topic and its lack of relevance. Some also refer to the very general text and the fact that the email comes from China and similar to other messages received over some time. Several would just have deleted the email and not report it.

**Results:** Percentage results from the first email, in descending order: Sender/From address (84.3), Message text (61.4), Poor language (56.6), Subject (33.7), None of the options (8.4), Attachment(s) (3.6), Fake URL(s) (2.4) and Date and time (0.0).

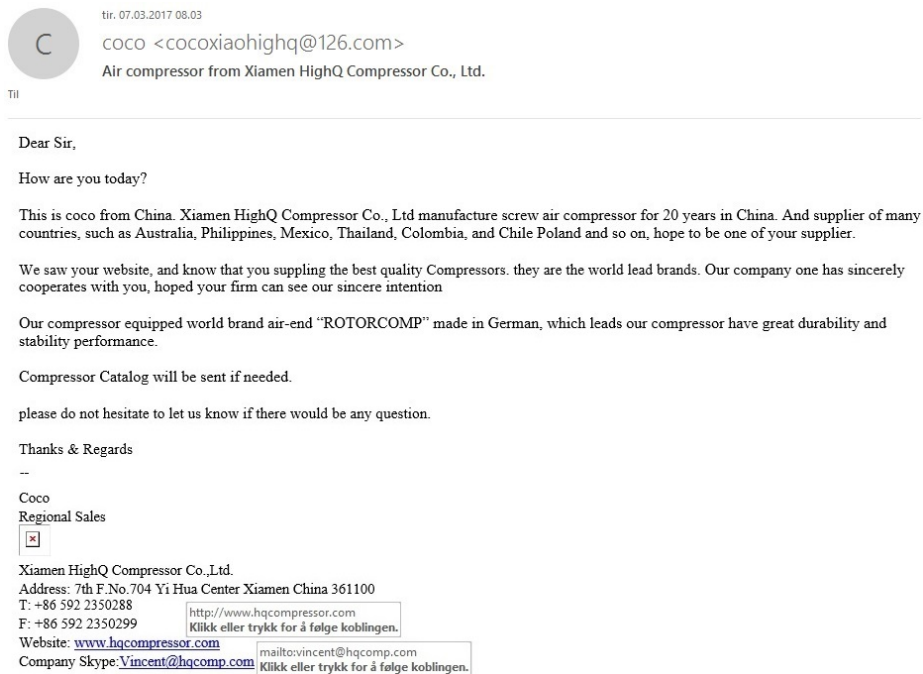
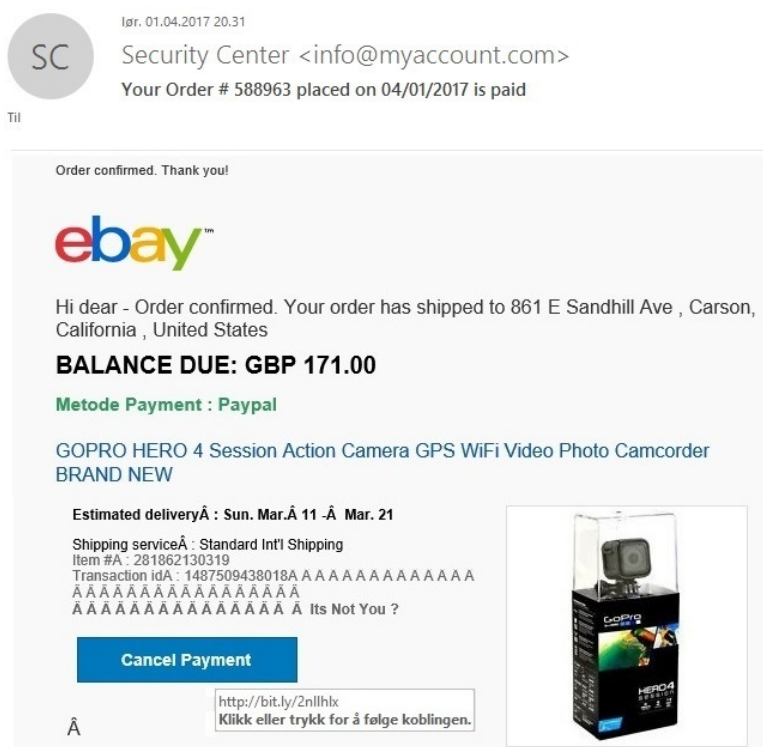


Figure 5.2: Questionnaire, email number 1.

**Email number 2**, shown in figure 5.3, has only been reported once with a SCL value of 1. This email is a phishing email claiming to be an order confirmation from Ebay. It does not try to spoof the domain name, but it uses an own domain name that passes the email authentication protocols. The content of this email tries to act on users reacting to having ordered something which they might not have done. This could trick the user to *cancel* it by clicking a link. Looking closer at the link shows a questionable URL. It could be an email phishing for sensitive information like card details, or a link which could run malicious code. Potential high immediate danger. General comments from the responses brings attention to that the users have not ordered something from Ebay, they do not have an account on Ebay, familiarity with similar emails and the message text which is very general. Some found it too obvious to report it.

**Results:** Percentage results from the second email, in descending order: Message text (78.3), Sender/From address (68.7), Fake URL(s) (44.6), Subject (42.2), Poor language (28.9), Attachment(s) (8.4), Date and time (4.8) and None of the options (2.4).

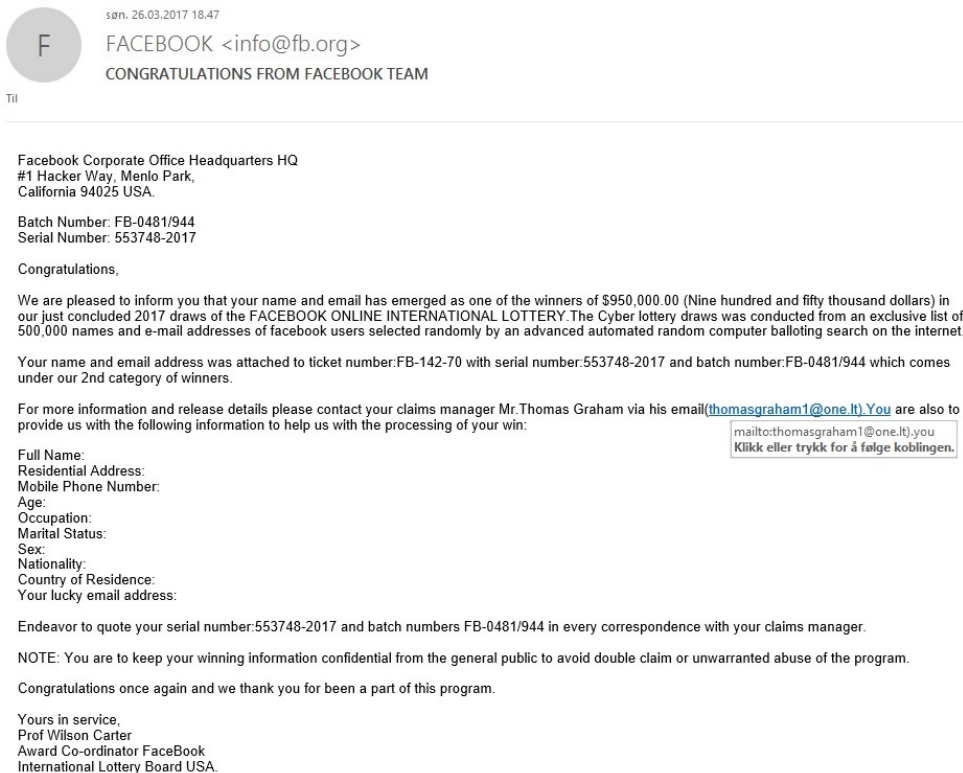


**Figure 5.3:** Questionnaire, email number 2.



**Email number 3**, shown in figure 5.4, has been reported eight times within two days. SCL value on these eight range from 5 to 7. Depending on the level on the spam filter, these should have been delivered to the spam folder. Meaning that these eight emails were all reported from end-users' spam folder. This is a typical *old fashioned* lottery scam asking for personal information. It claims to be from Facebook, but it does not do so by trying to spoof the domain and email address of Facebook. General feedback lists this email as a typical *to god to be true* untrustworthy email scam which asks for personal information. Some feedback points out the lack of legitimate Facebook email address and domain.

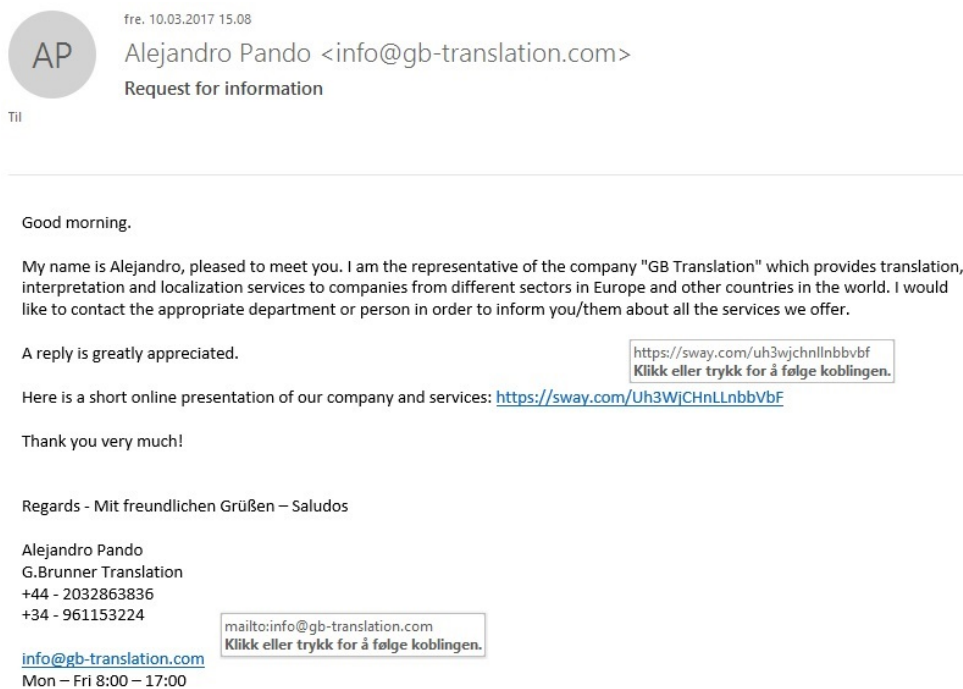
**Results:** Percentage results from the third email, in descending order: Message text (95.2), Sender/From address (61.4), Subject (61.4), Fake URL(s) (19.3), Poor language (12.0), Date and time (3.6), Attachment(s) (2.4) and None of the options (0.0).



**Figure 5.4:** Questionnaire, email number 3.

**Email number 4**, shown in figure 5.5, has been reported twice. They have both SCL value 1 and received within the same day. This email is hard to classify as nothing else than general marketing and perceived as spam. But it is hard to verify and clicking on the URL is in general not an appropriate solution when the sender is unknown. The URL is linking to a site, <https://sway.com>, which is site for creating and sharing presentations. But it is hard to identify this as malicious. To be sure, one should have had a virtual sandbox environment trying to access the content. General feedback lists this email as annoying spam email, and general marketing from someone unknown to the respondents.

**Results:** Percentage results from the fourth email, in descending order: Message text (44.6), Fake URL(s) (42.2), Sender/From address (28.9), None of the options (24.1), Subject (22.9), Poor language (3.6), Attachment(s) (2.4) and Date and time (0.0).



**Figure 5.5:** Questionnaire, email number 4.

**Email number 5**, shown in figure 5.6 has been reported twice. One has SCL value of 7, the second has no SCL header or value. It is not an internal email, so the second email should also have been given SCL with value 7. Reason for this could have been downtime on the receiving side, not adding the SCL header and value. The two emails were both received within the same day. This is a typical *you have won* winning notice scam email. The sender has tried to appear as a legitimate email from Google without spoofing the domain. The attachment has not been removed by anti-malware software, but there could still be malicious content for example URLs to malicious sites, this could also be some sort of phishing. General feedback lists these emails as common fake winning notice. Most of the respondent would have deleted these emails and not reported because they are too obvious.

**Results:** Percentage results from the fifth email, in descending order: Sender/From address (89.2), Message text (86.7), Subject (68.7), Attachment(s) (59.0), Poor language (20.5), Fake URL(s) (12.0), Date and time (4.8) and None of the options (0.0).



Dear Google User,

Congratulations,

You have been rewarded by Google Corporation, attached to this email is your notification package for being an active user of Google.

Sincerely,

Matt Brittin

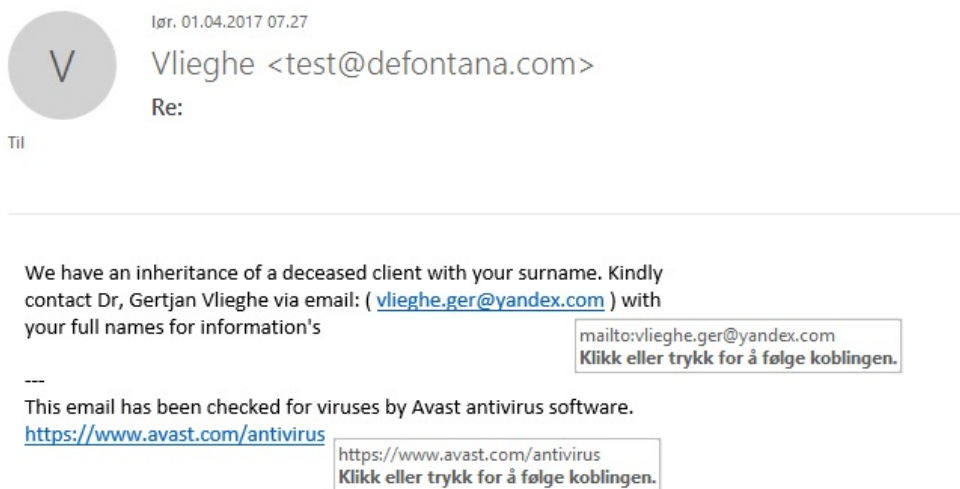
VP Operations

2017 Google Inc

**Figure 5.6:** Questionnaire, email number 5.

**Email number 6**, shown in figure 5.7, has been reported once. No SCL value. Typical style of scam email, often referred to as *Nigeria letters*. General feedback lists this email as to obvious to warrant consideration of reporting. Classified as spam only based on the first few sentences in the message body.

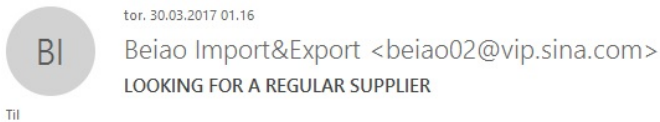
**Results:** Percentage results from the sixth email, in descending order: Message text (90.4), Sender/From address (83.1), Subject (68.7), Poor language (28.9), Fake URL(s) (14.5), Date and time (8.4), Attachment(s) (2.4) and None of the options (0.0).



**Figure 5.7:** Questionnaire, email number 6.

**Email number 7**, shown in figure 5.8, has been reported six times. All have SCL value of 1, and they have been received within one week. The sender domain used is a free email provider based in Asia known for being used in this type of scams or spam emails. General feedback lists this as email spam and not worth reporting.

**Results:** Percentage results from the seventh email, in descending order: Message text (83.1), Sender/From address (66.3), Subject (48.2), Poor language (36.1), None of the options (9.6), Fake URL(s) (4.8), Date and time (2.4) and Attachment(s) (0.0).



