



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Social-Engineering Experiment

Master's Thesis

Marco Studer

`studmarc@student.ethz.ch`

System Security Group
Institute of Information Security
ETH Zurich

Supervisors:

Dr. Raphael Reischuk
Prof. Dr. Srdjan Capkun

August 13, 2018

The icons used to indicate quotes and mitigation strategies appear under the Creative Commons Attribution 4.0 International license at

<https://fontawesome.com/license>.

No changes were made to the icons. Full credit is given to FontAwesome.

Acknowledgements

I thank Raphael Reischuk for his tireless effort and for the immense amount of time he invested in me and the project. His enthusiasm motivated me throughout the project. He taught me countless things and was more friend than supervisor.

I thank [YuuBlue](#) for giving me the opportunity to write this thesis. I cost [YuuBlue](#) and its employees a lot of time and nerves. I am thankful for their patience and for being so understanding.

Abstract

Social engineering, the act of manipulating people into committing actions beneficial to an attacker (and most likely detrimental to the manipulated target), is as old as time itself and is likely to remain a dangerous threat to organizations and individuals alike — especially today, in times where information is more valuable than ever before.

In order to better understand the implications of social engineering, this thesis constitutes research on social engineering in the context of an internationally operating organization. More specifically, the thesis presents the planning and execution of various social-engineering attacks that were performed over the course of approximately two months. All attacks were conducted without any insider knowledge about the targeted organization. None of the employees knew of the attacks, except for two members of management who approved and supported the project.

The findings include weaknesses regarding (i) the access to physical assets, (ii) the application of business processes, and (iii) the disclosure of confidential information, such as bank account numbers and salary statements; names, positions, and email addresses of employees; as well as network and account passwords.

These findings convincingly demonstrate that neither a formal education nor experience in the field of social engineering is required to successfully attack an international organization.

Contents

Acknowledgements	ii
Abstract	iii
Part I Ab Origine	1
1 Introduction	1
1.1 Why This Thesis?	1
1.2 The Company	2
1.3 The Employees	3
1.4 A Note on Preserving the Anonymity	3
1.5 What to Expect from This Thesis	4
1.6 Security Disclaimer	4
2 Definitions	5
2.1 Social Engineering	5
2.1.1 Phishing	8
2.1.2 Vishing	8
2.1.3 Pretexting	9
2.1.4 Water Holing	9
2.1.5 Baiting	10
2.1.6 Tailgating / Piggybacking	10
2.1.7 Dumpster Diving	11
2.1.8 Shoulder Surfing	11
2.2 Open-Source Intelligence (OSINT)	11
2.3 Malware	12
2.4 Denial of Service (DoS)	13
2.5 Man-in-the-Middle Attack (MitM)	13

2.6	Look-Alike Domain	13
2.7	Metasploit Framework	14
2.7.1	Meterpreter	14
2.8	Shell	15
2.9	Zero-Day Vulnerability	15
2.10	Regular Expression	16
2.11	Antivirus Software	16
2.11.1	Signature-based Detection	17
2.11.2	Behavior-based Detection	17
2.11.3	Antivirus Detection Evasion Techniques	18
2.12	Intrusion Detection System (IDS)	19
2.13	Greylisting	20
2.14	Sender Policy Framework (SPF) And Its Successors	20
3	Preparation	23
3.1	Virtual Private Server	23
3.2	Creating Fake Email Accounts	25
3.3	Creating a Payload	27
3.3.1	Evading Antivirus Software	27
3.3.2	Custom Payload	28
3.4	Creating a Malicious Word Document	28
4	Sources of Information	30
4.1	Email Addresses	30
4.1.1	Via the Company Website	30
4.1.2	Via LinkedIn	33
4.2	Subdomains via DNS Enumeration	34
4.3	Availability via Out-of-Office Replies	34
4.4	Names and Maps via YouTube	35
4.5	Organigram via the Company Website	35
4.6	Information via Social Media Networks	36

Part II Acta Non Verba	37
5 Attack Categorization and Overview	38
5.1 Unexecuted Attacks	40
6 SED04: The Mass Phishing	42
6.1 Preparation	42
6.2 Execution History	45
6.3 Findings and Post-Execution Insights	53
7 SED07: The Name Inquiry	60
7.1 Preparation	60
7.2 Execution History	61
7.3 Findings and Post-Execution Insights	65
8 SED09: The OoO Stress Scam	66
8.1 Preparation	66
8.2 Execution History	67
8.3 Findings and Post-Execution Insights	72
9 SED11: The Leaver	74
9.1 Preparation	74
9.2 Execution History	76
9.3 Findings and Post-Execution Insights	80
10 SED12: The OoO Provoker	82
10.1 Preparation	82
10.2 Execution History	85
10.3 Findings and Post-Execution Insights	89
11 SED13: The CV Crawler	92
11.1 Preparation	92
11.2 Execution History	93
11.3 Findings and Post-Execution Insights	94

12 SED15: The Customer Summary	95
12.1 Preparation	95
12.2 Execution History	96
12.3 Findings and Post-Execution Insights	98
13 SED16: Attack from Within	99
13.1 Preparation	99
13.2 Execution History	100
13.3 Findings and Post-Execution Insights	102
14 SED18: The Flash Mob	104
14.1 Preparation	104
14.2 Execution History	105
14.3 Findings and Post-Execution Insights	107
15 SED19: Orders from Above	109
15.1 Preparation	109
15.2 Execution History	110
15.3 Findings and Post-Execution Insights	111
16 SEP01: The Fake Invoice	112
16.1 Preparation	112
16.2 Execution History	113
16.3 Findings and Post-Execution Insights	117
17 SEP02: The Security Interview	118
17.1 Preparation	118
17.2 Execution History	119
17.3 Findings and Post-Execution Insights	119
18 SEP03: The Anniversary	121
18.1 Preparation	121
18.2 Execution History	123
18.3 Findings and Post-Execution Insights	126

CONTENTS	viii
19 SEP04: The Market Research Insitute	128
19.1 Preparation	128
19.2 Execution History	129
19.3 Findings and Post-Execution Insights	130
Part III Ad Meliora	132
20 Summary of Findings	133
20.1 Data	133
20.2 General Findings and Interpretation	134
20.3 Weaknesses	140
21 Mitigation and Countermeasures	146
21.1 A Culture of Security Awareness	146
21.2 Plans, Protocols, and Procedures	147
21.3 Educate, Drill, Repeat	153
22 Closing Remarks	154
Glossary	156
List of Mitigations	159
Bibliography	160
A Code	A-1
B PDF Documents	B-1
B.1 SED11: The Leaver	B-2
B.2 SED19: Orders from Above	B-3
B.3 SEP01: The Fake Invoice	B-4
B.4 SEP02: The Security Interview	B-5

Part I

Ab Origine

From the Beginning

Introduction

The importance of security grows more and more in a world where information is omnipresent and powerful, where all the knowledge of the world is at one's fingertip, and where having information about the person opposite is an invaluable asset. For all the good the so-called "Information Age" has done us, it has also opened the doors to our very living room, for everyone who cares to look. Organized crime scaled stealing and selling information into a multi-billion dollar business and pretty much every company is a target.

This Master's thesis, dubbed **Social-Engineering Experiment (SOEP)**, analyzes at how valuable information can be stolen from companies and how companies can protect themselves. A number of methods and attack schemes have been executed in a real environment, targeting an international corporation. Every step was meticulously documented, starting with the preparation and going all the way to the execution.

1.1 Why This Thesis?

In the last years, the number of sizable information and identity breaches has increased steadily. With the number of breaches, the average number of exposed identities per breach has increased as well. These developments come with enormous costs for afflicted companies and organizations, who have to pay significant compensation sums and penalties, accompanied by severe reputational losses [1].

According to the market research institute IDC, more than two thirds of all companies in Germany have been successfully attacked in the last two years, while the users remain the main risk factor [2].

In the face of these numbers, many organizations have recognized that traditional measures such as antivirus software do no longer suffice in the battle against hackers and social engineers. While organizations need to defend their assets against *all* possible attacks, the attackers need only to find a *single* weak link in the chain, shifting the advantage to the side of the attackers.

To test the strength of their security systems, many companies and organizations, such as Twitter, Facebook, and even the US Air Force and the Pentagon, have started to engage white-hat hackers, trying to find the weak links in their chains of defense [3][4]. The idea is that if an organization *hacks itself first*¹, it can find vulnerabilities before criminals do. The money seems well-invested since data breach costs are soaring.

The focus of these so-called *penetration tests* is by no means to control or expose the employees of a company. On the contrary, the goal is to educate and train the employees to be able to cope with cybercrime. The ability to recognize and withstand social-engineering attacks is getting more important with every day and is useful in all situations of life.