Dear sir/madam,

I'm the business manager of Beiao Import&Export company. Recently I'd like to import many kinds of products. For the reason I'm very interested in your product, can you provide me a detailed quotation on your product including types and prices so that we can expect a long-term cooperation.

mailto:beiao166@vip.sina.com  
Klikk eller trykk for å følge koblingen.

Send your quotation to [beiao166@vip.sina.com](mailto:beiao166@vip.sina.com). Looking forward to your reply.

Sincerely,

Wang Xiaohui

Business Manager, Beiao Import&Export Co. Ltd. China

**Figure 5.8:** Questionnaire, email number 7.

**Email number 8**, shown in figure 5.9, has been reported ten times within the same day. Four of them with SCL value 5, five with value 7 and one without. This is an old-fashioned fraud, *guaranteed* loans or credit and asking for personal information. Typical phishing. General feedback lists this as classical phishing and information gathering attempt. Some are familiarly with similar scams.

**Results:** Percentage results from the eighth email, in descending order: Message text (89.2), Sender/From address (85.5), Subject (66.3), Poor language (41.0), Fake URL(s) (3.6), Date and time (1.2), Attachment(s) (1.2) and None of the options (0.0).



Til

---

This is Unicredit Loans Group, we offer 3% for business loans and personal loans, you fill in the form if interested.

Full name:  
 Gender:  
 Country:  
 Required loan amount:  
 duration:

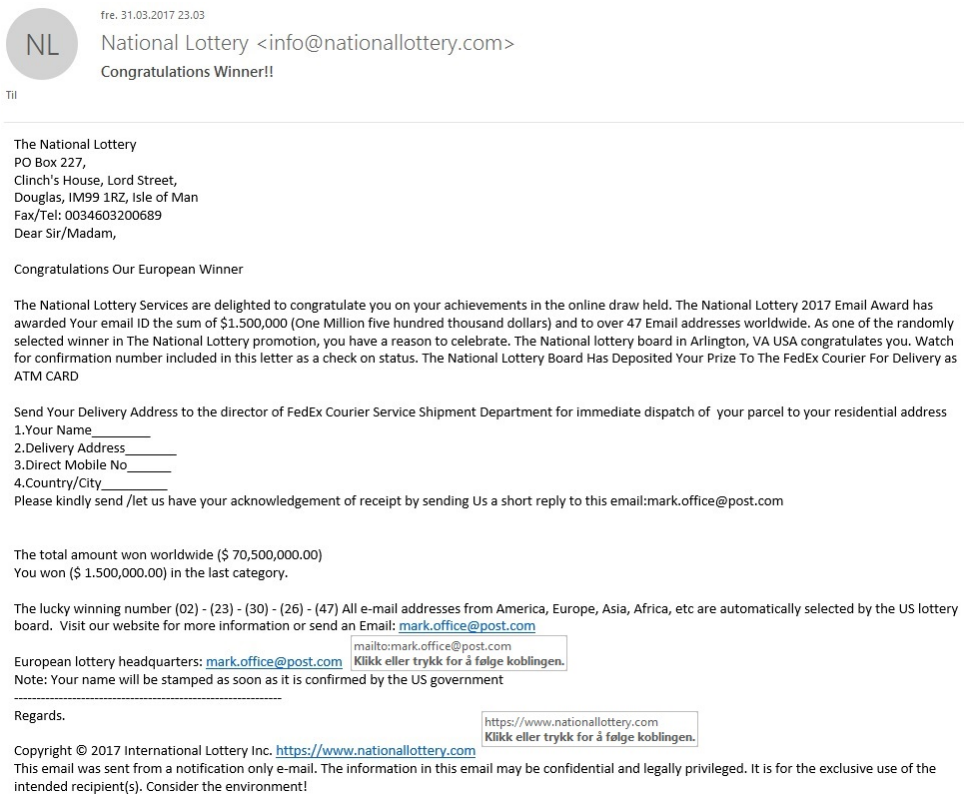
We Await your responds, Meeting your financial needs is our pride.

Veronica Angela

**Figure 5.9:** Questionnaire, email number 8.

**Email number 9**, shown in figure 5.10, has been reported twice within two days. One with SCL value of 6, the second without any header and value. These are examples of fake lottery scam emails phishing for personal information. General feedback lists this as classical lottery scam email and phishing attempt. Some would not even read the content after seeing the subject header.

**Results:** Percentage results from the ninth email, in descending order: Message text (94.0), Subject (68.7), Sender/From address (55.4), Fake URL(s) (9.6), Poor language (8.4), Date and time (3.6), Attachment(s) (1.2) and None of the options (0.0).



fre. 31.03.2017 23:03  
 NL National Lottery <info@nationallottery.com>  
 Congratulations Winner!!

Til

---

The National Lottery  
 PO Box 227,  
 Clinch's House, Lord Street,  
 Douglas, IM99 1RZ, Isle of Man  
 Fax/Tel: 0034603200689  
 Dear Sir/Madam,

Congratulations Our European Winner

The National Lottery Services are delighted to congratulate you on your achievements in the online draw held. The National Lottery 2017 Email Award has awarded Your email ID the sum of \$1,500,000 (One Million five hundred thousand dollars) and to over 47 Email addresses worldwide. As one of the randomly selected winner in The National Lottery promotion, you have a reason to celebrate. The National Lottery board in Arlington, VA USA congratulates you. Watch for confirmation number included in this letter as a check on status. The National Lottery Board Has Deposited Your Prize To The FedEx Courier For Delivery as ATM CARD

Send Your Delivery Address to the director of FedEx Courier Service Shipment Department for immediate dispatch of your parcel to your residential address

- Your Name \_\_\_\_\_
- Delivery Address \_\_\_\_\_
- Direct Mobile No \_\_\_\_\_
- Country/City \_\_\_\_\_

Please kindly send /let us have your acknowledgement of receipt by sending Us a short reply to this email:mark.office@post.com

The total amount won worldwide (\$ 70,500,000.00)  
 You won (\$ 1,500,000.00) in the last category.

The lucky winning number (02) - (23) - (30) - (26) - (47) All e-mail addresses from America, Europe, Asia, Africa, etc are automatically selected by the US lottery board. Visit our website for more information or send an Email: [mark.office@post.com](mailto:mark.office@post.com)

European lottery headquarters: [mark.office@post.com](mailto:mark.office@post.com) <mailto:mark.office@post.com>  
 Klikk eller trykk for å følge koblingen.

Note: Your name will be stamped as soon as it is confirmed by the US government

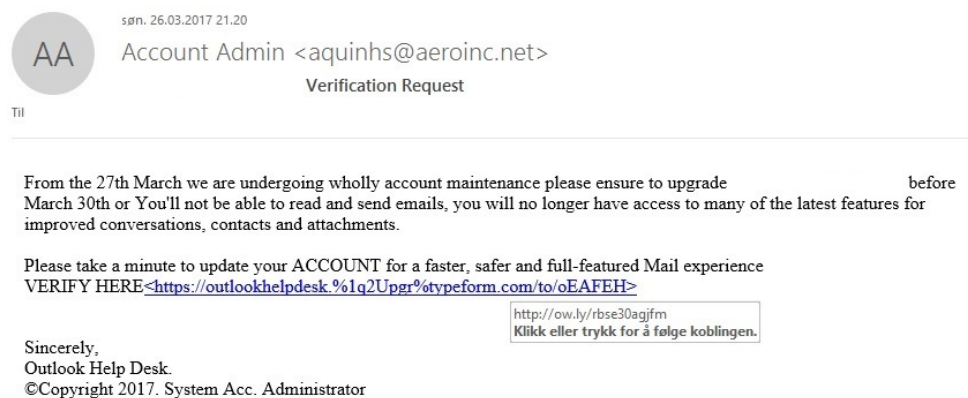
Regards. <https://www.nationallottery.com>  
 Klikk eller trykk for å følge koblingen.

Copyright © 2017 International Lottery Inc. <https://www.nationallottery.com>  
 This email was sent from a notification only e-mail. The information in this email may be confidential and legally privileged. It is for the exclusive use of the intended recipient(s). Consider the environment!

**Figure 5.10:** Questionnaire, email number 9.

**Email number 10**, shown in figure 5.11, has been reported once. SCL value of 7. This is a typical phishing attempt, and similar phishing emails have been reported many times. Messages pretending to concern services users depend on are particularly problematic. Email asking for users to click a suspicious URL. General feedback lists this as a phishing attempt. Unknown sender and a subject asking for verification. Some found the email to obvious to spend time reporting it.

**Results:** Percentage results from the tenth email, in descending order: Sender/From address (89.2), Message text (86.7), Subject (71.1), Fake URL(s) (68.5), Poor language (36.1), Date and time (8.4), Attachment(s) (1.2) and None of the options (0.0).

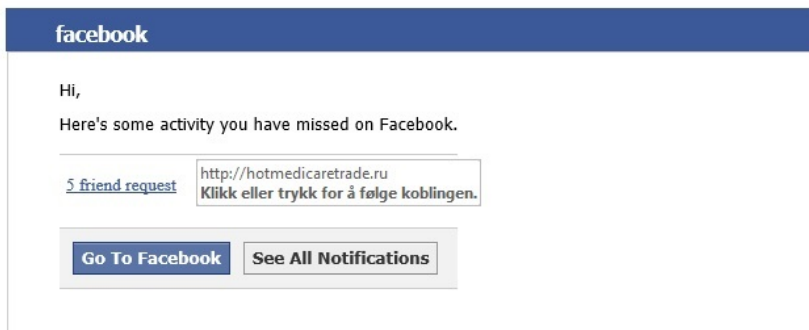
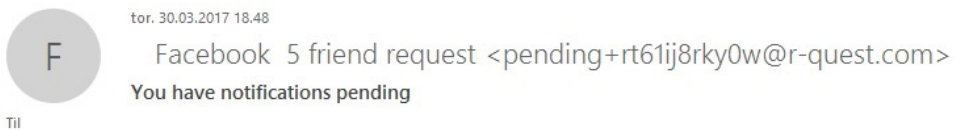


**Figure 5.11:** Questionnaire, email number 10.



**Email number 11**, shown in figure 5.12, has been reported once. There is a total of 23 reported emails with same subject, but with slight differences in the sender email address. These 23 reported emails have SCL values of 5, 6, 7 and 9. Typical phishing attempt trying to disguise as notifications from Facebook. Suspicious URL. General feedback for this email is subjective in the sense of that users report that they are not on Facebook, their account is not linked to the email address used for work or it is not in compliance with the notification settings. Some find this email to be close to that of an authentic email.

**Results:** Percentage results from the eleventh email, in descending order: Sender/From address (83.1), Fake URL(s) (60.2), Message text (50.6), Subject (34.9), Poor language (3.6), Attachment(s) (1.2), Date and time (0.0) and None of the options (0.0).



This message was sent to . If you don't want to receive these emails from Facebook in the future, please click: [unsubscribe](#).  
Facebook, Inc. Attention: Department 415 P.O. Box 10005 Palo Alto CA 94303

<http://www.facebook.com/>  
Klikk eller trykk for å følge koblingen.

**Figure 5.12:** Questionnaire, email number 11.

**Email number 12**, shown in figure 5.13, has been reported three times by the same user within one month. Two of these have SCL value 1, the third has no header or value. This is an internal email sent within the organization, but the SCL value does not reflect this for the two emails with SCL value 1. Email authentication is neither implemented for this sender. Most of the respondents know that this is an internal email. Some do not now that this is an internal email. One feedback says that there are multiple suspicious emails in English which comes from systems within the organization. Some choose to not access legitimate email because they do not know that it is internal email.

**Results:** Percentage results from the twelfth email, in descending order: None of the options (91.6), Fake URL(s) (6.0), Sender/From address (2.4), Date and time (2.4), Message text (1.2), Subject (1.2), Poor language (0.0) and Attachment(s) (0.0).



**Figure 5.13:** Questionnaire, email number 12.

# Chapter 6

## Data analyses and hypotheses testing

This chapter starts by presenting the dataset of reported emails and comparison of different message headers. It further presents the results obtained through data analyses and finally the hypotheses testing and results.

### 6.1 General data and statistics from dataset

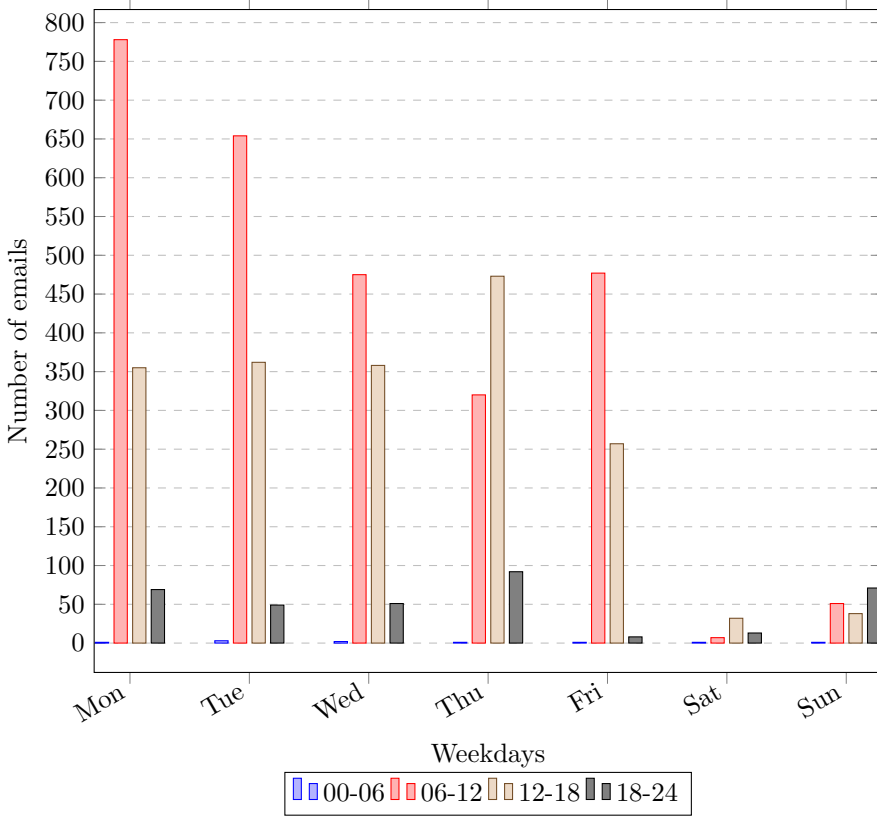
The dataset consists of data from 5000 reported emails, their email header and message header fields. Data was collected over 15 weeks from 26<sup>th</sup> January to the 10<sup>th</sup> May. An average of 48 emails were reported each day. The amount of reported emails ranges from 1 reported email per day, to 290 reported emails in one day. There were a total of 309 unique email and message header fields all together. To narrow down and choose specific headers, the data analyses and hypotheses testing looked only at anti-spam, and email authentication message headers found in the *inner-email*. It included also the message headers from, to, subject and date found both in the *outer-email* and *inner-email*. Results and data showing individual email addresses are not displayed in this chapter.

#### 6.1.1 Outer-email, which is generated when users report suspicious emails

General data from the *outer-email*, based on the email address and name found in the **From** message header field.

- 570 users have reported emails (total 5000 emails).
- 19 users have reported more than 50 emails each (total 2225 emails).
- 397 users have reported less than 5 emails each (total 720 emails).
- 44.5 % of the reported emails come from 3.3 % of the contributing users, reporting more than 50 emails each.

The following table and figure shows which day and at what time the suspicious emails were reported by users. Results are from looking at the **Date** message header field in the *outer-email*, which gives information on when the suspicious email was reported. Most of the emails, 90.2 %, are reported from Monday to Friday between 06:00 and 18:00.