Even though many organizations employ social engineering to test their strengths and weaknesses, little is known about the results of these penetration tests, and even less is known about the extent of real data breaches. The lack of published data is no surprise, considering the fact that organizations stand to lose significant amounts of money if the trust in their capabilities to secure assets or data is diminished.

To shed light on this unexplored area and the many unknown numbers, this thesis provides a detailed description of the investment, the knowledge, the tools, and the techniques required, as well as a description of the findings of a (typical) social-engineering attack. The thesis explores the amount of time, skills, and experience necessary to execute a social-engineering attack. All cards are on the table.

1.2 The Company

The enterprise against which this social-engineering experiment has been executed is referred to as *YuuBlue*, a Swiss corporation, with an employee count between 1 000 and 20 000, located in *Bern*. *YuuBlue* operates internationally, but was founded in Switzerland.

Two representatives of the company's management approved this social-engineering project. Only six people related to *YuuBlue* (i.e., less than 1% of the employees) knew about project SOEP, including the head of customer solutions, the head of human resources, and the head of the legal department. To keep the project as effective as possible, its existence was not divulged to *YuuBlue*'s CEO.

¹“Hack Yourself First” is a concept coined by WhiteHat Security. The company believes that cyber-offense is a means to cyber-defense [5].

1.3 The Employees

In order to successfully attack a company and to steal sensitive information, procuring information about the company's employees is an invaluable preparation task. Because of the strong online presence that a lot of companies and organizations have nowadays, gathering intelligence does not require a big effort. Reading the company's website, blog, flyers, or other resources, both online and offline, is an excellent first step to get acquainted with the target.

Often, somewhere on the company's website, there is an organizational chart or list. This chart will provide the names and corresponding positions of the management team. After having obtained the management's names, social networks such as Facebook and LinkedIn can be used to gain further information on who these persons are, where they have worked, where they live, what language they speak, what universities and schools they went to and so on (see [Section 2.2](#) for a definition of open-source intelligence).

Besides information about the management, websites usually deliver a wealth of names, phone numbers, and email addresses, giving the attacker valuable and direct handles to the employees of a company.

1.4 A Note on Preserving the Anonymity

The corporation that is targeted in this thesis is henceforth going to be referred to as [YuuBlue](#). To protect the company and its employees, all data that could lead to the identification of an involved individual or entity is redacted and replaced by a placeholder in angular brackets. For example, the name Bruce Wayne will turn into [Batman](#) in this thesis. The gender of the placeholder names does not necessarily match the corresponding employee's gender. For example, Bruce Wayne could also be represented by [Catwoman](#).

The names of individuals were replaced by the names of famous, historical personalities. We employ these replacements because first, [Marilyn Monroe](#) is far easier to read than [Name of the Head of Marketing](#), and second, the thesis might, at times, become more entertaining to the reader.

While we replaced the names of individuals by historical persons, the names of companies were replaced by colorful names (for example, TechCrunch might become [BlackBlog](#)).

The names of cities and towns were replaced by the capital city of the corresponding country. For example, Geneva will turn into [Bern](#) in this thesis.

We write [yuublu.com](#) to identify the [YuuBlue](#) website.

1.5 What to Expect from This Thesis

Except for an education in the field of electrical engineering and information technology, the author did not have any training or experience on the topic of social engineering or hacking in general. The attacks and ideas presented in this Master's thesis can be found on the internet and can be executed without any prior experience. **Social engineering is not rocket science, which is precisely what makes it so dangerous.**

Every attack described in this thesis was executed *without* prior insider knowledge about the corporation. As a consequence, the attacks presented could have been launched by any outside attacker just as they are described.

While we have tried to execute as many attacks as possible in the given time frame, the attacks that we have executed and described in this thesis are by no means an exhaustive list, and they do not represent the vast field of social engineering in its entirety. When it comes to social engineering, creativity knows no bounds. Every single attack can be executed in a plethora of different settings and circumstances, with details varying every time the attack is executed.

We do not claim to have executed the attacks perfectly or in the most effective manner. We were, after all, amateurs, bound by contract, and benevolent human beings. However, when looking at our findings, at the data we illegitimately obtained, and the ease with which the attacks could be executed, it is distressing to **imagine what havoc a professional with many years of experience could potentially wreak.**

1.6 Security Disclaimer

We secured the transmission of any information that we obtained via HTTPS to the best of our abilities and influence. Whenever possible, sensitive information was stored and transmitted in encrypted form.

The names of the employees involved in the attacks were never exposed in order to protect them, even internally.

All attacks were executed within the legal bounds specified in a contract signed by the author and a representative of [YuuBlue](#). No law was broken in the process, and no contract was violated.

Definitions

This chapter presents a number of techniques and tools for carrying out social-engineering attacks. We start off with a definition of social engineering. Thereafter, we present attacking techniques, as well as some defense techniques. The list of definitions is scoped to cover the focus of this thesis.

2.1 Social Engineering

While technological attacks and threats (e.g., viruses, trojans, DoS attacks) against systems, networks, and infrastructure are dangerous and a nuisance, it is often possible to set up adequate defenses which can catch a substantial percentage of attacks. Defending against *social engineering* is much more difficult. Social engineering is the act of manipulating an entity into doing something the attacker wants the entity to do, but which may be detrimental to the entity itself or any entities associated with the attacked entity. Social engineering employs psychology and takes advantage of hardwired biological functions in our brains. These functions include, but are not limited to, the following compliance tendencies and behaviors, among them the key principles of the *theory of influence* established by Robert Cialdini: [6][7]

Authority Many companies, organizations, states, or even cultures are organized hierarchically, which instills a deep desire for obeying authority to avoid any backlashes or to advance the own position by having strong friends higher up the chain.

Will you deny information to your “boss” who writes from a private email address, and thus risk to be fired or demoted?

Liking People are more likely to be influenced by people or other entities they like. Furthermore, people are social beings and (often) naturally fond of other people with whom they try to be cooperative and helpful. Since “being helpful” might, in turn, increase the chances of being liked by another person, “being helpful” can effectively increase one’s social status.

Will you demand a badge or a name of the poor, hard-working guy in uniform with a heavy box in his arms who desperately tries to open a door with his elbows, or will you politely hold the door for him?

Reciprocity People usually treat other people as they want to be treated themselves. It includes, for example, returning a favor. In this way, people can be obliged to do something for you by simply doing them a favor.

Will you ask questions of the fellow smoker who just gave you a cigarette enters the building with you?

Unity People love like-minded people to whom they can relate. Like-minded people share commonalities such as interests, hobbies, religion, or nationality. If a person is in the same situation as another one, they usually sympathize and can influence each other (a sorrow shared is a sorrow halved).

Will you deny information to the poor lady on the phone who is a stressed mother, like you, on the brink of tears because of her crying daughter (or YouTube video?) in the background?

Commitment and Consistency People try to be true to themselves and to what they say. Social engineering makes people deviate from this ambition so that they say or do things they do not mean. Many people will find that, once they committed to something, it is hard for them to go back in their word, even if they were tricked into doing or saying something in the first place. People try to uphold their integrity and not go back on their words.

If, during a real discussion over lunch, a man at the table “tricked” you into saying salaries should be public, will you give your salary if so requested, or will you oppose your own word?

Social Proof When making a decision, people often look up to their peers or superiors. What they are doing must be reasonable and right, and there is no need to double-check facts or to think twice. In unknown situations that people did not encounter before and are therefore unsure how to react to, people will often do what their peers do because people think that their peers are better informed than they are and that their peers are therefore acting in the (only) right way.

If everybody is afraid to enter a building for no apparent reason, will you enter it?

If everybody suddenly takes off their coat, will you keep yours on?

Scarcity If a good is scarce, people often want it. An example is a discount which is only available for a short amount of time. Such a discount makes the window of decision small, thus inducing stress which in turn hinders clear thinking. Like this, an artificial scarcity can be induced on anything.

Will you attend the office party where everyone can go or the one that is limited to “special” staff, to which you belong since you got one of the rare invitations?

Curiosity People are generally curious. If an attacker leaves a wrapped box on a desk, the person sitting there and coming to work in the morning is hardly going to ignore it.

Will you open the “Salaries 2018” Excel sheet on the USB stick you found on your desk this morning?

All of these hardwired behavior patterns can be used against most people in most situations of life, and are sometimes called the “bugs in the human hardware” [8]. They have been used for hundreds of years, the most famous example of which is the original Trojan horse, where the Trojans were social-engineered into carrying their enemy right into their midst, circumventing the Trojan security system, namely their walls and military.