**Figure 6.1:** Weekday and time when the emails were reported.

**Table 6.1:** Complementary data to figure 6.1.

Time / Weekday	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Total:
00-06	1	3	2	1	1	1	1	10
06-12	778	654	475	320	477	7	51	2762
12-18	355	362	358	473	257	32	38	1875
18-24	69	49	126	115	93	70	89	353
Total:	1203	1068	942	729	761	328	386	5000

Correlation between the **From** message header field found in the *outer-email* and the **X-MS-Exchange-Organization-SCL** anti-spam message header field value found in the *inner-email*.

Emails with SCL value 4 or lower, including emails without SCL header and value. These would be delivered to end-users' inboxes.

- 501 users have reported emails (total 2534 emails).
- 8 users have reported more than 50 emails each (total 603 emails).
- 398 users have reported less than 5 emails each (total 685 emails).
- 23.8 % of the reported emails come from 1.6 % of the contributing users.

Emails with SCL value 5 or higher. These would be delivered to end-users' spam folders.

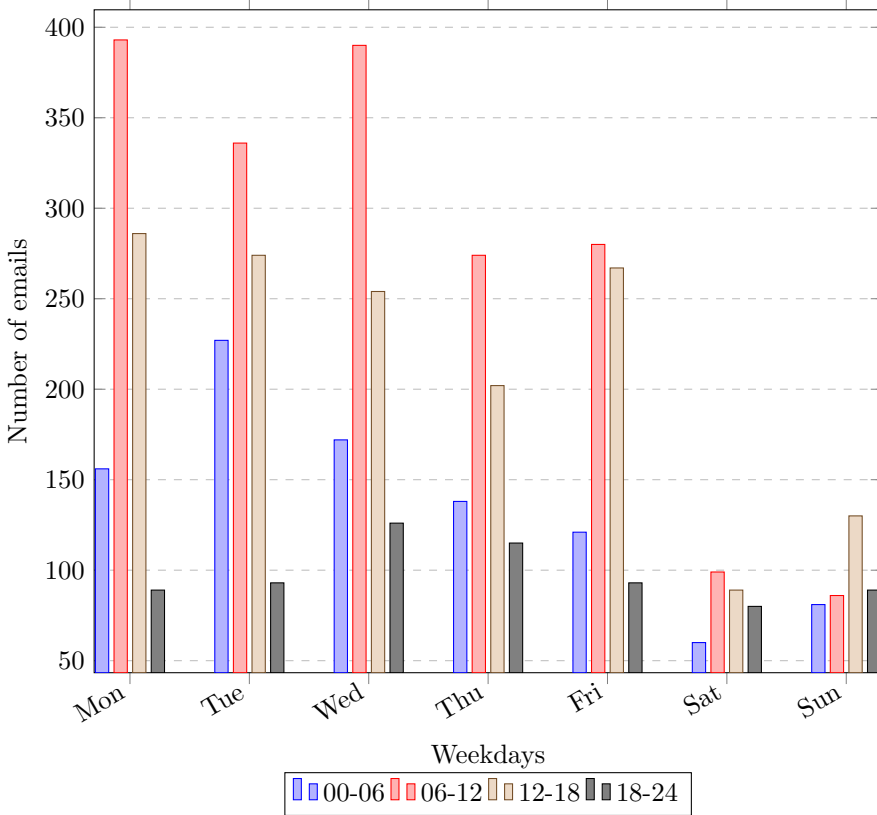
- 197 users have reported emails (total 2466 emails).
- 13 users have reported more than 50 emails each (total 1413 emails).
- 124 users have reported less than 5 emails each (total 233 emails).
- 57.3 % of the reported emails come from 6.6 % of the contributing users.

### 6.1.2 Inner-email, the suspicious emails being reported by users

General data from the *inner-email*, based on the domain name, email address and name found in the **From** message header field.

- 1642 different sender domains (total 5000 emails).
- 11 domains have sent more than 50 emails each (total 1260 emails).
- 1475 domains have sent less than 5 emails each (total 1919 emails).
- 25.2 % of the received emails come from 0.7 % of the sender domains, sending more than 50 emails.
- 2277 different sender email addresses (5000 emails).
- 4 sender email addresses have sent more than 50 emails each (total 330 emails).
- 2048 sender email addresses have sent less than 5 emails each (total 2655 emails).

The following table and figure shows which day and at what time the suspicious emails were received by users. Results are from looking at the **Date** message header field in the *inner-email*, which gives information on when the suspicious email was received. Most of the emails, 59.1 %, are received from Monday to Friday between 06:00 and 18:00.



**Figure 6.2:** Weekday and time when the emails were received.

**Table 6.2:** Complementary data to figure 6.2.

Time / Weekday	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Total:
00-06	156	227	172	138	121	60	81	955
06-12	393	336	390	274	280	99	86	1858
12-18	286	274	254	202	267	89	130	1502
18-24	89	93	126	115	93	70	89	685
Total:	924	930	942	729	761	328	386	5000

Correlation between both the **From** message header field and the **X-MS-Exchange-Organization-SCL** anti-spam message header field value found in the *inner-email*.

Emails with SCL value 4 or lower, including emails without SCL header and value. These would be delivered to end-users' inboxes.

- 674 different sender domains (total 2534 emails).
- 7 domains have sent more than 50 emails (total 719 emails).
- 583 domains have sent less than 5 emails (total 808 emails).
- 28.4 % of the received emails come from 1.0 % of the sender domains, sending more than 50 emails.
- 1080 different sender email addresses (2534 emails).
- 4 sender email addresses have sent more than 50 emails (total 299 emails).
- 979 sender email addresses have sent less than 5 emails (total 1281 emails).
- 11.8 % of the sent emails come from 0.4 % of the sender addresses, sending more than 50 emails.

Emails with SCL value 5 or higher. These would be delivered to end-users' spam folders.

- 1041 different sender domains (total 2466 emails).
- 4 domains have sent more than 50 emails (total 305 emails).
- 956 domains have sent less than 5 emails (total 1195 emails).
- 12.4 % of the received emails come from 0.4 % of the sender domains, sending more than 50 emails.
- 1258 different sender email addresses (2466 emails).
- 0 sender email addresses have sent more than 50 emails (total 0 emails).
- 1168 sender email addresses have sent less than 5 emails (total 1453 emails).

Correlation between both the **From** message header field and the **X-MS-Exchange-Organization-SCL** anti-spam message header field value found in the *inner-email*. The data found in the **From** message header field have been manipulated to only show the domain name. This data has been added to a list, counted and sorted in descending order.

Top reported sending domains where SCL value is 4 or lower, and emails without header or SCL value. Table 6.3 showing domains that have been reported sending more than 20 emails.

**Table 6.3:** Top reported sending domains where SCL is -1,0,1,2,3,4 or No Value.

Domain name:	Number of emails:
163.com	192
ntnu.no	132
126.com	124
gmail.com	95
tarim.gov.tr	62
acieu.co.uk	58
nam-mail.com	56
vip.163.com	49
enea.it	34
hotmail.com	30
lgm.gov.my	29
rfidhy.com	28
hstek-cn.com indepthnrg.com	27
vip.sina.com faktura-program.net	25
sintef.no	24
tuisong.wiremesh.me	23
phenixbelt.com daum.net	22
krausens.lv	21



Top reported sending domains where SCL value is 5 or higher. Table 6.4 showing domains that have been reported sending more than 20 emails.

**Table 6.4:** Top reported sending domains where SCL value is 5, 6, 7, 8 or 9.

Domain name:	Number of emails:
gmail.com	133
yahoo.com	62
163.com	59
acieu.co.uk	51
outlook.fr	47
outlook.com	41
indepthnrg.com	38
yandex.com	37
PayPal.cc	33
kuzeymarine.com	30
gadmarine.com	27
126.com	26
it.evergreen-line.com	25
alpmarine.com	24
vip.163.com	
moononline.info	

## 6.2 Comparison of different message header fields

This section presents results from comparison of message header fields found in the *inner-email*. These message header fields are anti-spam headers and headers providing results from the email authentication protocols. Individual results for each message header can be found in the appendices.

- **X-MS-Exchange-Organization-SCL** in appendix A.
- **Received-SPF** in appendix B.
- **DKIM-signature** in appendix C.
- **Authentication-Results** SPF, DKIM and DMARC in appendix D.
- **X-Forefront-Antispam-Report** SCL, and SPF in appendix E.
- Complete tables for the comparison of the message headers can be found in appendix F.

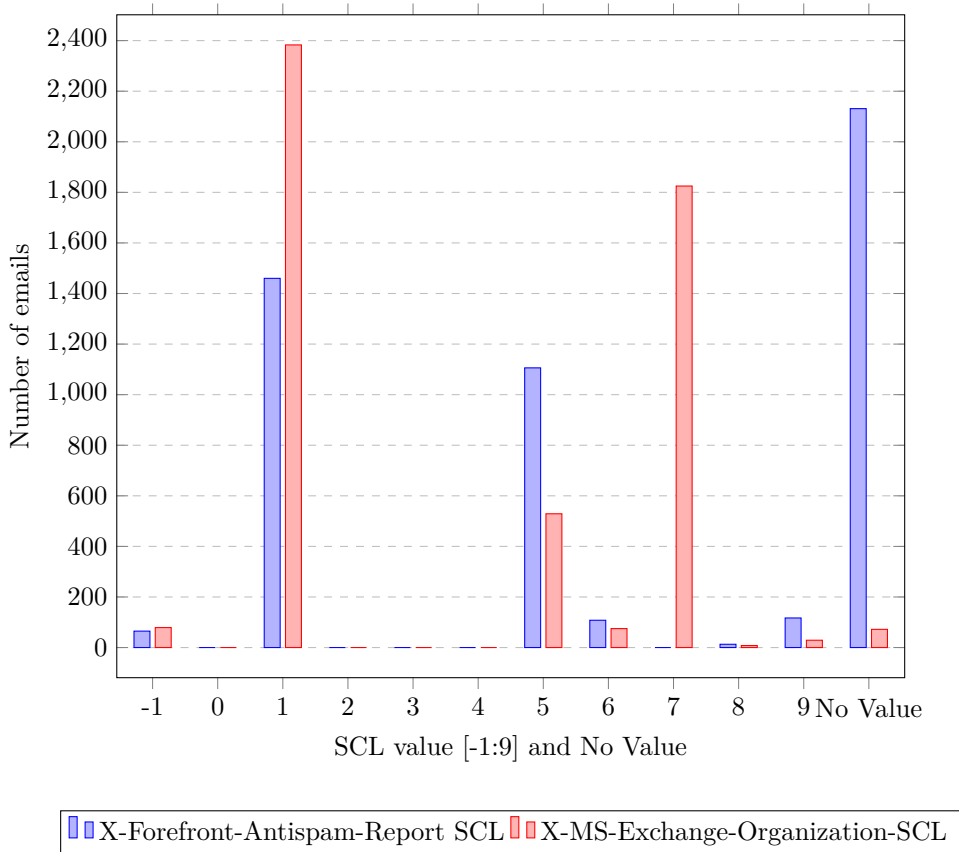
**X-Forefront-Antispam-Report SCL and X-MS-Exchange-Organization-SCL**

The following figure and table shows the results from looking at X-Forefront-Antispam-Report and X-MS-Exchange-Organization-SCL message headers found in the emails. X-Forefront-Antispam-Report SCL shows the SCL value provided by the Microsoft Exchange Online Protection email filtering service. While the X-MS-Exchange-Organization-SCL is another anti-spam stamp from a content filter agent using Microsoft SmartScreen technology. The first message header is found in 2869 of the 5000 reported emails. The second message header is found in 4928 of the reported emails.

As seen in the complementary table to figure 6.3 there are no major differences. X-Forefront-Antispam-Report SCL has a tendency to rank emails with SCL value 5, while X-MS-Exchange-Organization-SCL rank more emails with SCL value 7. Both could be used to filter emails by a spam filter if a threshold of SCL value 5 was used. What is alarming is that X-Forefront-Antispam-Report SCL has ranked an email with SCL -1, saying that the sender email address or domain is white-listed. X-MS-Exchange-Organization-SCL have ranked the same email with SCL 7. This could be considered closer by IT security personnel. The X-MS-Exchange-Organization-SCL message header is found in most of the emails and it has a tendency to rank emails with a higher SCL value. It is found to be the best of these two message headers for filtering spam emails.

**Table 6.5:** Complementary table to figure 6.3. X-Forefront-Antispam-Report SCL is shown horizontally, X-MS-Exchange-Organization-SCL is shown vertically in the table.

SCL	-1	1	5	6	7	8	9	No Value	Total
-1	64	0	0	0	1	0	0	0	65
1	0	1456	0	0	0	0	0	4	1460
5	0	0	285	0	820	0	0	1	1106
6	0	0	0	43	65	0	0	0	108
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	8	5	0	0	13
9	0	0	0	0	103	0	14	0	117
No Value	15	927	244	32	828	3	15	67	2131
Total	79	2383	529	75	1825	8	29	72	5000

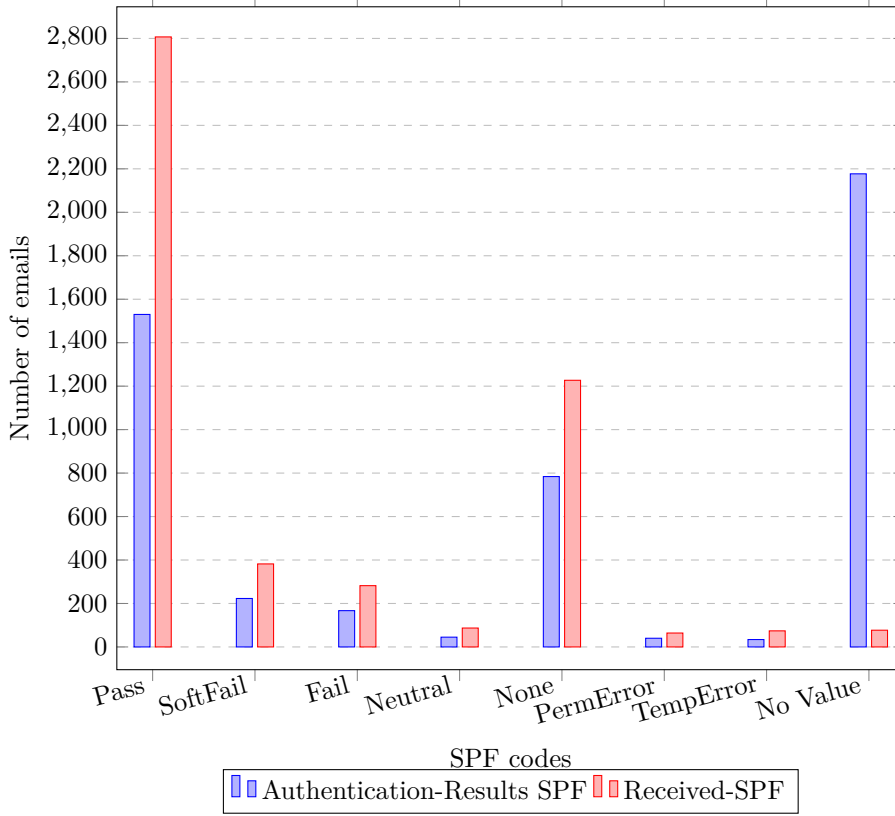


**Figure 6.3:** X-Forefront-Antispam-Report SCL and X-MS-Exchange-Organization-SCL.

### Authentication-Results SPF and Received-SPF

The following figure and table shows the results from looking at Authentication-Results and Received-SPF message headers found in the emails. Authentication-Results provided the results of checks against SPF, DKIM and DMARC used by Microsoft Office 365 email authentication. The Received-SPF is another email authentication message header. The first message header is found in 2823 of the 5000 reported emails. The second message header is found in 4923 of the reported emails.

As seen in the complementary table to figure 6.4 there are no major difference. There is one email that have failed the SPF check in Authentication-results, while it has passed the Received-SPF check. This should be looked into. The Received-SPF message header is found in most of the emails. It is found to be the best of these two message headers for filtering based on SPF email authentication results.



**Figure 6.4:** Authentication-Results SPF and Received-SPF.

**Table 6.6:** Complementary table to figure 6.4.

SPF	Pass	Fail	Soft Fail	Neutral	None	Perm-Error	Temp-Error	No Value	Total
Pass	1530	0	0	0	0	0	0	0	1530
Fail	1	166	0	0	0	0	0	0	167
SoftFail	0	0	223	0	0	0	0	0	223
Neutral	0	0	0	45	0	0	0	0	45
None	0	0	0	0	784	0	0	0	784
PermError	0	0	0	0	0	40	0	0	40
TempError	0	0	0	0	0	0	34	0	34
No Value	1276	116	159	42	443	24	40	77	2177
Total	2807	282	382	87	1227	64	74	77	5000

## 6.3 Hypotheses testing

This section presents the hypotheses and results from testing conducted on the dataset. Each subsection is divided into a specific hypothesis with corresponding results from the testing.

### 6.3.1 Hypothesis 1, Strict enforcement of SCL

**Hypotheses 1:** Most of the reported emails could have been avoided by a strict enforcement of the SCL anti-spam message header value.

**Results:** Emails with a SCL header value of 4 or lower, including the emails without a SCL header value counts for 2534 (50.68 %) of the total 5000 reported emails. Emails with a SCL header value of 5 or higher counts for 2466 (49.32 %) of the total 5000 reported emails. This means that either the anti-spam filtering has not worked as supposed or implemented, or that almost half of the reported emails come from end-users' spam folders. The latter would mean that users have actively used the reporting functionality as a spam button on emails already marked as spam by the technical solutions.

### 6.3.2 Hypothesis 2, Strict enforcement of email authentication

**Hypothesis 2:** Most of the reported emails could have been avoided by a strict enforcement of the results from the email authentication protocols found in different message header fields.

#### 2.1, Strict enforcement of SPF

Filtering or blocking emails where the SPF has a different result of the email authentication than Pass. Received-SPF (found in 4923 emails), Authentication-Results (SPF found in 2823 emails) and X-Forefront-Antispam-Report (SPF found in 2865 emails) are different message header fields with SPF results.

**Results:** Of the total reported emails, 2884 (57.68 %) emails have Received-SPF status Pass, or no Received-SPF message header. This means that 2116 (42.32 %) emails have different statuses than Pass and could have been blocked or filtered as spam. If comparing SCL and SPF, only looking at emails with SCL value of 4 or lower and SPF Pass including emails without SCL and SPF message header fields, 1908 (38.16 %) emails would have been accepted. This means that 3092 (61.84 %) of the total emails could have been blocked or filtered to spam folder if SCL and Received-SPF had been used together. Authentication-Results SPF and X-Forefront-Antispam-Report SPF status are found in too few emails to be considered. However, if they all appear in the same email, they should provide the same results.

## 2.2, Strict enforcement of DKIM

Filtering or blocking emails where DKIM has another status than Pass. DKIM-signature (found in 1426 emails) message header field does not provide any results on the authentication. This is found in the Authentication-Results message header.

**Results:** Only 2888 (57.76 %) of the total reported emails are found with Authentication-Results and DKIM. Emails with DKIM Pass counts for 952 (19.04 %) of the total emails. Blocking or filtering based on DKIM status could have been used on 1936 (38.72 %) of the emails. Authentication-Results DKIM status is found in too few emails to further be considered with SCL.

## 2.3, Strict enforcement of DMARC

Filtering or blocking emails where DMARC has another status than Pass. Status about DMARC can be found in Authentication-Results (DMARC found in 2862 emails).

**Results:** Only 2862 (57.24 %) of the total reported emails are found with Authentication-Results and DMARC. Emails with DMARC Pass counts for 628 (12.56 %) of the total emails. Blocking or filtering based on DMARC status could have been used on 2234 (44.68 %) of the emails. Authentication-Results DMARC status is found in too few emails to further be considered with SCL.

### 6.3.3 Hypothesis 3, Notify IT security personnel based on reported emails

**Hypothesis 3:** A set of rules based on data from the reported emails can be used to notify IT security personnel.

#### 3.1, notify about *bulk-emails* reported

Identify email addresses used to send *bulk-emails*. The *bulk-emails* term is in this context used loosely, because the amount of emails are quite low. The test is based on data from the reported emails on sender email address, subject and date.

**Results:** There are 106 different emails reported (total 590 emails) that have been received 3 or more times with same sender email address, subject and date. This only amount to 11.8 % of the total reported emails. The username part of the email sender addresses is removed to anonymize. It only shows the domain name. The different listings with the same domain name also have the same username, so they are sent from the same user email account. These are linked to emails sent from hacked email accounts [75].

**Table 6.7:** Top reported *bulk-emails*, more than 10 reported.

Domain name:	Number of emails:
anonymous@ntnu.no	37
anonymous@ntnu.no	25
anonymous@tuisong.wiremesh.me	23
anonymous@tarim.gov.tr	22
anonymous@ntnu.no	21
anonymous@ntnu.no	20
anonymous@acieu.co.uk	18
anonymous@acieu.co.uk anonymous@tarim.gov.tr	16
anonymous@ntnu.no	13
anonymous@tarim.gov.tr	12
anonymous@ntnu.no	11

### 3.2, notify about emails reported that are white-listed

Notify about reported emails where the sender is white-listed. The test is based on data from the reported emails on sender email address, and with SCL value -1.

**Results:** Only 79 (1.58 %) of the reported emails have SCL value -1 and will be regarded as white-listed passing the spam filter. This is not significant, but it could be of useful information to IT security personnel to know of email addresses or domains that wrongfully have been white-listed. Either from users marking emails mistakenly as white-listed in their email clients, or from emails that have fooled the spam filter in any way.

**Table 6.8:** Showing domain names, from emails reported with SCL value -1, as white-listed.

Domain name:	Number of emails:
nam-mail.com	54
sintef.no	18
gmail.com	3
+4400441625810710 +4400441625810710 usa.com rambler.ru kwadratuur.be	1

### 3.3, notify about internal emails reported

Notify about reported emails where the sender is internal. The test is based on data from the reported emails on sender email address, missing SCL and Received-SPF message header fields which indicate that the email is sent internal within the organization.

**Results:** Only 61 (1.22 %) of the reported emails have no SCL or Received-SPF message header, and should be regarded as internal emails. However, only 6 of these 61 emails have email sender address from the internal domain. This information could be useful information showing that some emails lose their SCL and Received-SPF message headers after passing through the email system. This could potentially lead to emails with initially a high SCL value being delivered to end-users' inboxes.

**Table 6.9:** Showing domain names, from emails reported with no SCL or Received-SPF message header.

Domain name:	Number of emails:
riwuled.com	11
163.com sintef.no	5
wvschools.ca gmail.com	3
idrettsforbundet.no hotmail.com weltranscn.com	2

#### 6.3.4 Hypothesis 4, Block or deliver emails to spam folder based on reported emails

**Hypothesis 4:** Could emails be used proactively for blocking or filtering future emails to spam folder after some of them have been reported.

##### 4.1, block emails based on email address from reported emails (for all SCL values)

The thought behind this is to blacklist reported emails for a period (for an example 7 days) if two or more emails are reported with the same email sender address.

There are 326 different email addresses that have been reported to send 3 or more emails. These senders have sent a 2798 (55.78 %) of the total reported emails. The remaining 2202 (44.04 %) emails have been sent from 1951 email addresses, but all of them sending less than 3 emails.



**Results:**

- If two emails are reported within 10 minutes before the third email is received, 1539 (30.78 %) of the 5000 reported emails could have been blocked or automatic delivered to spam folder. If only looking at senders that have sent 3 or more emails (2798 emails), the number is 55.00 %.
- If two emails are reported within 60 minutes before the third email is received, 1289 (25.78 %) of the 5000 reported emails could have been blocked or automatic delivered to spam folder. If only looking at senders that have sent 3 or more emails (2798 emails), the number is 46.07 %.
- If two emails are reported within 24 hours before the third email is received, 952 (19.04 %) of the 5000 reported emails could have been blocked or automatic delivered to spam folder. If only looking at senders that have sent 3 or more emails (2798 emails), the number is 34.02 %.

**Table 6.10:** Complementary results to test 4.1, only showing a selection of senders with more than 25 emails. Usernames, part of the email address, identifying people by name are anonymized.

Sender email address:	Emails	10 min	60 min	24 hrs
anonymous@ntnu.no	127	31	0	0
anonymous@acieu.co.uk	79	76	76	76
anonymous@tarim.gov.tr	62	0	0	0
info@nam-mail.com	62	60	60	60
anonymous@outlook.fr	47	45	45	45
sj@indepthnrg.com	46	27	27	27
wfen_452@163.com	45	42	42	42
reci111@yahoo.com	39	26	0	0
cheetah_team.project_list-subscribe@enea.it	37	35	35	35
Services@PayPal.cc	33	31	31	31
anonymous@kuzeymarine.com	30	26	19	0
devaraj@lgm.gov.my	29	0	0	0
gad@gadmarine.com	27	23	21	0
info@rfdhy.com	26	24	24	24
post@faktura-program.net	26	24	24	24
anonymous@it.evergreen-line.com	25	21	17	0
Total:	2798	1539	1289	952

**4.2, block emails based on email address from reported emails (for all SCL with value 3 or lower and No Value)**

Initially the same as in test 4.1, but this is a more specific test only looking at emails where SCL is -1,0,1,2,3 or No Value. These are the emails that would have been delivered to end-users' inboxes. The amount of emails which meet these criteria is 2534 (50.68 %) out of the total 5000 reported emails.

There are 172 different email addresses that have been reported to send 3 or more emails. These senders have sent a 1496 (29.92 %) of the total reported emails. The remaining 1038 (20.76 %) emails have been sent from 907 email addresses, but all of them sending less than 3 emails.

**Results:**

- If two emails are reported within 10 minutes before the third email is received, 686 (27.07 %) of the 2534 reported emails with the SCL criteria could have been blocked or automatic delivered to spam folder. If only looking at senders that have sent 3 or more emails (1496 emails), the number is 45.86 %.
- If two emails are reported within 60 minutes before the third email is received, 594 (23.44 %) of the 2534 reported emails with the SCL criteria could have been blocked or automatic delivered to spam folder. If only looking at senders that have sent 3 or more emails (1496 emails), the number is 39.71 %.
- If two emails are reported within 24 hours before the third email is received, 525 (20.72 %) of the 2534 reported emails with the SCL criteria could have been blocked or automatic delivered to spam folder. If only looking at senders that have sent 3 or more emails (1496 emails), the number is 35.09 %.

**Table 6.11:** Complementary results to test 4.2, only showing a selection of senders with more than 15 emails. Usernames, part of the email address, identifying people by name are anonymized.

Sender email address:	Emails	10 min	60 min	24 hrs
anonymous@ntnu.no	127	31	0	0
anonymous@tarim.gov.tr	62	0	0	0
info@nam-mail.com	56	54	54	53
anonymous@acieu.co.uk	54	52	52	52
cheetah_team.project_list-subscribe@enea.it	34	32	32	32
anonymous@lgm.gov.my	29	0	0	0
sj@indepthnrg.com	27	11	11	11
wfen_452@163.com	27	23	23	23
info@rfdhy.com	26	24	24	24
post@faktura-program.net	25	23	23	23
anonymous@tuisong.wiremesh.me	23	0	0	0
sale@phenixbelt.com	22	17	17	17
anonymous@krausens.lv	21	19	19	19
noreply@ndc.easyfairs.com	20	16	16	16
kontakt@ryka.no	19	2	2	2
info@levering-go.com	18	16	16	16
anonymous@hotmail.com	17	13	10	5
anonymous@stortinget.no	17	0	0	0
news@ds.scandicblog.com	16	14	13	0
Total:	1496	686	594	525



# Chapter 7

## Discussion and suggested solutions

This chapter is a discussion based on the background in chapter 2, the results from chapter 5 and 6 within the scope of the research questions presented in 1.3. It starts with a more general discussion addressing the first research question on how email is used, threats and possible ways of mitigating these threats within the organization. Including possible sources of error with the datasets. The chapter is further divided into two sections discussing the second and third research questions with suggested solutions.

Email has been and still is the most widely used means of internal and external communication around the world. Earlier this technology did not have good enough standards and implementations addressing problems with email security. As the technology evolved and the use of email was adapted to new sets of functionality, e.g. extensions to support sending of data, other content than plain text, new vulnerabilities have appeared. With vulnerabilities, comes potential threats and different threat agents trying to exploit them. Often during information technology and software development, security measures have been *a nice to have feature* added at the end of projects, or added when something in the existing framework needed to be altered. Whether this is still the case today is another discussion. However, it is evident that the large community involved in developing and evolving email have had challenges when it comes to how to make it more secure. Development and implementation of technical solutions the past 15 years, e.g. several email authentication protocols, spam filters and anti-malware protection software, are attempts to improve the vulnerabilities and to mitigate threats.

The importance of having these technical solutions implemented cannot be emphasized enough. They are vital tools, internally or externally implemented, within the organizations email system for stopping the amount of unsolicited and potential malicious emails in circulation. Without them, end-users' inboxes would most likely be flooded with emails. This would be to disadvantage of using email as a way of communicating, a lack of trust towards using email and be of annoyance to users.

Even though the technical solutions have improved email security, they do not come with a magical fix or a *silver bullet* stopping all malicious emails. Attacks through using email happen all the time, most recently with the wave of ransomware distributed via email across Europe. Some email will pass the technical security measures, and when they do it is up to the human user's choice to either discard or open the email and its contents. This choice is shown to be closely linked with the user's prior experiences, level of risk perception and security awareness. Users react to different things than the technical security measures do and can. Some users, whether intentionally or not, fail to see the potential warning signs if any. Users are fooled time after time into submitting personal information, clicking on questionable URL links to malicious sites or opening attachments. Phishing emails containing falsified URL links, claiming to be something else, and attachments have been the most common in the dataset of reported emails. They are effective, and it is expected that this method in some sort of way will continue in years to come. This can also be seen in the dataset along with a huge amount of spam emails.

How to address the challenges concerning email security. Technical solutions evolve and solutions are implemented. This is also true for threat agents, their methods and types of attack. Many emails could have been filtered or blocked if the email authentication protocols had been implemented and fully working for all the different email use cases. Technical security measures must be weighed against the wanted functionality. Either stopping most of the illegitimate emails and at the same time some legitimate emails, or opening for all legitimate emails and at the same time some illegitimate emails. This is a double-edged sword, and it is hard to facilitate for full functionality and full security at the same time. Some technical measures are not implemented by choice because of this. This is related to the hypothesis in 6.3.2 for strict enforcement of the email authentication protocols DKIM and DMARC. Out of the emails found having DMARC authentication results, 2234 (44.68 %) of the reported 5000 emails could have been blocked or filtered.

The de facto reality is that no matter what technical measures are put in place some malicious emails pass and reach end-users' inboxes. Ultimately security comes down to people and their day to day practices. One countermeasure to face this reality has come to attention via the case study conducted. This countermeasure is a security culture initiative in an organization working to protect users which aim to reduce the success rate of attacks. The initiative consists of two main methods, one method which is informative on the threats associated with email and a method applying functionality in users' email clients for reporting suspicious emails. The first method is useful in boosting peoples' security awareness and scepticism. The latter is a smart way of using email users as a *network of sensors*. This could alert IT security personnel, and if the reported emails are found to be dangerous, preventive information could be distributed within the organization.