The research community seems to largely agree that the weakest link in any security system are people, the so-called “wetware” [2, 9, 10, 11, 12]. Though some say, the weakest link is ever-shifting [13].

Social-engineering attacks usually follow the same pattern: research, hook, play, and exit [9].

Research The social engineer gathers information about the target. Different sources can be used, such as social media, personal or company websites, dumpster diving (see Section 2.1.7), or even physically asking the target or people associated with the target for information.

Hook The social engineer builds a relationship with the target by starting a conversation (either online or offline). The attacker tries to find ways to excite the target.

Play The social engineer continues to build trust by using any of the techniques mentioned above.

Exit In the last and crucial phase, the social engineer executes the attack (e.g., gaining information), stops to communicate with the target, and, if all went well, vanishes without the target being any wiser.

Social engineering is a creative field with ever-changing attacks, ideas, and schemes. The following sections present some of the more popular attacks of social engineering, but are by no means a complete list.

2.1.1 Phishing

The term *phishing* denotes the procedure of tricking a target into opening a link or a document by posing as a trusted entity, in order to gain further information by either letting the target itself surrender it (e.g., by entering data on a website) or by installing malware upon opening an infected document or file.

Phishing attacks do usually not require much effort and are thus extremely common. The number of attacks is steadily increasing with no end in sight [14]. Even worse, alongside quantity, quality is increasing too. It becomes more difficult by the day to tell real websites from scams, despite the fact that users are regularly being warned by their employees, by email providers, and by the media. Even though users grow more attentive as a consequence [15], the quality of phishing seems to be able to counter this development easily so far. Phishing is executed prevalently via email. However, other channels such as instant messaging are also being used [16].

Phishing emails often target as many people as possible to maximize the effect. However, in some cases, specifically chosen people are targeted. This type of phishing is called “**spear-phishing**” (in contrast to phishing with a “phishing net”, which is catching everyone). Spear-phishing emails can be quite sophisticated and are tailored to their respective targets which can make them very effective [16], resulting in high financial and reputational damage.

When spear-phishing attacks are aimed at high-profile targets, such as CEOs, one speaks of “**whaling**”.

We have used phishing in SED04 and SED12 (see [Chapter 6](#) and [Chapter 10](#)).

2.1.2 Vishing

Vishing is a combination of the terms “voice” and “phishing” and describes an attack that works similar to phishing (see [Section 2.1.1](#)), but over the phone [17].

Just like with phishing, the goal is to exfiltrate credentials, credit card numbers, or PIN codes from the target. Instead of luring the target via email onto a fake website where the target is tricked into entering its login credentials, the social engineers pretend to be a trustworthy entity, such as a bank, and convince the target that a call to a telephone number is necessary, for example, because fraudulent behavior was allegedly noticed on one of the target’s credit cards. To convince the target, social engineers can use different channels, with email being the most common channel because of its simplicity. If the email protocol is used as a channel, vishing is often combined with phishing [17].

The target who then calls the number provided by the social engineers is often met with a prerecorded message that automatically tries to extract information from the target by prompting it to enter a credit card number or similar infor-

mation. Alternatively, the social engineers may pick up the phone themselves and try to exfiltrate the information they are after directly from the target [17].

With the advent of artificial intelligence, software can fully autonomously perform calls to humans without them having any idea that they are connected to a machine. Google Duplex, “a technology for conducting natural conversations to carry out real-world tasks over the phone”, was recently presented at the Google I/O conference [18][19].

Social engineers not only use vishing to gain sensitive information, but also to convince the target to perform actions on their behalf, possibly by initiating the call themselves. Vishing is a powerful tool because the direct contact with the target over the phone is much more convincing than, for example, an email message, especially if the social engineer possesses an authoritative-sounding voice. In the context of AI, any voice (type) can be trained and applied within seconds [20].

Simultaneously, it allows the social engineers to hide behind the phone and hang up whenever the situation gets too hot.

We have attempted vishing in SEP04 (see [Chapter 19](#)).

2.1.3 Pretexting

As the name implies, *pretexting* involves a pretext that is used to extract information from a target. This pretext is an invented story tailored explicitly to the target. The social engineers act out the story in front of the target and thereby convince the target of the truthfulness of the scenario. After being convinced of the urgency to act, the target might provide the social engineers with information or access to infrastructure, or the target might perform other acts requested by the social engineers [8].

The better the pretext and the more elaborate the construct of lies are, the harder it gets to see through the masquerade and to stop the attackers, provided the social engineers can navigate the deep waters of lies they have created.

We have used pretexting in SED09, SED11, SED13, SED15, SED16, and SED18 (see [Part II](#)).

2.1.4 Water Holing

Water holing is a technique whereby social engineers take advantage of the trust users have in websites they visit often [8]. The technique gets its name from the metaphorical watering hole (a website) which unsuspecting deer (the targets) repeatedly visit. The nearby predators (social engineers) wait for the deer at the watering hole to which they will eventually return [21].

While attentive users might never click an unknown link in an email, let alone enter any credentials on that link, the same users might blindly trust a website they regularly visit. After having chosen a target or a group of targets, the social engineers investigate what websites the target visits regularly. Gathering information about the websites visited can be achieved in multiple ways, including traffic surveillance or an everyday conversation over lunch, if the social engineer has access to the target [8].

The most challenging part of the attack is to find vulnerabilities on the frequently visited website, so that code can be injected that will infect the target's system with specifically tailored malware. Often, the feat is achieved using zero-day vulnerabilities (see [Section 2.9](#)). If no vulnerability can be found, the social engineer is forced to either find another watering hole or to try to hack the website's server to gain full access to the website [21].

Eventually, the targets are going to revisit the watering hole, at which point the malware will infect their systems.

2.1.5 Baiting

Baiting is a technique where social engineers play with the natural curiosity or greed of their targets. The social engineers leave removable storage media devices such as USB flash drives, CDs, or MP3 players as baits in strategic places, for example, the parking lot, the restroom, or the kitchen table. These storage media containers might carry an enticing label, such as "Salaries 2018", or similar [8].

The unsuspecting but curious or greedy targets are going to pick up the device and are likely to insert it into their computer, upon which software placed on the device is going to infect the computer automatically (if compatible and not blocked) [8].

Multiple studies have been done about baiting with USB flash drives. A famous one, executed on the campus of the University of Illinois, reports that dropped USB flash drives are picked up with a probability of 98% and that users opened files on the drive in at least 45% of the cases. The study furthermore notes that the location, time of day, and day of the week did not have a significant influence on the rates with which the drives were picked up and with which the files were opened [22].

2.1.6 Tailgating / Piggybacking

Tailgating and *piggybacking* refer to the technique of entering a restricted area without authorization by taking advantage of authorized personnels' access to an area [23].

While tailgating, the social engineer follows an authorized person into the

restricted area without this person's consent or knowledge. In contrast to tailgating, piggybacking refers to the situation when the authorized person knows about the piggybacker and is escorting him or her into the restricted area, after having been convinced that the piggybacker is in fact required and in the right to access the restricted area [23].

Tailgating and piggybacking can be executed in many different settings and many different ways. It can be done stealthily or in a delivery uniform, holding a large crate and asking someone to hold the door. It can be done without a badge, or with a fake badge pretending to be authorized (and even faking the action of holding a badge to the badge scanner). It can be done alone or while hiding in a group [8].

We have used tailgating in SEP03 and SED18 (see [Chapter 18](#) and [Chapter 14](#)).

2.1.7 Dumpster Diving

In the context of social engineering, *dumpster diving* describes the act of searching trash bins and dumpsters for information such as contracts, business numbers, internal memos, emails, salaries, or passwords.

Even though companies usually go to great lengths to protect information, this protection does not always extend to trashed documents. Many employees are not aware that, once their trash leaves the company building, it is often unprotected and can be stolen by anyone.

However, companies are catching on and often offer special trash bins for sensitive information that is destroyed before it is thrown out.



See [Prevent Dumpster Diving](#) on page 148.

2.1.8 Shoulder Surfing

Shoulder surfing is the act of looking over the target's shoulder while the target is entering a password or code.

2.2 Open-Source Intelligence (OSINT)

Open-source intelligence denotes intelligence that can be gathered freely and publicly from social networks and similar sources.

2.3 Malware

The word *malware* is a composition of the two words “malicious software”. Its definition has changed multiple times during the last ten years, getting broader as different types and variants of malware became popular. Today, the Merriam-Webster Online Dictionary defines malware as “software designed to interfere with a computer’s normal functioning [24].”

There is a myriad of different malware, each one with its very own purpose. The following list contains the most prevalent types:

Trojan A trojan is a program that seemingly fulfills a legitimate purpose but secretly executes malicious code.

Worm A worm is a self-replicating piece of software whose goal it is to spread as fast and wide as possible. Upon infection of a new machine, malicious code may be executed.

Bot A bot is a piece of software that executes all sorts of attacks, such as Denial-of-Service attacks (see [Section 2.4](#)). A bot is part of a larger botnet and allows the botmaster (the owner of the botnet) to use the infected machine for malicious purposes by giving the botmaster access to every part of the machine. The bot may also be used as a new vector to spread different malware.

Rootkit A rootkit is a piece of malicious software that is activated during boot. It is already running at the time the operating system starts up and can thus effectively hide from the operating system and control it, including any antivirus that might be running on the target system.

Keylogger Keyloggers are famous pieces of software that record the keys pressed on a keyboard attached to the target machine. The recorded keys are stored in a file or directly sent to the attacker by email or by a different protocol. The recordings are valuable to attackers because they might contain passwords, credit card numbers or even more valuable information, such as private messages, health, or medical data.

Ransomware Ransomware is software that encrypts parts of a hard disk on the target machine and subsequently demands a ransom for the decryption key. Ransomware has become much more prevalent in the last years, totaling almost half a million detections in 2016 alone. The average ransom amount has gone up by a factor of more than three to \$1077 in 2016, compared to \$294 in 2015 [25].

In the context of malware, the part of the malware that executes an exploit or performs another malicious action is called the *payload*.

2.4 Denial of Service (DoS)

Denial of Service is a type of attack defending against which is extremely difficult. The goal of a DoS attack is to overwhelm a system with many requests until eventually, the system cannot serve legitimate requests anymore, often due to a lack of resources [26].

The attack is not limited to computer systems. For example, organizing a flash mob that sits down in front of a train, thereby preventing it from leaving the train station and serving the legitimate, paying customers, is a form of denial of service, too.

The only way to defend against a DoS attack is to either increase the number of resources a public system has at its disposal, or to outsource part of the load to more powerful systems. In both cases, the threat is not entirely vanquished but merely slowed down; the attackers might come back with an even more powerful attack.

Alternatively, the resource and its existence may be hidden from the attackers or access may be constrained to a limited circle of users to prevent a DoS attack from being executed in the first place.

Shutting down the attackers is generally hard, because a DoS attack is often decentralized (e.g., many people are participating in blocking the train in the example above), which makes it hard to pinpoint a single person or group responsible, and even harder to shut them and their botnet down.

2.5 Man-in-the-Middle Attack (MitM)

In a *Man-in-the-Middle* attack, an attacker is secretly relaying messages between two parties who think that they are directly communicating with each other. In the process, the attacker may read and possibly alter the messages [27].

The attack can be prevented if the protocol in use demands mutual endpoint authentication, such that one endpoint can ensure that the message came from the expected endpoint. Alternatively, hash functions can be used to allow tamper detection, such that one endpoint can detect if a message has been tampered with in transit [27].

2.6 Look-Alike Domain

A *look-alike domain* is a domain that resembles a well-known domain. For example, gooogle.com is a look-alike domain of google.com¹.

¹gooogle.com redirects to google.com. Google has bought gooogle.com to prevent misuse of the domain as a look-alike domain.

Such domains are often used by attackers to trick targets into clicking and trusting a link. Because the domains look the same at first glance, the target is usually not suspicious and enters credentials on the website.

Attackers also register look-alike domains in the hope that someone will accidentally type, for example, faxebook.com (the “x” is right next to the “c” on most keyboards). The attackers then direct faxebook.com to a perfect clone of the Facebook login page and hope that users will not recognize that they are on the wrong domain. When the users log in on this fake login page, the attackers capture their credentials.

2.7 Metasploit Framework

The open-source *Metasploit Framework* is a penetration testing framework. It offers a large number of tools with which one can gather information about a target and subsequently exploit the vulnerabilities found [28][29][30].

It is not the only framework of its kind, but one of the most prevalently used ones because of the ease with which it makes its tools available [30].

Metasploit can be downloaded for free on its website. It runs on Windows and Linux and offers a large, open source community where everybody can contribute [28].

2.7.1 Meterpreter

Offensive Security describes the *Meterpreter*, a combination of the words “Metasploit” and “interpreter”, as follows [31]:

“ Meterpreter is an advanced, dynamically extensible payload that uses *in-memory* DLL injection stagers and is extended over the network at runtime. It communicates over the stager socket and provides a comprehensive client-side Ruby API. It features command history, tab completion, channels, and more. ”

A stager is a small piece of software that opens a connection between the attacker and the target [29]. In the case of the Meterpreter, this stager is loaded directly into memory upon execution by the unsuspecting target. Consequently, the stager never touches the disk and thus diminishes the chances of being discovered by an antivirus software. After the stager has opened the connection, the attacker can issue commands and load further payloads (stages) via an API running on the attacker’s machine. [31].

Meterpreter gives the attacker extensive access to the target computer. Files can be downloaded, edited, replaced, or planted. Keyloggers can be run, pictures can be taken, and conversations can be recorded using the cameras and microphones that are installed [32]. The Meterpreter is a powerful tool, where the only challenging part is to convince the target to execute the stager.

2.8 Shell

A *shell* is a command-line interface for access to the services of an operating system.

There are two types of shells [33]:

Bind Shell A *bind shell* is a type of shell in which a *server* opens a port for incoming connections. The client can then connect to this port on the server and issue commands that the server is supposed to execute. This type of shell is most often employed to connect remotely via SSH.

Reverse Shell A *reverse shell* is a type of shell in which a *client* opens a port for incoming connections. The server can then connect to this port on the client and allow the server to issue commands.

The latter is appealing to an attacker. Firewalls often block incoming connection attempts, but might be more liberal about outgoing connections. If the attacker achieves to install a piece of code on the target machine that opens a connection to a server of the attacker's choice, the attacker can get a reverse shell anytime the target computer is online.

The Meterpreter (see [Section 2.7.1](#)) is an example of a reverse shell.

2.9 Zero-Day Vulnerability

In the context of computer software, a *vulnerability* is a weakness in the code that allows attackers to perform unauthorized actions [34], such as gaining access to a system without credentials or executing arbitrary code on the target's system upon opening a website.

A *zero-day vulnerability* is a vulnerability that is unknown to the parties interested in patching the vulnerability. Until such a patch is released, attackers can use the vulnerability at their discretion. Day zero is the day on which the party responsible for patching the vulnerability learns of its existence. The time between discovery of the vulnerability and its zero-day may be arbitrarily long. Even after the vulnerability has become common knowledge, it can take quite a while until a patch is available, depending on the vendor and platform [35].

2.10 Regular Expression

A *regular expression* is a string that defines a search pattern. With the use of regular expressions, different strings that are similar in nature (such as email addresses or phone numbers) can be found quickly in large text documents.

Example

The regular expression `/([a-zA-Z]+)/g` utilized on the following text

```
Elephant: grey  
Crocodile: green  
Fox: orange
```

will find the words `Elephant`, `Crocodile`, and `Fox`.

- `/.../` are delimiters, defining the beginning and end of the regular expression.
- `(...)` defines a capturing group. A regular expression can consist of multiple groups whose contents can later be extracted.
- `[a-zA-Z]` matches any character of the alphabet, both lower case, and upper case.
- `+` is a quantifier, saying at least one.
- `:` matches the character `:` literally.
- `g` is a flag. It stands for *global* and tells the regular expression not to stop after one occurrence has been found.

Regular expressions quickly become complicated and hard to read as the pattern grows more complex. However, for many tasks, they are instrumental.

2.11 Antivirus Software

Antivirus software is the most widely used malware countermeasure (often offering prevention, detection, and removal of malicious software). It is employed by almost every computer in some form or another. This section gives an overview of how current antivirus software works.

2.11.1 Signature-based Detection

Every piece of software has a unique fingerprint, a so-called *signature*, which is derived from the binary file that makes up the software. Known malware can be analyzed in an automated fashion to extract a signature. This analysis takes time [36].

Antivirus software comes with a massive database filled with signatures of known malware, a sort of blacklist. As soon as a new file is written to the hard disk or a periphery (such as a USB drive) of your computer, the antivirus software scans the file and compares its fingerprint to the fingerprints in its database. If they match, the file is assumed to be malware, and the antivirus software reacts accordingly by preventing or stopping the execution of the file, and, depending on the settings and the antivirus, deleting it immediately [36].

The advantage of signature-based detection is the low false positive and the high true positive rate, meaning that most files flagged as malware *are* actual malware and that few false alarms are triggered (a false alarm means flagging a benign file as malicious) [36].