Challenges with this security measure is firstly how users choose to use the functionality for reporting suspicious email. Secondly how could data from the reported emails be used efficiently as a preventive measure against email threats. The first challenge address end-users' ability to report emails that are suspicious and potentially dangerous. This is closely connected with the veracity of the dataset with reported emails used for data analyses and hypotheses testing. The second challenge address what type of data from the emails are to be used in an automated system for alerting users and system administrators. These will be further discussed in the next two sections.

There are sources of error with these results.

## 7.1 How is the functionality for reporting email used

This section discusses the research question on how the functionality for reporting emails is used and some suggested solutions for possible improvements.

The purpose, or intention, of the email reporting functionality is for users to report email they find suspicious or possibly dangerous, which could be used for alerting other users. These would be emails that have passed the technical security measures and ended up in end-users' inboxes. To better understand what characteristics of the emails that make users choose to use the functionality for reporting emails, a questionnaire was conducted. The questionnaire is presented in chapter 5. A hypothesis test and data analyses, presented in 6.3.1 and in 6.1, was further conducted to be able to verify the veracity of the dataset from the questionnaire. The latter was important because there were only 83 responses from the questionnaire. This amounts to only 17.66 % of the 570 users who were found to have reported emails in the dataset of reported emails.

The emails provided for the questionnaire was not particularly well designed by the sender, meaning that the emails were quite general in text and context, exemplifying typical spam and phishing emails. The pictures in the questionnaire on each email did not provide all the message headers found in the email. But the results from the questionnaire show that most users would report emails based on its message text and its context, unknown or bogus email sender address in the **From** message header and the subject text found in the **Subject** message header. Other characteristics which raise suspicion are attachments and falsified URLs found in the emails. Feedback from the questionnaire can further be used to divide users reporting suspicious emails into two groups. The first group is users who report based on the previous mentioned criteria found in the emails. The second group of users delete emails they find to be obvious only by looking at a few of the first message headers. They only report those emails found to be potentially dangerous. How users choose to report emails,

their thought-process and reasoning, is a major task on its own. To have more measurable data on how users use the functionality for reporting emails the attention shifted towards the dataset of reported emails. Results from looking at the **From:** message header in the *outer-email* and the **X-MS-Exchange-Organization-SCL:** anti-spam message header in the *inner-email* are interesting.

Of the nearly 2000 users within the organization, 570 users are found to have reported the 5000 emails in the dataset. There is a significant difference in the frequency of users reporting emails. Of the 570 users, 397 users have reported less than 5 emails each while 19 users have reported more than 50 emails each. By looking closer at the SCL anti-spam message header, the results can provide information whether emails are reported from end-users' inboxes or spam folders. Emails without this message header, or a SCL value of 4 or lower, would be delivered to end-users' inboxes. Emails with a SCL value of 5 or higher would be delivered to end-users' spam folders. The results in 6.1 show that the amount of emails reported from end-users' inboxes are similar to that of the emails reported from spam folders. There are 197 users found to have reported from their spam folders, with 13 users reporting more than 50 emails and 124 users reporting less than 5 emails. These results need to be considered with the fact that spam filters could have been modified or they could have been none-functional for some time during the data collection when users have reported emails. These sources of error would have caused emails with high values in their anti-spam message header to be wrongly delivered to end-users' inboxes. By looking closer at some of the users that supposedly have reported emails from spam folders, the dataset shows a numerous times that 10, 20, and 30 emails have been reported within a short period of time. These observations suggest that users report emails that have accumulated in their spam folders over some time. From the results, it seems that the reporting functionality for some users is similar to that of a deleting functionality. This would not agree with the intention of the reporting functionality. Because emails with a high SCL have been intercepted by some technical security measures and delivered to end-users' spam folders. The emails of interest are emails delivered to end-users' inboxes.

### **7.1.1 Information about the functionality, and expanding the functionality for reporting emails**

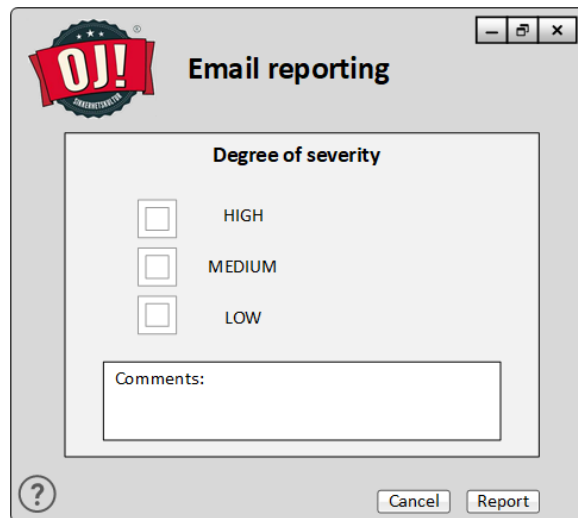
Improvements are found to be necessary to better achieve the intention of the reporting functionality. This subsection presents two suggested solutions.

A question was raised based on the findings in the dataset of reported emails. It is related to whether users should have some sort of guidelines or instructions on what kind of emails to report or not. However, this could be difficult based on the varying and changing nature of malicious emails. It would also be difficult to have a



set of rules because users are different in the aspects of what they find suspicious, their security awareness and risk perception. One suggested solution which could be sufficient is to provide users with information which clearly states the intention and how the reporting functionality works. This information should also state that emails found in spam folders are not to be reported. This would facilitate for a mutual understanding and it could give a data collection of reported emails being malicious. This would reduce the number of false positives reported emails IT security should process.

The second solutions are directly related to the email reporting functionality. The though behind this solution is to make users report all unsolicited emails found in their inboxes, and at the same time state the degree of severity the user finds the email. This would meet the use of the two distinct groups of users, users who report *everything* and those who only report emails they find dangerous. Hopefully, this would compel users to look closer at emails and make a thorough assessment. This solution could be implemented in the existing functionality. At the time users click report, the chosen option and commentary would be added as values to a new optional X- message header on the *outer-email* by the email agent. This message header would be parsed as any other header, and would work with the environment used in this thesis. The second, but not the recommended implementation, is to add the data to the *outer-email* as an attachment. The solution could look like the examples shown in figures 7.1 and 7.2.



**Figure 7.1:** Adding functionality to email reporting, example 1.

The first example shows three different options for the degree of severity, from High

to Low, with an added *comments* field. The three options are to be used with an automated system, which could use this information to alert IT security personnel. The commentary-field is thought to be used in combination with the web application for visualizing data to users and system administrators.

**Figure 7.2:** Adding functionality to email reporting, example 2.

The second example shows two different options for the degree of severity, from Dangerous to Spam, with an added *comments* field. These are to be used in the same way as explained for figure 7.1.

## 7.2 How could the reported emails be used more efficiently

This section discusses the research question on how data from the reported emails could be further used in the continuous work with email security. It also discusses some suggested solutions on how to use this data to the benefit of users.

There is a huge amount of data being reported through this email reporting functionality. It is not reasonable for IT security personnel to manually process all these emails. Especially when email security is just one of the responsibilities within information security. Some sort of automation of the reported emails needs to be implemented to handle parts of this formidable task. Automation would not take over all manual processing, but it can help in some aspects of using data from the reported emails more efficient.

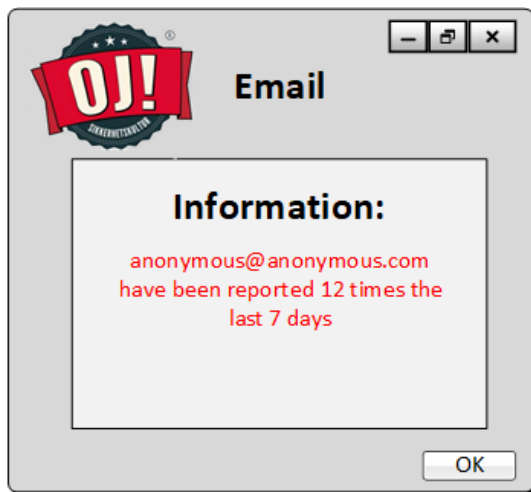
### 7.2.1 Temporarily block or filter emails for some time

Data analyses and results in 6.1 show that 60 % of emails are received users from Monday to Friday between 06.00 and 18.00. While users report 90 % of the emails within the same weekday and time-period. This led to the hypothesis testing in 6.3.4. It shows with satisfactory results that the reported emails can be used proactive for blocking or filtering future emails after two emails with the same sender email address had been reported. If looking at the complete dataset 30.78 % of the reported emails would have been blocked or filtered. While, if only looking at the dataset where email sender addresses have been reported 3 or more times, 55.0 % of the reported emails would have been blocked or filtered. These are high numbers, but it is not that easy. If this was to be implemented several questions would have to be answered. During testing all emails and email sender addresses were assumed to be illegitimate because they were reported. How is white-listing going to be implemented, and which domains or email addresses are going to be on this list? What if a legitimate email being reported is coming from within the organization? The solution would stop a lot of the emails which is shown in the testing, but it would potentially stop legitimate email as well. This solution could be implemented with some changes to existing code. The script would need an automated module for setting rules in the organizations spam filters based on the database with parsed emails and a white-list over safe email sender addresses and domains. The latter could also be implemented as some sort of *inter-organizational* spam rating of domains. In order to reduce the total amount of unsolicited emails. Based on the reported emails as shown in table 6.3 some domains appear more often than others and this is not beneficial for the domains' reputation.

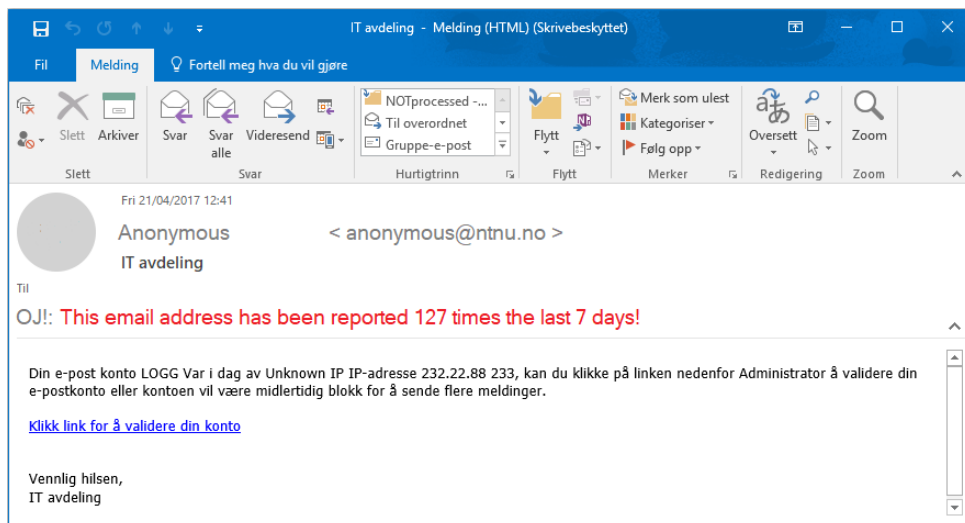
### 7.2.2 Functionality to alert users when receiving emails

The last suggested solution is a functionality for alerting users when they receive emails. This is a functionality thought implemented in users email clients or as a small pop-up window on the desktop. It will alert users based on data from the database of reported emails, e.g. on the sender email address and how many times it has been reported within the last 7 days. This is an example of how the reported emails can actively be used in the email security work. The colour scheme thought used is initially yellow for the email sender address reported once and increasingly more towards red and dark red as the number the email sender address has been reported increases. This is not the vital part. But it is important that the functionality does not show an email as safe, e.g. in green text saying that the email sender address has not been reported. The relevance of this functionality is closely linked with section 7.1.1, because it would only work if users report emails. The solutions could look like the examples shown in figure 7.3 and 7.4. Both could be implemented using a script reading the **From:** message header and checking the existing database if

the sender email address has previously been reported. The solution with a pop-up window could be viewed as very annoying, but it could be implemented for emails that have a high potential of damage. The second solution would be less annoying and a good method of alerting users for all emails they receive.



**Figure 7.3:** Example of alerting users about a reported email address with a pop-up window.



**Figure 7.4:** Example of alerting users in their email client.

# Chapter 8

## Conclusion and further work

The concluding chapter of this thesis presents the conclusion and suggestions for future work with the topic. The case study and the study of some of the existing literature has given context to the problem description and the first research question presented in this thesis. Following data collection through a questionnaire, data analyses and hypotheses testing on a set of reported emails have given measurable data and provided some results to the second and third research questions.

### 8.1 Conclusion

What is certain is that email is a major security concern. It is the most common way of attack, and it has shown to be highly effective. Email is based on older technology and a very direct means of communication. Technical security measures do not give a hundred percent solution to the challenges concerning email security. Malicious email ends up in end-users' inboxes. Users are fooled and tricked into revealing sensitive information or installing malicious code which can be used for destructive or criminal purposes which, in turn, can lead to unfortunate consequences. New email security vulnerabilities are exploited and attacks evolve. It is not very likely that email will ever be completely secure. System owners and administrators should do a thorough assessment on having an email system and using it as a tool for internal and external communication. This is based on the use today, the widespread use of malicious attachments and falsified URLs. There are alternatives to using email, both for communication within the organization and to external parties.

The only truly effective protection is to promote email security and make email users aware of the potential threats. One method to improve email security is the security initiative with functionality to report suspicious emails covered in this thesis. Along with it, it boosts peoples' security awareness and scepticism. To further improve email security, the functionality should add a possibility for giving feedback in some sort of a degree of severity. This would invite user to report all emails, and at the

same time alert specific email they see as potentially dangerous or malicious. This ranking could give IT security personnel a possibility to better choose which reported email that are important to investigate. Data from the reported emails can also be used in the email security work. To be sure reported emails would still need to go through some manual analysis. Data on email senders' addresses, and the number of times they have been reported can be used to alert users in their email clients or on their desktops. That said, the topic addressed is very large and it would need much more work to put the improvements into operation.

## 8.2 Further work

This section present suggestions for further work with the topic.

### 8.2.1 Virtual sandbox environment

This is somewhat out of scope, but still important. It is hard to classify or identify the reported emails as malicious, based on the work conducted in this thesis. Virtual sandboxing technology could have been used on the reported emails where URLs or attachments are found and test if they are malicious or not. The following presentation are thoughts for a continuation of the work on *Automating Email Attachments Scanning with Cuckoo*[76] done by Xavier Mertens in 2012 [77]. Building a Cuckoo sandbox [78] and setting up the CuckooMX [79] did not work in this instance, because the test domain email server did not support Postfix mail server.

A flowchart of this environment can be seen in figure 8.1. This example is good way to automate the processing of the reported emails. The entire flowchart is mostly based on the work presented in chapter 4 and figure 4.5, but the blue dotted box shows the added functionality. It works by processing reported emails. First it parses the *outer-email* and extracts the RFC822 attachment. *Inner-emails* without attachments will be processed, message headers are parsed, data will be sanitized and saved to the database. If the *inner-email* is found to have attachments it will be sent to the virtual sandbox environment. The attachments will be submitted to malware analysis tools to confirm if it is malicious or not. The email will be processed as a normal if it is found to be safe. If the email is found to be malicious or if the tests are inconclusive, it will be left for manual analysis by a security analyst. Still, even virtual sandbox environments have weaknesses and they will as with other security measures not be a hundred percent solution.