The disadvantages of this method, though, are daunting. New malware is being developed at an alarming rate. The Symantec Security Threat Report states that in the year 2016 alone 357 million new malware variants were identified by their antivirus software, and this number is increasing every year [1]. Because old malware might still be around, no signature that made it into the database can ever be allowed to leave it again. As a consequence, the database keeps growing. Additionally, a single signature can only cope with limited polymorphism, meaning that a malware executable that does the same action as another malware executable might have an entirely different signature, due to various methods employed by the malware creators (see [Section 2.11.3](#) for a list of detection evasion techniques). Maintaining such an extensive database consumes a lot of storage space, memory, and CPU cycles. Signature-based detection can only react to known malware and is thus called **reactive** [36].

To combat these disadvantages, some antivirus software vendors offer so-called **heuristic** signatures, which identify a certain program-logic or action an executable tries to perform [37]. Consequently, signature-based detection is enabled to flag unknown threats proactively.

2.11.2 Behavior-based Detection

Another approach to detecting malware is the behavior-based approach. This approach requires a so-called ground truth or baseline defining the “normal” behavior of software. This ground truth assumes that legitimate software will probably not access system files, edit the registry, or use the network in an unusual manner (which has to be defined as well) [36].

The advantage of behavior-based detection is that it will not only catch known threats but also emerging new ones, as long as they categorize as anomalous or unusual. The behavior-based approach is thus called **proactive**. This approach makes it harder for malware creators to hide their intention and to evade the antivirus [36].

The disadvantage, on the other hand, is that the ground truth is hard to define and, as a consequence, the false positive rate (categorizing a file as unusual and thus malware, even though it is not) is quite high. High false positive rates are irritating to the user which is why antivirus software frequently prefer to remain silent, even though it has discovered a potential threat [36].

2.11.3 Antivirus Detection Evasion Techniques

Malware creators work hard to evade detection by antivirus software. The following list features some of the techniques that can be employed to make a piece of software undetectable [38]:

Polymorphism By adding randomness of some sort to the program code, by adding code blocks that have no effect at random places, or by changing the code execution order one can effectively produce executables with different signatures. The process of changing the source code according to the suggestions mentioned before can easily be automated, making it possible to produce variants on a massive scale in seconds, making signature-based detection inefficient and almost useless. The compiler, its version, and its settings do also affect the resulting binaries. By changing these parameters alone, one can create many variants without breaking functionality. The choice of compiler and the change of the settings can be automated as well.

Code Obfuscation Binaries are hard to read, but there are programs, so-called decompilers, that try to recreate the source code from a binary. Even though the result is stripped of useful comments and any meaningful variable and method names, eventually one can still figure out what the binary actually does (and as a consequence maybe find the server it connects to and thus the criminals running it). To protect against this sort of analysis, one can obfuscate the code by using unusual methods to achieve a behavior. Obfuscation can be automated as well, and there are contests around this “art”.

Encryption To make it impossible for signature-based antivirus software to detect a malign executable, the malware creators can encrypt the actual payload. The encrypted payload is then added to a so-called stub which, when executed, brute forces the encryption key (which is easy enough so that it will not take too long), decrypts and executes the payload.

If the signature of the stub is known to the antivirus software, encryption is of no use, and a new stub has to be written.

Protection Antivirus vendors might try to open a file in a virtual machine to analyze an executable and find out its intentions. Opening the executable in a virtual machine (a so-called sandbox) limits the possible harm a malware can do. To protect themselves, criminals may add anti-debugging features to their executables, preventing successful analysis in a sandbox by not executing the payload if a VM or a debugger is detected.

Binding An excellent method of evading an antivirus is to embed the payload in another application, for example, Notepad. An antivirus may not detect the resulting Trojan because it looks like a legitimate program. It may also facilitate the payload delivery because people trust specific programs blindly.

These methods can be combined and ordered arbitrarily.

2.12 Intrusion Detection System (IDS)

Intrusion detection systems monitor the traffic to a computer and try to identify malicious software before it reaches the target computer.

Just as malware detection in antivirus software (see [Section 2.11](#)), intrusion detection can be either *signature-based* or *behavior-based*. While signature-based detection results in higher precision (fewer false positives), it is incapable of detecting unknown attacks. Behavior-based detection, on the other hand, can detect unknown attacks but suffers from lower precision and the intricacies of establishing a ground truth (as mentioned in [Section 2.11.2](#)) [39].

In a *Host-based IDS (HIDS)*, the IDS usually is a device monitoring a single host, or software that runs on a single host. A HIDS requires one sensor per host and is potentially cumbersome to deploy if many hosts need to be protected [39].

In a *Network-based IDS (NIDS)*, the IDS monitors entire network segments, simultaneously protecting multiple hosts on that segment. Since only a single NIDS needs to be installed to protect many hosts, deployment is more straightforward. However, the NIDS might run into performance issues if the hosts are producing much traffic [39].

While these are the two main types of IDS, combinations thereof are conceivable.

In combination with Firewalls, Intrusion Detection Systems build the first line of defense against malware and intruders [39].

2.13 Greylisting

In the context of email, the usual strategies to prevent spam involve blocking unwanted senders and emails. Blocking can be done using *blacklisting* and *whitelisting*.

While a blacklist is a list of all the senders who should be blocked, a whitelist, on the contrary, lists only the senders who are allowed to send emails to an address and blocks everyone else automatically. A blacklist is extremely hard to maintain because new perpetrators need to be added continuously, and legitimate senders might be added accidentally. Whitelists, on the other hand, do not have their issues in maintainability, but in flexibility. Since only senders who are on the list can communicate with the email account employing the whitelist, the system becomes unable to receive messages from unknown senders.

A third option to combat spam is called *greylisting*. When employing greylisting, an email server may put emails into a greylist upon receiving them, thus temporarily rejecting the emails. The server counts on the fact that legitimate emails will be resent after a delay, while spam emails are only sent once, for reasons of saving resources. After the server has accepted an email upon its second arrival, the sender of the email may be put on a temporary whitelist [40].

Even though greylisting may work against regular spammers, its effectivity in the context of social engineering is questionable, since there are two significant disadvantages:

- Greylisting is useless against targeted attacks because every spammer can set up an email server that correctly queues and resends spam emails (see SED16 in [Chapter 13](#)).
- Greylisting renders the instant email protocol into a protocol with uncertain delays, depending on how long the sending server takes to resend the emails. The delays irritate users and relieve the email protocol of its instantaneity, one of its most significant advantages [40][41].
- Greylisting may lose legitimate emails in the process which can result in great irritation and damage [41].

2.14 Sender Policy Framework (SPF) And Its Successors

Sender Policy Framework (SPF) is a protocol that allows the owner of a domain (such as [yuublu.com](#)) to specify which servers are allowed to send emails on behalf of that domain. The IP addresses of these servers are published as an *SPF*

record in the domain name system (DNS)² so that email receivers can check the authenticity of an email, namely if a sending server is authorized to send an email for a particular domain [42]. SPF does not provide cryptographic security and merely allows to match hostnames and IP addresses [43].

Example

If the owner of the domain `abc.com` publishes an SPF record stating that only `server1.abc.com` is allowed to send emails on behalf of `abc.com` (meaning emails with sender address `anything@abc.com`), then every email server receiving an email from an address in the domain `abc.com` can check, if the IP address of the sending server matches the IP address of `server1.abc.com`, which can be read from the DNS entry.

If not, then `server1.abc.com` was not the sender and the email should possibly be ignored, because anyone could have sent the message.

DomainKeys Identified Mail (DKIM)

The SPF protocol allows matching the IP address of the sending email server with all the IP addresses that are allowed to send emails on behalf of a particular domain. Unfortunately, it is possible to spoof IP addresses. Unforgeable cryptographic guarantees are necessary that allow unambiguous authenticity checks.

DomainKeys Identified Mail (DKIM) provides these cryptographic guarantees by signing every outgoing email with the sending domain's private key. A corresponding public key is provided via the domain name system (DNS). DKIM allows the verification of both the authenticity, as well as the integrity of the signed parts of the email [43].

Since DKIM is more complicated than SPF, not all email servers have implemented the protocol. Thus, the absence of a DKIM signature does not automatically make an email fraudulent [44]. To increase the chances of catching fraudulent email despite its low adoption, DKIM should always be used in concert with SPF [43].

Domain-Based Message Authentication, Reporting, and Conformance (DMARC)

Domain-Based Message Authentication, Reporting, and Conformance (DMARC) is based on both SPF and DKIM. If both of these protocols are set up for a domain, DMARC allows monitoring the domain for validation errors of the two protocols [43]. Every time an email fails both the SPF and the DKIM validation,

²Used for turning a domain name such as `yuublu.com` into an IP address.

the validating email server applies a policy, such as quarantine or rejection, provided by the domain owner (of the domain whence the email allegedly came from) to the fraudulent email and additionally sends a report to the domain owner. Both the domain owner's policy regarding fraudulent emails and the destination address of the reports are published via the domain name system (DNS) as part of the DMARC protocol [45].

Additionally, DMARC prevents spoofing of the **From:** header, which is still possible under DKIM and SPF [45].

Preparation

While every attack described in [Part II](#) features a preparation section, this chapter describes efforts that were necessary repeatedly (see [Creating Fake Email Accounts](#) in [Section 3.2](#)) or took substantial amounts of time to complete (see [Virtual Private Server](#) in [Section 3.1](#), [Creating a Payload](#) in [Section 3.3](#), and [Creating a Malicious Word Document](#) in [Section 3.4](#)).

Even though the preparatory work described in [Sections 3.3](#) and [3.4](#) did not find application in any schemes (see [Unexecuted Attacks](#) in [Section 5.1](#)), we still mention the details due to the time spent on the matter.

3.1 Virtual Private Server

To execute attacks on people or their computers, it is often necessary to deliver a payload (see [Section 2.3](#) about [Malware](#)), or to have a web server running through which people can be tricked into logging in (and thereby surrender their login credentials) or into downloading malicious files. To preserve the attacker's anonymity, incriminating files are, if possible, not hosted on the attacker's personal server. Buying or renting a virtual private server (VPS) can solve this problem.

A VPS is a virtual machine that can be bought or rented as a service. On this server, an operating system is running which is solely used by a single customer (making the server private) and which often can be chosen from an array of operating systems during checkout. The server comes with a dedicated IP address¹, such that it can be reached from all over the world. Depending on the hosting service and on the amount of money one is willing to pay, the server comes equipped with more RAM, CPU cores, and storage space. Often, the package includes a domain name.

The customer can log in to his or her VPS via SSH and has then full access

¹A dedicated IP address is a unique IP address that is exclusively used by a single host, in contrast to many hosts sharing a single IP address behind a NAT

to all of the server's resources via a command line interface. The customer can then run any program whatsoever on the server, such as a web server, a cloud service, or merely an offline service that makes some computations.

The service we subscribed to in this experiment was based in Germany. It came with two dedicated IP addresses and was running Ubuntu 14.04 LTS. Additional server specifications are listed in Table 3.1 on the following page. It cost 10\$ per month.

In order to stay as anonymous as possible, we registered the server in the name of the CEO of `<YuuBlue>`, `<Julius Caesar>`, who was also holding `<YuuBlue>`'s primary domain. Because all bills and invoices would be sent via email, we used the real address of the company as a contact address, gambling on the fact that no messages would ever be delivered by the postal service. We used an email account (`<julius.caesar>@protonmail.com`) created explicitly for this purpose to communicate with the VPS provider (see Section 3.2 about email account creation). Registering business domains with personal email addresses is not unusual because a business email address may not yet exist at the time of registering the business domain name.

The only data that could expose the fraudulent use of the CEO's name for registering the server was the billing account which was in the author's real name. Having the account in one's real name might endanger anonymity. If anonymity is of importance, a hosting service that accepts BitCoin or another anonymous payment method may be better suited. On the other hand, the hosting service seemed trustworthy enough to not simply give out any customer information without being severely pressured. They want to keep their customers and make money, after all.

As mentioned above, the VPS plan came with a domain name included. The domain we chose was `yuubblue-support.com` (in the original domain, "company" was the name of `<YuuBlue>`). The domain name was registered using the same name with which the VPS was bought, such that anyone trying to find out who registered the domain would find that the same person had registered it who had also registered the company's primary domain `yuubblue.com` all these years ago (the spelling mistake in the name of the CEO that was made when registering the main domain `yuubblue.com` was copied as well, although it was only later that we discovered this fact). The purpose of choosing `-support.com` was to make the domain look as legit as possible.

On our VPS, a web server was running on port 80 (HTTP) and 443 (HTTPS) to host any payloads that might need to be delivered and any phishing websites or other websites used to deceive a target or execute a scheme. Additionally, we manually installed Metasploit (see Section 2.7) to take advantage of its readily available exploits.

To free port 22 (standard SSH port) for connecting backdoors (see Sec-

tion 3.3.2), we moved the SSH service from port 22 to port 23.

Service	Used
RAM	4 GB
CPU cores	2
Hard disk	70 GB
Traffic	unlimited

Table 3.1: Virtual Private Server Specifications

3.2 Creating Fake Email Accounts

Creating fake email accounts is not quite as easy as anticipated. Although the process itself is simple, large providers such as Gmail, Yahoo, and Microsoft Outlook try to prevent the creation of fake accounts by limiting the number of email addresses that can be registered by single IP addresses and phone numbers.

Naturally, attempts can be made to circumvent these measures by using a different IP address for every email address one might want to create. Many VPNs and proxies are offering this exact service to everyone. However, because everyone can use these IP addresses, the addresses are usually blocked from the beginning, because somebody has already used them to create an email address. The use of home or work IP addresses needs to be avoided if an attacker would like to stay anonymous. With VPN, proxy, home, and work IP addresses off the table, IP addresses have to be found elsewhere.

To find virgin IP addresses, public networks are prime targets. Using IP addresses of public networks, fake email accounts could be created in theory. The first obstacle, however, is that most public networks require their users to provide their phone numbers to log in. Providing the phone number immediately eliminates privacy, depending on how much is logged on the network. Even if a less paranoid stance is assumed, where the network provider is not logging all activities or where the network provider is not even asking for a phone number, the following problem still emerges: virgin IP addresses are rare. During a testing phase at the beginning of this experiment, we were unable to create a fake account with two of three public networks.

The email providers offer a way to prove one's authenticity via short messages with codes. In that case, the IP address used does not matter anymore.

However, using a private phone number is not possible if the goal is to stay anonymous and even if anonymity is not imperative, after a certain amount of accounts the phone number will allegedly be blocked as well². Consequently, more phone numbers are required. Although there are countries where a SIM card can be bought anonymously and relatively cheaply, Switzerland is not one of them [46]. To prevent criminals from communicating anonymously, Switzerland requires customers to provide identification and to register when buying a SIM card, which eliminates privacy. Despite these measures though, it is still possible to get anonymous SIM cards, because once bought, the SIM cards can freely change owner [46].

Although other countries might handle the issuing of SIM cards differently, Switzerland will probably not be the last country to tighten its control over SIM cards.

Since buying SIM cards anonymously is cumbersome, “Phone Numbers as a Service (PNaS)” become more interesting. There are services offering the use of phone numbers to protect one’s privacy. Most of them charge a monthly fee while a few are free of charge. During the experiment, we tested some of the free services in the attempt to create email accounts. Unfortunately, none of the phone numbers that we tested worked. More often than not, the numbers were not even accepted by the email providers, indicating that there exists a blacklist. While paid services might work better, potential logging might be a reason to avoid them because the payment information could link back to the attackers.

The use of smaller email providers (like Protonmail³) might fix the IP and phone number problems because no anti-fraud measures are in place. However, at the same time, these smaller providers lack the inherent trust a recipient has towards a sender that is using a reputable email provider. This trust may be a result of the above discussed anti-fraud measures.

In the end, it might very well be possible to create fake email accounts while staying anonymous, but it is not possible to do it on a large scale. The email providers successfully enforce the need to go to great lengths (buying SIM cards anonymously, visiting different public networks, testing a lot of PNaS numbers) to create a single account. The time expenditure is immense. However, it seems like none of the email providers enforce a limit on the number of accounts that can be created with a single phone number (or we did not reach this limit). Therefore, a single anonymous phone number might be enough to create many accounts.

Summed up, we created three accounts anonymously. Thereafter, we used a single private phone number to create more than ten other accounts with Gmail,

²In our tests, this claim did not hold true.

³Protonmail lets users create accounts without a phone number by donating a freely choosable amount of money via credit card. Provided that the credit card does not link back to the attacker in any way, Protonmail basically lets people *buy* fake email accounts.

Yahoo, and Microsoft Outlook, without getting blocked.

3.3 Creating a Payload

Metasploit offers an array of potent tools, exploits, backdoors and other malicious software. One of them is the Metasploit Meterpreter (see [Section 2.7.1](#)).

In essence, a reverse shell (see [Section 2.8](#)) is opened as soon as the target executes the Meterpreter payload. Once the reverse shell is established, the attacker has complete control over the target machine. Due to Metasploit and similar tools, it has become uncomplicated to craft powerful payloads such as the Meterpreter in seconds. Tools like Metasploit enable aspiring hackers to maximize their impact while minimizing the time invested.