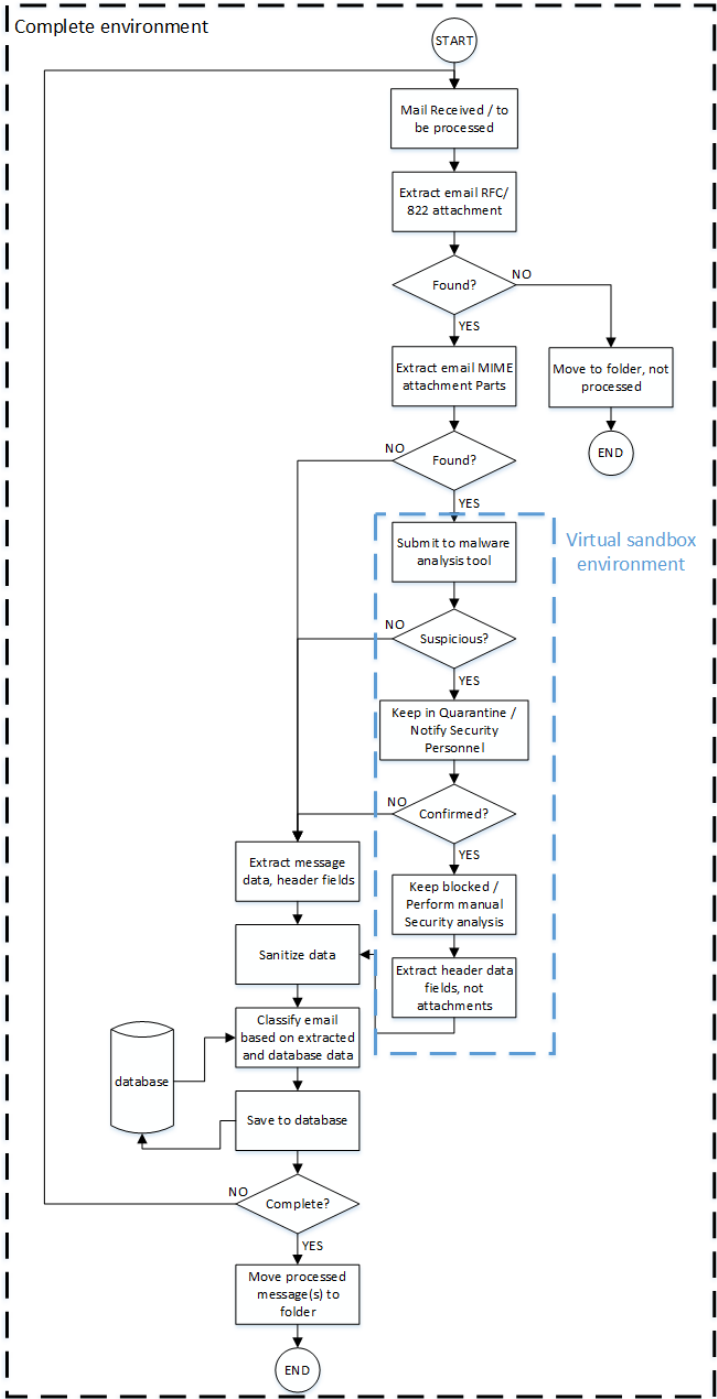


Figure 8.1: Flowchart of the environment with sandbox.





# References

- [1] Statista. Global spam volume as percentage of total e-mail traffic from january 2014 to dec 2016, by month, 2016. URL <https://www.statista.com/statistics/420391/spam-email-traffic-share/>. Last visited: 22-06-2017.
- [2] Internet Engineering Task Force. Rfc5322, internet message format, 2008. URL <https://datatracker.ietf.org/doc/rfc5322/>. Last visited: 22-06-2017.
- [3] United States Computer Emergency Readiness Team. Recognizing and avoiding email scams, 2008. URL [https://www.us-cert.gov/sites/default/files/publications/emailscams\\_0905.pdf](https://www.us-cert.gov/sites/default/files/publications/emailscams_0905.pdf). Last visited: 21-06-2017.
- [4] BBC. Massive ransomware infection hits computers in 99 countries, 2017. URL <http://www.bbc.com/news/technology-39901382>. Last visited: 21-06-2017.
- [5] Jurica Dujmovic. The 10 most digitally savvy countries in the world, 2016. URL <http://www.marketwatch.com/story/the-10-most-digitally-savvy-countries-in-the-world-2016-07-19?page=2>. Last visited: 02-07-2017.
- [6] INSEAD World Economic Forum and Cornell University. The global information technology report 2016, 2016. URL [http://www3.weforum.org/docs/GITR2016/WEF\\_GITR\\_Full\\_Report.pdf](http://www3.weforum.org/docs/GITR2016/WEF_GITR_Full_Report.pdf). Last visited: 21-06-2017.
- [7] Statistics Norway. Use of ict in the business sector, 2016. URL <https://www.ssb.no/statistikbanken/SelectVarVal/Define.asp?MainTable=IKTpfTeknologi1&KortNavnWeb=iktbruken&PLanguage=0&checked=true>. Last visited: 21-06-2017.
- [8] The RADICATI GROUP. Email statistics report 2015-2019, 2015. URL <http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf>. Last visited: 21-06-2017.
- [9] The RADICATI GROUP. Email statistics report 2016-2020, 2016. URL [http://www.radicati.com/wp/wp-content/uploads/2016/01/Email\\_Statistics\\_Report\\_2016-2020\\_Executive\\_Summary.pdf](http://www.radicati.com/wp/wp-content/uploads/2016/01/Email_Statistics_Report_2016-2020_Executive_Summary.pdf). Last visited: 21-06-2017.

- [10] The Norwegian Intelligence Service. Focus, 2016. URL <https://forsvaret.no/en/ForsvaretDocuments/Focus%202016%20English.pdf>. Last visited: 21-06-2017.
- [11] Microsoft Security. Microsoft security intelligence report january through june 2016, 2016. URL <https://www.microsoft.com/en-us/security/intelligence-report/>. Last visited: 21-06-2017.
- [12] Techtarget. Uniform resource locator, 2017. URL <http://searchnetworking.techtarget.com/definition/URL>. Last visited: 26-06-2017.
- [13] Norwegian National Security Authority. Helhetlig IKT-risikobilde, 2016. URL [https://www.nsm.stat.no/globalassets/rapporter/nsm\\_helhetlig\\_ikt\\_risikobilde\\_2016\\_web\\_enkel.pdf](https://www.nsm.stat.no/globalassets/rapporter/nsm_helhetlig_ikt_risikobilde_2016_web_enkel.pdf). Last visited: 21-06-2017.
- [14] The Norwegian Business and Industry Security Council. Norwegian computer crime and data breach survey 2016, 2016. URL [http://www.nsr-org.no/getfile.php/Bilder/M%C3%B8rketallsunders%C3%B8kelsen/morketallsundersokelsen\\_2016\\_eng.pdf](http://www.nsr-org.no/getfile.php/Bilder/M%C3%B8rketallsunders%C3%B8kelsen/morketallsundersokelsen_2016_eng.pdf). Last visited: 21-06-2017.
- [15] Norwegian Centre for Information Security. The Norwegian Cyber Security Culture, 2016. URL <https://norsis.no/wp-content/uploads/2016/09/The-Norwegian-Cybersecurity-culture-web.pdf>. Last visited: 21-06-2017.
- [16] Jamie Carter. Forget IM: why email is still the ultimate form of online communication, 2014. URL <http://www.techradar.com/news/internet/you-ve-got-mail-why-email-is-the-enduring-form-of-internet-communication-1263200>. Last visited: 21-06-2017.
- [17] Roar Thon. Sikkerhetstilstanden er ikke tilfredsstillende, 2013. URL <http://blogg.nsm.stat.no/index.html%3Fp=3092.html>. Last visited: 21-06-2017.
- [18] Norwegian National Security Authority. Fire effektive tiltak mot dataangrep, 2016. URL <https://www.nsm.stat.no/globalassets/dokumenter/veiledninger/systemteknisk-sikkerhet/s-01-fire-effektive-tiltak-mot-dataangrep.pdf>. Last visited: 21-06-2017.
- [19] Norwegian National Security Authority. Ti viktige tiltak mot dataangrep, 2016. URL <https://www.nsm.stat.no/globalassets/dokumenter/veiledninger/systemteknisk-sikkerhet/s-02-ti-viktige-tiltak-mot-dataangrep.pdf>. Last visited: 21-06-2017.
- [20] Techtarget. Computer emergency readiness team, 2017. URL <http://whatis.techtarget.com/definition/CERT-Computer-Emergency-Readiness-Team>. Last visited: 26-06-2017.
- [21] Padma Tyagi, Kavita Misra. Advanced technical communication, 2007. URL [https://books.google.no/books/about/ADVANCED\\_TECHNICAL\\_COMMUNICATION.html?id=GslbGCG7a2kC&redir\\_esc=y](https://books.google.no/books/about/ADVANCED_TECHNICAL_COMMUNICATION.html?id=GslbGCG7a2kC&redir_esc=y). Page 12. Last visited: 21-06-2017.

- [22] Techtarget. Request for comments, 2017. URL <http://whatis.techtarget.com/definition/Request-for-Comments-RFC>. Last visited: 26-06-2017.
- [23] Internet Engineering Task Force. Rfc822, standard for the format of arpa internet text messages, 1982. URL <https://datatracker.ietf.org/doc/rfc822/>. Last visited: 22-06-2017.
- [24] Techtarget. Internet engineering task force, 2017. URL <http://searchmicroservices.techtarget.com/definition/IETF-Internet-Engineering-Task-Force>. Last visited: 26-06-2017.
- [25] Internet Engineering Task Force. Rfc2822, internet message format, 2001. URL <https://datatracker.ietf.org/doc/rfc2822/>. Last visited: 22-06-2017.
- [26] Internet Engineering Task Force. Rfc5321, simple mail transfer protocol, 2008. URL <https://datatracker.ietf.org/doc/rfc5321/>. Last visited: 22-06-2017.
- [27] Internet Engineering Task Force. Rfc 6854, update to internet message format to allow group syntax in the "from:" and "sender:" header fields, 2013. URL <https://datatracker.ietf.org/doc/rfc6854/>. Last visited: 22-06-2017.
- [28] Internet Engineering Task Force. Rfc821, simple mail transfer protocol, 1982. URL <https://datatracker.ietf.org/doc/rfc821/>. Last visited: 22-06-2017.
- [29] Internet Engineering Task Force. Rfc2821, simple mail transfer protocol, 2001. URL <https://datatracker.ietf.org/doc/rfc2821/>. Last visited: 22-06-2017.
- [30] Internet Engineering Task Force. Rfc 3501, internet message access protocol, 2003. URL <https://tools.ietf.org/html/rfc3501>. Last visited: 22-06-2017.
- [31] Internet Engineering Task Force. Rfc 2045, multipurpose internet mail extensions (mime) part one: Format of internet message bodies, 1996. URL <https://datatracker.ietf.org/doc/rfc2045/>. Last visited: 22-06-2017.
- [32] Internet Engineering Task Force. Rfc 2046, multipurpose internet mail extensions (mime) part two: Media types, 1996. URL <https://datatracker.ietf.org/doc/rfc2046/>. Last visited: 22-06-2017.
- [33] National Institute of Standards and Technology. Guidelines on electronic mail security, 2007. URL <http://csrc.nist.gov/publications/nistpubs/800-45-version2/SP800-45v2.pdf>. Last visited: 21-06-2017.
- [34] Techopedia. Message transfer agent, 2017. URL <https://www.techopedia.com/definition/1691/message-transfer-agent-mta>. Last visited: 26-06-2017.
- [35] Techtarget. Mail user agent, 2017. URL <http://searchnetworking.techtarget.com/definition/mail-user-agent>. Last visited: 26-06-2017.
- [36] The SPF Project. Introduction to sender policy framework, 2010. URL <http://www.openspf.org/Introduction>. Last visited: 22-06-2017.

- [37] Internet Engineering Task Force. Rfc 7208, sender policy framework (spf) for authorizing use of domains in email, 2014. URL <https://tools.ietf.org/html/rfc7208>. Last visited: 22-06-2017.
- [38] Techtarget. Domain name system, 2017. URL <http://searchnetworking.techtarget.com/definition/domain-name-system>. Last visited: 26-06-2017.
- [39] Adventures in security. Email authentication with sender id, 2006. URL <http://adventuresinsecurity.com/blog/?p=78>. Last visited: 22-06-2017.
- [40] Google Security Blog. Internet-wide efforts to fight email phishing are working, 2016. URL <https://security.googleblog.com/2013/12/internet-wide-efforts-to-fight-email.html>. Last visited: 22-06-2017.
- [41] Terry Zink. Where email authentication falls flat at stopping phishing – impersonation attacks using display tricks, 2016. URL <https://blogs.msdn.microsoft.com/tzink/2016/12/06/where-email-authentication-falls-flat-at-stopping-phishing-impersonation-attacks-using-display-tricks/>. Last visited: 23-06-2017.
- [42] DMARC. Faq, i operate a mailing list and i want to interoperate with dmarc, what should i do?, 2016. URL [https://dmarc.org/wiki/FAQ#I\\_operate\\_a\\_mailing\\_list\\_and\\_I\\_want\\_to\\_interoperate\\_with\\_DMARC.2C\\_what\\_should\\_I\\_do.3F](https://dmarc.org/wiki/FAQ#I_operate_a_mailing_list_and_I_want_to_interoperate_with_DMARC.2C_what_should_I_do.3F). Last visited: 23-06-2017.
- [43] Microsoft Office Support. How office helps protect you from phishing schemes, 2017. URL <https://support.office.com/en-US/article/How-Office-helps-protect-you-from-phishing-schemes-BE0DE46A-29CD-4C59-AAAF-136CF177D>. Last visited: 29-06-2017.
- [44] DKIM. Dkim introduction, 2005. URL <http://www.dkim.org/#introduction>. Last visited: 23-06-2017.
- [45] IETF. Domainkeys identified mail (dkim) signatures, 2007. URL <https://tools.ietf.org/html/rfc4871>. Last visited: 23-06-2017.
- [46] IETF. Domainkeys identified mail (dkim) signatures, 2011. URL <https://tools.ietf.org/html/rfc6376>. Last visited: 23-06-2017.
- [47] Matt Moorehead. How to explain dkim in plain english, 2015. URL <https://blog.returnpath.com/how-to-explain-dkim-in-plain-english-2/>. Last visited: 23-06-2017.
- [48] DMARC. What is dmarc?, 2017. URL <https://dmarc.org/>. Last visited: 23-06-2017.
- [49] IETF. Domain-based message authentication, reporting, and conformance (dmarc), 2015. URL <https://tools.ietf.org/html/rfc7489>. Last visited: 23-06-2017.