The Metasploit exploits are thus relatively often used and, as a consequence, well-known to antivirus solutions. This popularity proved to be a significant problem.

3.3.1 Evading Antivirus Software

Due to the ubiquity of Metasploit, antivirus vendors have studied its exploits well and have added many of them to their signature database (see [Signature-based Detection](#) in [Section 2.11.1](#)). This good antivirus coverage makes some Metasploit payloads (in their pure, unadulterated form) less useful in real attacks (of course they serve as excellent examples for educational purposes), because as soon as the target downloads such a payload, the antivirus is going to prevent execution, followed by immediate termination of the threat (depending on the antivirus settings).

To evade the antivirus software, we used mutation via random padding and simple code obfuscation in the case of Word documents (see [Antivirus Detection Evasion Techniques](#) in [Section 2.11.3](#)).

Because the payload would only load a so-called *stager* (see [Section 2.7.1](#)) into memory, additional capabilities need to be loaded at a later time. These additional stages were, just as the initial stager, quite famous and just as easily detected by IDS (see [Section 2.12](#)) and antivirus software. While during the creation of the payload with `msfvenom` (see [Section 3.3.2](#)) an encoder was used to avoid bad characters that would stop the payload from executing, an encoder can also be used to introduce entropy into bytecode. Exploiting this fact, Metasploit offers the option to enable stage encoding for the Meterpreter's second stage, which allows preventing the stage from being detected.

Despite of all these measures, we were not able to create an undetectable payload. Due to a lack of time, we had to give up on this kind of attack eventually.

3.3.2 Custom Payload

After unsuccessfully trying to use plain Metasploit payloads, we wrote a custom executable in C, in which we hid the actual payload. On the one hand, this made the payload considerably larger and somewhat harder to deliver than the compact Metasploit payload. On the other hand, it created the opportunity to not only execute the Meterpreter reverse TCP payload but to also visit a statistics page in case the antivirus would analyze the behavior of the Meterpreter payload and shut it down before the reverse shell could be opened (see [Behavior-based Detection](#) in [Section 2.11.2](#)). The statistics page was hosted on our web server (see [Section 3.1](#)) and used PHP to store the IP addresses of machines executing our payload. By letting the payload issuing GET requests with different parameters, differentiation between different payloads was possible so that the success of individual attacks could be assessed.

Because companies tend to employ firewalls that block not only incoming but also outgoing traffic, the payload connected to the statistics page on port 80 and the Meterpreter payload was configured to connect to port 22. These two ports correspond to HTTP (80) and SSH (22). These are standard ports that are usually not blocked by companies because blocking them would impede their employees in doing their work (for example, if port 80 is blocked, many websites cannot be accessed).

We used MinGW to compile the payload for Windows. The resulting executable could be placed on a USB stick or hosted on a web server, ready to be downloaded and executed.

To throw the antivirus off-track, we wrote a small Python script that generated random padding, slightly obfuscated the code, and generated a Metasploit payload (using `msfvenom`) every time it was executed. Every parameter was fully customizable so that large amounts of different and individual payloads could be created in a short amount of time.

While we were able to evade Microsoft's Windows Defender, we could not fool Symantec's antivirus and remained unable to do so, due to a lack of time and experience in malware development.

3.4 Creating a Malicious Word Document

Word offers multiple ways to execute code:

Macro Macros are well-known and are written in Visual Basic for Applications (VBA). VBA has access to most Windows system calls and can be automatically executed when an Office document is opened (depending on the settings, the user needs first to enable the content). This broad ac-

cess makes it simple to write malicious code, which led to the creation of so-called Macro viruses.

Object Linking and Embedding (OLE) OLE allows to link files and executable scripts (VBA and JavaScript) in a Word document. If the user agrees to execute the script, malicious code can be executed. Usually, the script downloads further payloads and executes them.

Dynamic Data Exchange (DDE) The DDE protocol enables different processes on Windows to communicate with each other. It enables Word to get data from another process and load it into the document, for example. Of course, any data can be loaded, including malicious payloads. In fact, the DDE protocol was abused so much that Microsoft decided to “disable the Dynamic Data Exchange protocol (DDE) in all supported editions of Microsoft Word [47]”.

In the experiment, we wanted to use macros, because Metasploit’s `msfvenom` offers a macro out of the box. OLE was found to be too well protected in newer versions of MS Office and DDE did not work anymore (because of patches applied to newer versions of the Office suite).

Unfortunately, the antivirus flagged the Word document produced by `msfvenom` as malicious. To circumvent the antivirus, we had to obfuscate the VBA code heavily. Fortunately, there are tools for VBA code obfuscation available on Github. After the code obfuscation, the antivirus stopped flagging our Word document but kept preventing the execution of the infamous Metasploit payload.

The way the payload is executed is that the VBA code looks for the binary payload data in the Word document itself. This payload data is a long string of numbers in a hexadecimal format which was pasted into the document, just as regular text would be placed in a document. It would then write this binary string to a file and execute this file. With this method, any code can be executed by pasting the executable in its hexadecimal representation into the document. By using the custom payload (see [Section 3.3.2](#)), a Word document was created that executed the payload without being stopped by the antivirus. The resulting Word document contained a string that was 30 pages long and lagged considerably when viewed. However, this did not matter, since the payload could now be executed without any problems, providing access to the machine via the Meterpreter reverse shell.

Unfortunately, further testing showed that many antivirus solutions still detected the payload, which is why we had to forego Word documents in our attacks.

Sources of Information

This chapter takes a look at the sources we used to gain the necessary information in order to be able to execute the schemes and attacks described in [Part II](#).

If this chapter seems to be rather short, it may be because it is. Starting with zero internal knowledge about [YuuBlue](#), we acquired information step by step, as described in the following sections. As it turned out, most of the attacks did, surprisingly, not require much information, but instead creativity and boldness.

Any external actor is capable of acquiring an impressive level of knowledge about a target by patiently asking questions, by reading documents, websites, and other sources, and by looking in the less apparent and, more importantly, in the apparent places.

4.1 Email Addresses

Using the hereinafter described techniques, an attacker is able to extract large numbers of email addresses with zero prior knowledge.

The same techniques were used on [YuuBlue](#) to gather some 500 distinct email addresses, which is a substantial percentage of the total number of employees of the company. Depending on the size of the company and their level of security awareness, this number may be considerably higher or lower.

4.1.1 Via the Company Website

On company websites, oftentimes a lot of employee names and corresponding email addresses can be found. This is inevitable if a company wants to present contact information online that is easily accessible to its customers.

The downside of displaying email addresses on the company website is, that it gives an attacker a lot of potential entry points, as every employee could become the target of a social-engineering attack.

What is even worse is that after manually reviewing only a handful of the addresses found online, the pattern the company uses to create email addresses for its employees becomes apparent. In the case of the target company of this thesis, the pattern was

`firstname.lastname@yuublue.com`

Internationally acting companies employ people from various countries with different kinds of names, including special characters and last names consisting of multiple words. After analyzing the email addresses for some special cases (found on the company website), the following rules crystallized:

- Characters with diaeresis (for example ö, ä, ü) are appended an *e* (oe, ae, ue).
- Characters with diacritical marks like acute accent (é), grave accent (è), or a circumflex (â) are stripped of the diacritical marks.
- Last names consisting of multiple words are stripped of any spaces. Hyphens are left intact.
- Any middle names are omitted.

The email address pattern combined with the above rules forms a powerful instrument. It allows the attacker to guess the email address of any employee whose name the attacker knows, from the intern all the way up to the C-level executives.

The attacker is now in the position to gather a lot of email addresses from the company website by visiting every page and writing down the names and email addresses of all the employees the company discloses on their website.

For smaller companies, this can easily be done manually. For larger companies, like [YuuBlue](#), the attacker might write a script to speed up the process (see [Appendix A](#)). In the case present, the script employed regular expressions (see [Section 2.10](#)) to find any email addresses and links leading to different sub-pages of the company website. It then stored the email addresses and put the links into a queue.

All pages in the queue are called subsequently and the same process was repeated time and again until the script terminated after about 24 hours, after having crawled more than 8000 pages.

As it turned out, [YuuBlue](#) employed client-side decryption of email addresses. This involves a simple, char-by-char encryption of the email address. The encrypted address is then put into the source code and decrypted upon pageload, resulting in source code like this:

```
<a href="znvygb-grfgMznvy;pbz">grfgMznvy;pbz</a>
```

which translates to

```
<a href="mailto:test@mail.com">test@mail.com</a>
```

when properly decrypted.

As mentioned in [Section 4.1.1 Mitigation](#), such mechanisms do not work against targeted attacks because the attacker can simply copy the decryption algorithm after having analyzed what it does.