- [50] Danielle Tristao. What is identifier alignment?, 2015. URL <https://agari.zendesk.com/hc/en-us/articles/202952519-What-is-Identifier-Alignment->. Last visited: 23-06-2017.
- [51] LEARNTEMAIL. Dmarc secured your email identity, but see how it ruined mailing lists, 2017. URL <https://learntemail.sam.today/blog/dmarc-secured-your-email-identity-but-see-how-it-ruined-mailing-lists/>. Last visited: 29-06-2017.
- [52] Terry Zink. Solving the problem of dmarc’s incompatibility with mailing lists – part 1, 2015. URL <https://blogs.msdn.microsoft.com/tzink/2015/05/28/solving-the-problem-of-dmarcs-incompatibility-with-mailing-lists-part-1/>. Last visited: 02-07-2017.
- [53] Terry Zink. Solving the problem of dmarc’s incompatibility with mailing lists – part 2, 2015. URL <https://blogs.msdn.microsoft.com/tzink/2015/05/28/three-options-for-solving-the-problem-of-dmarcs-incompatibility-with-mailing-lists-part-2/>. Last visited: 02-07-2017.
- [54] Terry Zink. Solving the problem of dmarc’s incompatibility with mailing lists – part 3, 2015. URL <https://blogs.msdn.microsoft.com/tzink/2015/05/29/a-fourth-option-for-solving-the-problem-of-dmarcs-incompatibility-with-mailing-lists-part-3/>. Last visited: 02-07-2017.
- [55] Microsoft TechNet. Spam confidence levels, 2017. URL <https://technet.microsoft.com/en-us/library/jj200686.aspx>. Last visited: 26-06-2017.
- [56] Microsoft TechNet. Anti-spam message headers, 2016. URL [https://technet.microsoft.com/en-us/library/dn205071\(v=exch.150\).aspx](https://technet.microsoft.com/en-us/library/dn205071(v=exch.150).aspx). Last visited: 26-06-2017.
- [57] Microsoft TechNet. Antispam stamps, 2016. URL [https://technet.microsoft.com/en-us/library/aa996878\(v=exch.160\).aspx](https://technet.microsoft.com/en-us/library/aa996878(v=exch.160).aspx). Last visited: 23-06-2017.
- [58] Brien Posey. 10 tips for spotting a phishing email, 2015. URL <http://www.techrepublic.com/blog/10-things/10-tips-for-spotting-a-phishing-email/>. Last visited: 22-06-2017.
- [59] Terry Zink. Security talk, 2017. URL <https://blogs.msdn.microsoft.com/tzink/>. Last visited: 02-07-2017.
- [60] Ian Bittner, Kurt Spence. *Managing Iterative Software Development Projects*. Pearson Education Inc, 2007. Page 4.
- [61] ReviverSoft. .msg file extension, 2017. URL <http://www.reviversoft.com/file-extensions/msg>. Last visited: 24-06-2017.
- [62] ReviverSoft. .eml file extension, 2017. URL <http://www.reviversoft.com/file-extensions/eml>. Last visited: 24-06-2017.

- [63] Microsoft Developer Network. An introduction to javascript object notation (json) in javascript and .net, 2007. URL <https://msdn.microsoft.com/en-us/library/bb299886.aspx>. Last visited: 24-06-2017.
- [64] w3schools. Json objects, 2017. URL [https://www.w3schools.com/js/js\\_json\\_objects.asp](https://www.w3schools.com/js/js_json_objects.asp). Last visited: 24-06-2017.
- [65] tutorialspoint. Mongodb tutorial, 2017. URL <https://www.tutorialspoint.com/mongodb/>. Last visited: 24-06-2017.
- [66] Python. imaplib — imap4 protocol client, 2017. URL <https://docs.python.org/2/library/imaplib.html>. Last visited: 24-06-2017.
- [67] Jung Paul Totg Georges. Python eml parser module, 2013. URL [https://github.com/GOVCERT-LU/eml\\_parser](https://github.com/GOVCERT-LU/eml_parser). Last visited: 24-06-2017.
- [68] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations, 1996. URL <https://www.mat.ucsb.edu/g.legrady/academic/courses/11w259/schneiderman.pdf>. Last visited: 25-06-2017.
- [69] OWASP. Top 10 application security risks - 2017, 2017. URL [https://www.owasp.org/index.php/Top\\_10\\_2017-Top\\_10](https://www.owasp.org/index.php/Top_10_2017-Top_10). Last visited: 25-06-2017.
- [70] Slim. About slim, 2017. URL <https://www.slimframework.com/>. Last visited: 25-06-2017.
- [71] PHP. What is php?, 2017. URL <http://php.net/manual/en/intro-whatis.php>. Last visited: 25-06-2017.
- [72] Sensiolabs. Twig is a modern template engine for php, 2017. URL <https://twig.sensiolabs.org/>. Last visited: 25-06-2017.
- [73] Mozilla Developer Network. Javascript, 2017. URL <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Last visited: 25-06-2017.
- [74] MongoDB. Pymongo documentation, 2017. URL <https://api.mongodb.com/python/current/>. Last visited: 25-06-2017.
- [75] Universitetsavisa. Virus-epost sendt ut fra ntnu-kontoer, 2017. URL <http://www.universitetsavisa.no/campus/2017/04/24/Virus-epost-sendt-ut-fra-NTNU-kontoer-65759.ece>. Last visited: 28-06-2017.
- [76] Cuckoo Sandbox. Cuckoo sandbox, 2017. URL <https://cuckoosandbox.org/#about>. Last visited: 01-07-2017.
- [77] Xavier Mertens. Cuckoomx: Automating email attachments scanning with cuckoo, 2012. URL <https://blog.rootshell.be/2012/06/20/cuckoomx-automating-email-attachments-scanning-with-cuckoo/>. Last visited: 01-07-2017.

- [78] Sean Whalen. Howto: Build a cuckoo sandbox, 2015. URL <https://infosecspeakeasy.org/t/howto-build-a-cuckoo-sandbox/27>. Last visited: 01-07-2017.
- [79] Xavier Mertens. Cuckoomx, 2012. URL <https://github.com/xme/cuckoomx>. Last visited: 01-07-2017.



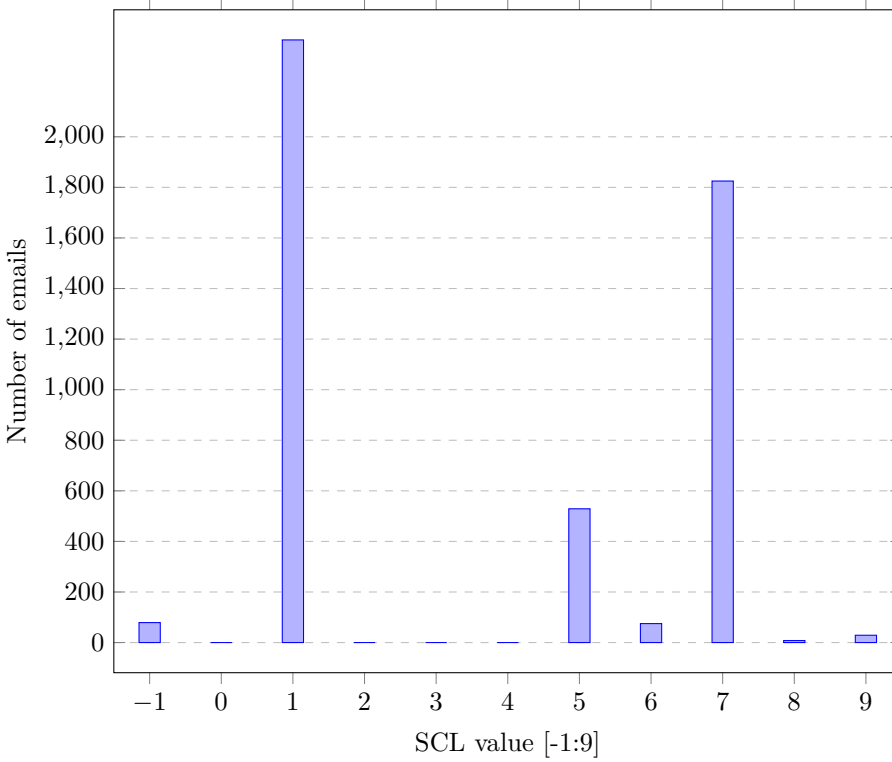


# Appendix

## X-MS-Exchange-Organization-SCL

**Table A.1:** Complementary data to graph in figure A.1

	Number of emails:	% of total emails:
Emails with a SCL header value	4928	98.56
Emails without a SCL header value	72	1.44
Total number of emails in dataset	5000	100.00



**Figure A.1:** X-MS-Exchange-Organization-SCL

**Table A.2:** Complementary data to graph in figure A.1

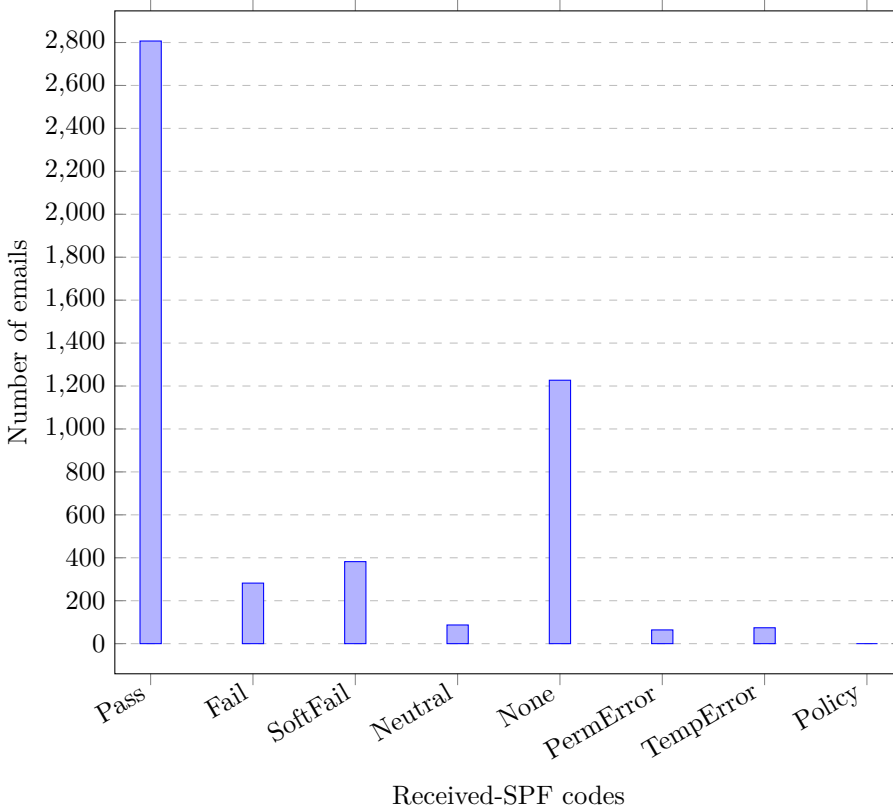
SCL values:	Number of emails:	% of total emails:
-1	79	1.58
0	0	0
1	2383	47.66
2	0	0
3	0	0
4	0	0
5	529	10.58
6	75	1.50
7	1825	36.50
8	8	0.16
9	29	0.58

# Appendix **B**

## Received-SPF

**Table B.1:** Complementary data to graph in figure B.1

	Number of emails:	% of total emails:
Emails with a Received-SPF header	4923	98.46
Emails without a Received-SPF header	77	1.54
Total number of emails in dataset	5000	100.00



**Figure B.1:** Received-SPF status

**Table B.2:** Complementary data to graph in figure B.1

Received-SPF codes:	Number of emails:	% of total emails:
Pass	2807	56.14
Fail	282	5.64
SoftFail	382	7.64
Neutral	87	1.74
None	1227	24.54
PermError	64	1.28
TempError	74	1.48
Policy	0	0

# Appendix

## DKIM-signature

**Table C.1:** DKIM-signature header data

	Number of emails:	% of total emails:
Emails with DKIM-signature	1426	28.52
Emails without DKIM-signature	3574	71.48
Total number of emails in dataset	5000	100.00

**Table C.2:** Top domains in DKIM-signature header field, more than 18 received

Domain name (d= )	Number of emails:
163.com	88
126.com	74
nam-mail.com	61
gmail.com	56
yahoo.com	38
tarim.gov.tr	38
sendinblue.com	34
hotmail.com	26
moononline.info	24
news.aussiesofferz.com	20
ayeagree.com	18



# Appendix **D**

## Authentication-Results

**Table D.1:** Complementary data to figure D.1, D.2 and D.3

	Number of emails:	% of total emails:
Emails with Authentication-Results	2900	58.00
Emails without Authentication-Results	2100	42.00
Total number of emails in dataset	5000	100.00

**Table D.2:** Complementary data to figure D.1

	Number of emails:	% of total emails:
without SPF in Authentication-Results	77	1.54
with SPF in Authentication-Results	2823	56.46

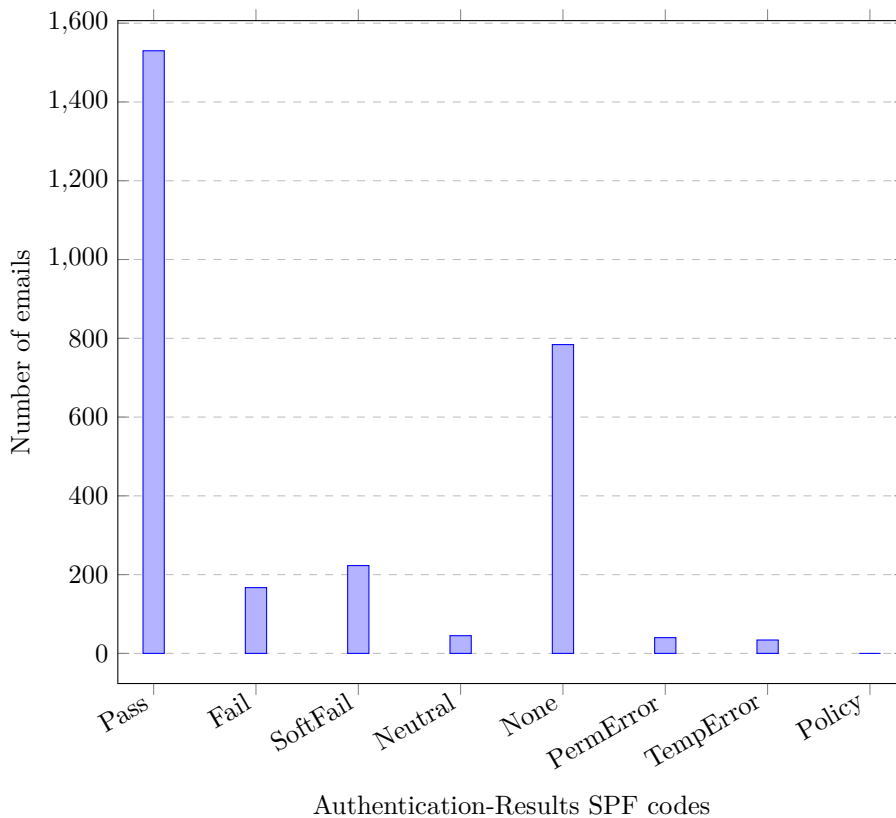
**Table D.3:** Complementary data to figure D.2

	Number of emails:	% of total emails:
Other not standard codes (4), NoValue(40)	44	0.88
without DKIM in Authentication-Results	12	0.24
with DKIM in Authentication-Results	2888	57.76

**Table D.4:** Complementary data to figure D.3

	Number of emails:	% of total emails:
Other not standard codes	648	12.96
without DMARC in Authentication-Results	38	0.76
with DMARC in Authentication-Results	2862	57.24

### D.1 Authentication-Results SPF



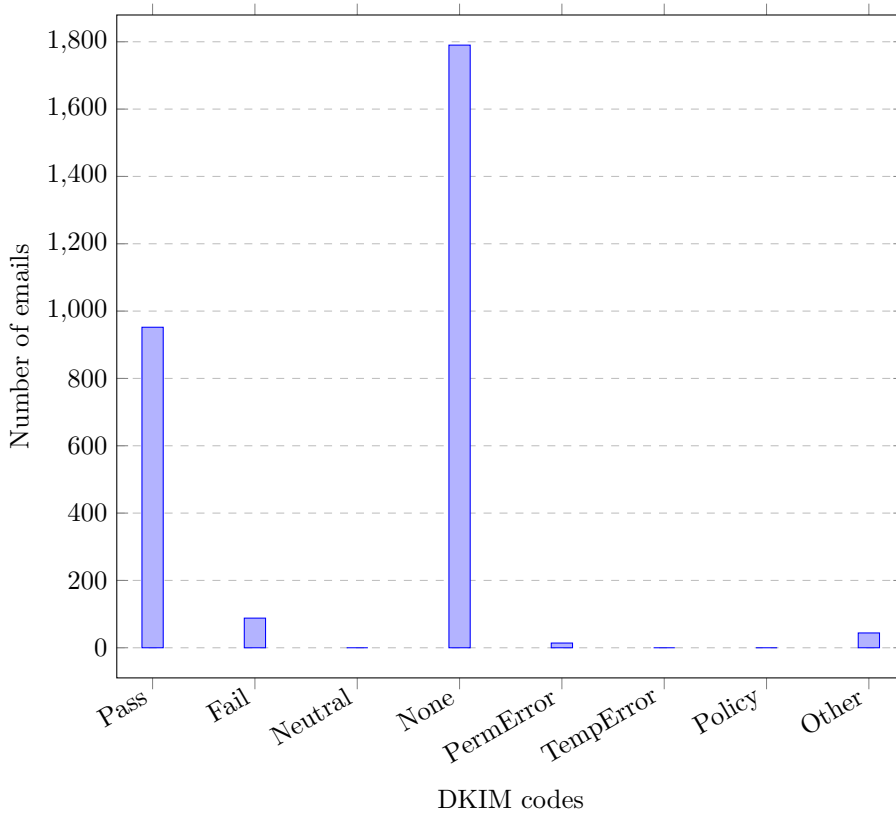
**Figure D.1:** Authentication-Results SPF status

**Table D.5:** Complementary data to figure D.1

Authentication-Results SPF codes:	Number of emails:	% of total emails:
Pass	1530	30.60
Fail	167	3.34
SoftFail	223	4.46
Neutral	45	0.90
None	784	15.68
PermError	40	0.80
TempError	34	0.68
Policy	0	0



## D.2 Authentication-Results DKIM

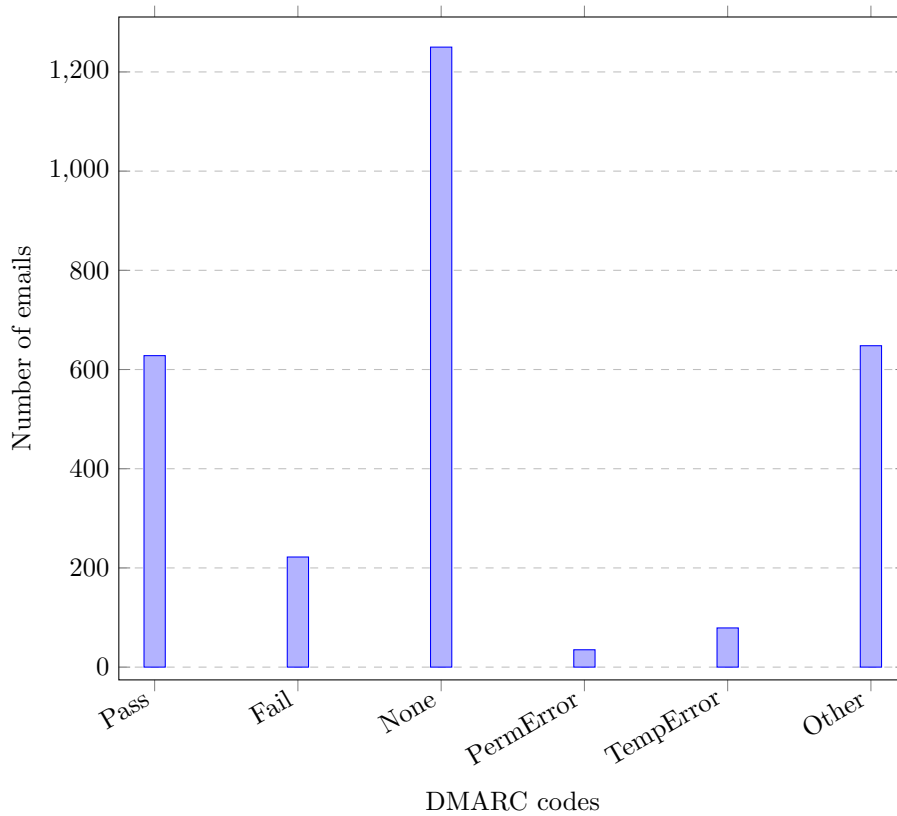


**Figure D.2:** Authentication-Results DKIM status

**Table D.6:** Complementary data to figure D.2

DKIM codes:	Number of emails:	% of total emails:
Pass	952	19.04
Fail	88	1.76
Neutral	0	0
None	1790	35.80
PermError	14	0.28
TempError	0	0
Policy	0	0

### D.3 Authentication-Results DMARC



**Figure D.3:** Authentication-Results DMARC status

**Table D.7:** Complementary data to figure D.3

DMARC codes:	Number of emails:	% of total emails:
Pass	628	12.56
Fail	222	4.44
None	1250	25.00
PermError	35	0.70
TempError	79	1.58

# Appendix **E**

## **X-Forefront-Antispam-Report**

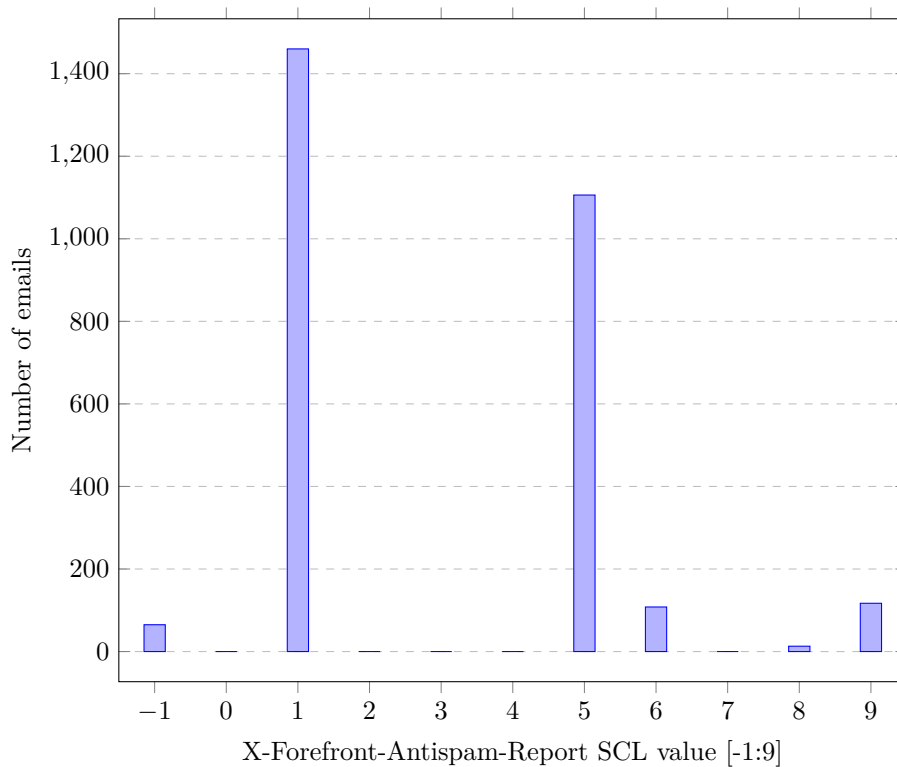
**Table E.1:** Complementary data to figures E.1 and E.2.

	Number of emails:	% of total emails:
Emails with X-Forefront-Antispam	2869	57.38
Emails without X-Forefront-Antispam	2131	42.62
Total number of emails in dataset	5000	100.00

### **E.1 X-Forefront-Antispam SCL**

**Table E.2:** Complementary data to figure E.1

SCL values:	Number of emails:	% of total emails:
-1	65	1.30
0	0	0
1	1460	29.20
2	0	0
3	0	0
4	0	0
5	1106	22.12
6	108	2.16
7	0	0
8	13	0.26
9	117	2.34

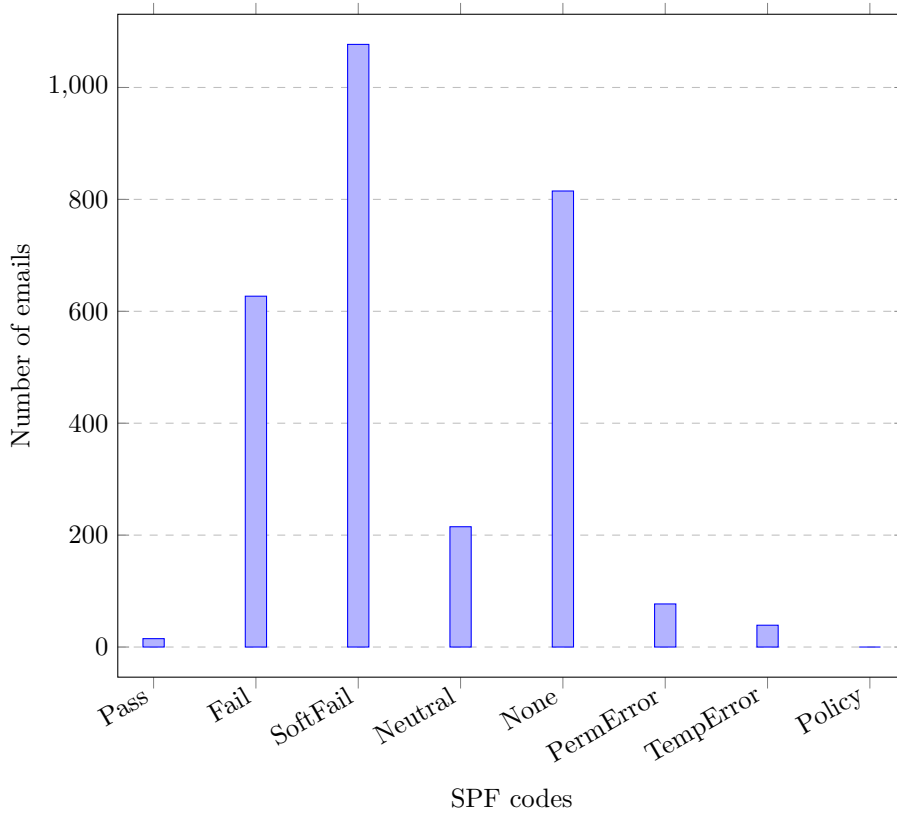


**Figure E.1:** X-Forefront-Antispam-Report SCL status

## E.2 X-Forefront-Antispam SPF

**Table E.3:** Complementary data to figure E.2

SPF codes:	Number of emails:	% of total emails:
Pass	15	0.30
Fail	627	12.54
SoftFail	1077	21.54
Neutral	215	4.30
None	815	16.30
PermError	77	1.54
TempError	39	0.78
Policy	0	0
without SPF in X-Forefront-Antispam-Report	4	0.08
with SPF in X-Forefront-Antispam-Report	2865	57.30



**Figure E.2:** X-Forefront-Antispam SPF status



# Appendix **F**

## Comparison of results from email headers

- **Table F.1** X-MS-Exchange-Organization-SCL and Received-SPF
- **Table F.2** X-Forefront-Antispam-Report SCL and X-MS-Exchange-Organization-SCL
- **Table F.3** X-Forefront-Antispam-Report SPF and Received-SPF
- **Table F.4** X-Forefront-Antispam-Report SPF and Authentication-results SPF
- **Table F.5** Authentication-Results SPF and Received-SPF

## F.1 X-MS-Exchange-Organization-SCL and Received-SPF

Table F.1: X-MS-Exchange-Organization-SCL and Received-SPF

SCL/SPF	Pass	Fail	SoftFail	Neutral	None	PermError	TempError	Policy	No Value	Total:
-1	57	2	1	0	0	0	1	0	16	79
0	0	0	0	0	0	0	0	0	0	0
1	1774	28	88	14	428	22	29	0	0	2383
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	256	51	48	32	133	5	4	0	0	529
6	38	7	9	3	15	0	3	0	0	75
7	670	183	235	36	628	37	36	0	0	1825
8	0	8	0	0	0	0	0	0	0	8
9	8	2	1	1	16	0	1	0	0	29
No Value	4	1	0	1	5	0	0	0	61	72
Total:	2807	282	382	87	1227	64	74	0	77	5000



## F.2 X-Forefront-Antispam-Report SCL and X-MS-Exchange-Organization-SCL

**Table F.2:** X-Forefront-Antispam-Report SCL and X-MS-Exchange-Organization-SCL

SCL	-1	0	1	2	3	4	5	6	7	8	9	No Value	Total:
-1	64	0	0	0	0	0	0	0	1	0	0	0	65
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1456	0	0	0	0	0	0	0	0	4	1460
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	285	0	820	0	0	1	1106
6	0	0	0	0	0	0	0	43	65	0	0	0	108
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	8	5	0	0	13
9	0	0	0	0	0	0	0	0	103	0	14	0	117
No Value	15	0	927	0	0	0	244	32	828	3	15	67	2131
Total:	79	0	2383	0	0	0	529	75	1825	8	29	72	5000

### F.3 X-Forefront-Antispam-Report SPF and Received-SPF

Table F.3: X-Forefront-Antispam-Report SPF and Received-SPF

SPF	Pass	Fail	SoftFail	Neutral	None	PermError	TempError	Policy	No Value	Total:
Pass	15	0	0	0	0	0	0	0	0	15
Fail	490	133	0	0	2	0	2	0	0	627
SoftFail	864	2	197	0	2	5	7	0	0	1077
Neutral	173	0	0	42	0	0	0	0	0	205
None	73	13	7	0	712	0	8	0	2	815
PermError	37	0	2	0	0	38	0	0	0	77
TempError	13	0	1	1	3	0	21	0	0	39
Policy	0	0	0	0	0	0	0	0	0	0
No Value	1142	134	175	44	508	21	36	0	75	2135
Total:	2807	282	382	87	1227	64	74	0	77	5000

## F.4 X-Forefront-Antispam-Report SPF and Authentication-results SPF

**Table F.4:** X-Forefront-Antispam-Report SPF and Authentication-results SPF

SPF	Pass	Fail	SoftFail	Neutral	None	PermError	TempError	Policy	No Value	Total:
Pass	14	0	0	0	0	0	0	0	1	15
Fail	360	112	0	0	1	0	2	0	152	627
SoftFail	662	2	163	0	1	5	4	0	240	1077
Neutral	121	0	0	35	0	0	0	0	59	215
None	66	10	5	0	638	0	6	0	90	815
PermError	18	0	0	0	0	29	0	0	30	77
TempError	10	0	1	1	3	0	12	0	12	39
Policy	0	0	0	0	0	0	0	0	0	0
No Value	279	43	54	9	141	6	10	0	1593	2135
Total:	1530	167	223	45	784	40	34	0	2177	5000

## F.5 Authentication-Results SPF and Received-SPF

Table F.5: Authentication-Results SPF and Received-SPF

SPF	Pass	Fail	SoftFail	Neutral	None	PermError	TempError	Policy	No Value	Total:
Pass	1530	0	0	0	0	0	0	0	0	1530
Fail	1	166	0	0	0	0	0	0	0	167
SoftFail	0	0	223	0	0	0	0	0	0	223
Neutral	0	0	0	45	0	0	0	0	0	45
None	0	0	0	0	784	0	0	0	0	784
PermError	0	0	0	0	0	40	0	0	0	40
TempError	0	0	0	0	0	0	34	0	0	34
Policy	0	0	0	0	0	0	0	0	0	0
No Value	1276	116	159	42	443	24	40	0	77	2177
Total:	2807	282	382	87	1227	64	74	0	77	5000