The script yielded 305 distinct email addresses, all from the [YuuBlue](#) website. In theory, the script could be adapted to also look for names, although this is harder to do because it is difficult to distinguish normal text from a name. Therefore, the script would probably result in a lot of false positives (text that looks like a name but really is not). [Section 4.1.2](#) presents a better option to turn names into email addresses.

Mitigation

Defending against email address crawling is not an easy task. The only real defense is to not display any email addresses on the website or to limit the amount of disclosed addresses to an absolute minimum.

Contact forms can be used in place of directly displayed email addresses. Contact forms allow transmitting data to the server, which then sends a message in the name of the user. This essentially prevents the disclosure of email addresses on the website. However, the users do not really know whom they contact when using a contact form and might be displeased about this fact.

If these restrictions are not an option, there are several possibilities to display email addresses more securely, each with its own, sometimes considerable, downsides [48]:

Replace Email Addresses through Images

Effective against *simple* spambots. However, screen readers cannot read images. Images do not scale the same as text, which can lead to ugly results. Users have to retype the email address if they want to use it. Furthermore, artificial intelligence came a long way in recent years, allowing better and better character recognition on images.

Employ CAPTCHAs

Only display the email address, if the user can solve a CAPTCHA. The downside is that solving CAPTCHAs might upset the users.

Replace Email Addresses through Redirect Pages Spambots think that they encountered a regular link. When users click on the link, they are

redirected to the correct email address. The downside is, that a redirect is necessary, which costs resources and time.

Disguise Email Addresses

Using special characters, HTML entities, URL encoding, or HTML comments, an email address can be disguised, so that the address is still correctly interpreted by the browser, but looks different in the source code. Disguising email addresses is not sufficient to prevent crawling of the website, because each email link still contains the `mailto:` identifier. Spambots are usually intelligent enough to get past the above-mentioned disguises.

Client-Side Email Address Composition or Decryption

Via client-side email address composition/decryption, the browser uses a recipe defined in JavaScript to reconstruct the email address directly in the browser from a seemingly random string. Done right, this can be effective against spambots because the email address will not appear plainly in the source code. Most spambots do not run JavaScript because they do not know if and when it will terminate, which would cost them too much computation time¹. The downside is that users who have JavaScript disabled will, just as the spambots, not see the email addresses.

The last two options have no effect whatsoever against targeted attacks in which the attacker analyzes the website and what it does to its email addresses. The other options come with considerable inconveniences to the users.

Which option to use (if any) is subject to individual risk assessment. There is always a trade-off between convenience and security.

4.1.2 Via LinkedIn

Once the attacker knows the email address pattern of the targeted institution (see [Section 4.1.1](#)), LinkedIn is a great source for more names and information. LinkedIn lets you search for all the people whose current employer is a certain company. It even lets you specify the location at which they work. This allows for accurate and targeted crawling of names and thus email addresses.

To be fair though, LinkedIn does a lot to prevent such malicious behavior:

- LinkedIn users can only see people who are no more than three links away from themselves. However, it is usually enough to befriend a single employee of the target company to get access to the entire company. This employee might be chosen strategically, for example because this person is especially sociable and not likely to turn a friend request down.

¹It is cheaper to go for the lower hanging fruits.

- LinkedIn's user agreement states that no user is allowed to crawl user accounts or to enumerate companies [49]. If too many page visits are registered for an account, the account gets restricted². After restriction, a proof of identity is necessary to unrestrict the account. Although it makes enumeration much more cumbersome by requiring the attacker to create an account for every enumeration, the attacker's account is likely fake anyway, limiting the effectiveness of this measure.

Again, the attacker can employ a script to crawl all the necessary information. In the present case, LinkedIn provided some 386 employees, resulting in 386 email addresses and job descriptions.

4.2 Subdomains via DNS Enumeration

Usually, a company has multiple subdomains registered for its domain. These subdomains often serve employees or customers as easy access points to internal services and usually contain valuable information. If the subdomains are not directly linked to from the company website, tools can be used to enumerate a domain.

This is called DNS enumeration. With the help of a so-called dictionary file containing words that are usually found in front of a domain (such as `admin.yuubluе.com` or `login.yuubluе.com`), a DNS enumeration tool enumerates a domain by taking one such word after another and by checking if the DNS returns an IP address for a specific subdomain.

The dictionary file used against `<YuuBlue>` contained 100 000 words and took about four hours to complete, yielding a list of 16 valid subdomains.

4.3 Availability via Out-of-Office Replies

Availability in this context describes whether or not an employee is in the office or not.

The findings of SED04 and SED12 (see [Chapter 6](#) and [Chapter 10](#)) turned out to be major sources of information. The two schemes namely provided us with automatic out-of-office replies that specified the availability of specific people, among them C-level executives.

Knowing from when until when an employee is going to be out-of-office (sometimes the replies even specified the reason, like vacation, child birth, or sickness) opens the door for an attacker to exploit other employees lower in the hierarchy

²LinkedIn blocked our account a day after we had crawled all the `<YuuBlue>` employees. Even though it was too late, LinkedIn seems to be monitoring user behavior strictly.

(or, if done right, even higher ups). Knowledge of availability allows to execute schemes like SED09, SED11, SED13, SED15, and SEP01 (see [Part II](#)).

Besides availability, the automatic out-of-office replies provided us with employee names, project names, telephone numbers, further email addresses, relationships between employees (for example, in the automatic out-of-office reply, an employee might refer to a person as his or her boss, colleague, or deputy whom one should contact in urgent cases, because the employee is not in the office), work quotas, and in some cases even the employment status of a person (there were out-of-office replies for people who no longer worked for [YuuBlue](#)).

4.4 Names and Maps via YouTube

[YuuBlue](#) maintained an active YouTube channel featuring their latest products, events, and talks.

In the videos, freely available on YouTube, employees answered questions about the development process and talked about current topics. From these videos, we could harvest names, job positions, and email addresses. Even though the process is much slower than gathering names and email addresses on the [YuuBlue](#) website or LinkedIn (see [Section 4.1](#)), we learned a lot about [YuuBlue](#) and its values and culture by watching their videos.

Of particular interest was one video that featured a group of developers proudly flying their camera-mounted drone through the [YuuBlue](#) main building. From this video, we gained the knowledge to draw a rough layout of the rooms and areas inside the main building. Knowing the layout of a building can be of significant help when tailgating and trying to orientate oneself in a new environment.

4.5 Organigram via the Company Website

Aside from email addresses and names, a company website usually features much more information. From current projects and customer names to partners, company values and goals, and public sales numbers all the way to an organigram of the management, if one gets lucky.

The organigram on the [YuuBlue](#) website, complete with photos of the management, was of great help when we were getting acquainted with the company. Later, after we had received our first out-of-office replies (see SED04 in [Chapter 6](#)), the website and the data we crawled from LinkedIn (see [Section 4.1.2](#)) served as a work of reference in which we could read up on the positions of specific employees.

4.6 Information via Social Media Networks

Social networks are rich sources of information. On social networks, employees talk about their companies relatively freely. Once an employee is befriended, almost any information about a company can potentially be gained by just asking the employee in a well-crafted conversation online.

However, we were also able to gain information by simply reading the posts of employees. On [PurpleGroup](#), a social network on which employees can rate their employers, information is divulged to explain situations, benefits, and disadvantages of companies (the name of the social network is not printed to protect the identity of [YuuBlue](#)). There, we learned of the [YuuBlue Day](#) which enabled the execution of SEP03 (see [Chapter 18](#)).

Part II

Acta Non Verba

Deeds Not Words

Attack Categorization and Overview

To keep the multitude of different attacks under control, we differentiated between *digital* and *physical* social-engineering attacks. The attacks are assigned a prefix (**SED** for Social Engineering Digital, **SEP** for Social Engineering Physical), a number, and a name, to make them distinguishable and simultaneously accessible for discussion.

We planned and prepared each attack carefully and added them strategically to the attack schedule so that we could execute them subsequently. Figure 5.1 on page 39 gives an overview of the executed attacks. Often, specific details of additional attacks arose from the results of previous ones. Figure 5.1 tries to emphasize this inspirational effect by describing what information enabled which attacks and vice versa.

The **dark boxes** in the top row in Figure 5.1 depict initial information that was available to everybody. The small, **brighter boxes** depict information that was gained through actions on the part of the attacker. Finally, the **large boxes** depict the actual attacks that were executed. The connecting arrows show the flow of information, from the point where the information originated (its source) to the places it was used again.

The remainder of this chapter addresses the ideas and schemes which, unfortunately, could not be executed for various reasons (e.g., a lack of time, or the effort not to burden the employees any further). The following chapters then detail all the attacks that *were* executed.

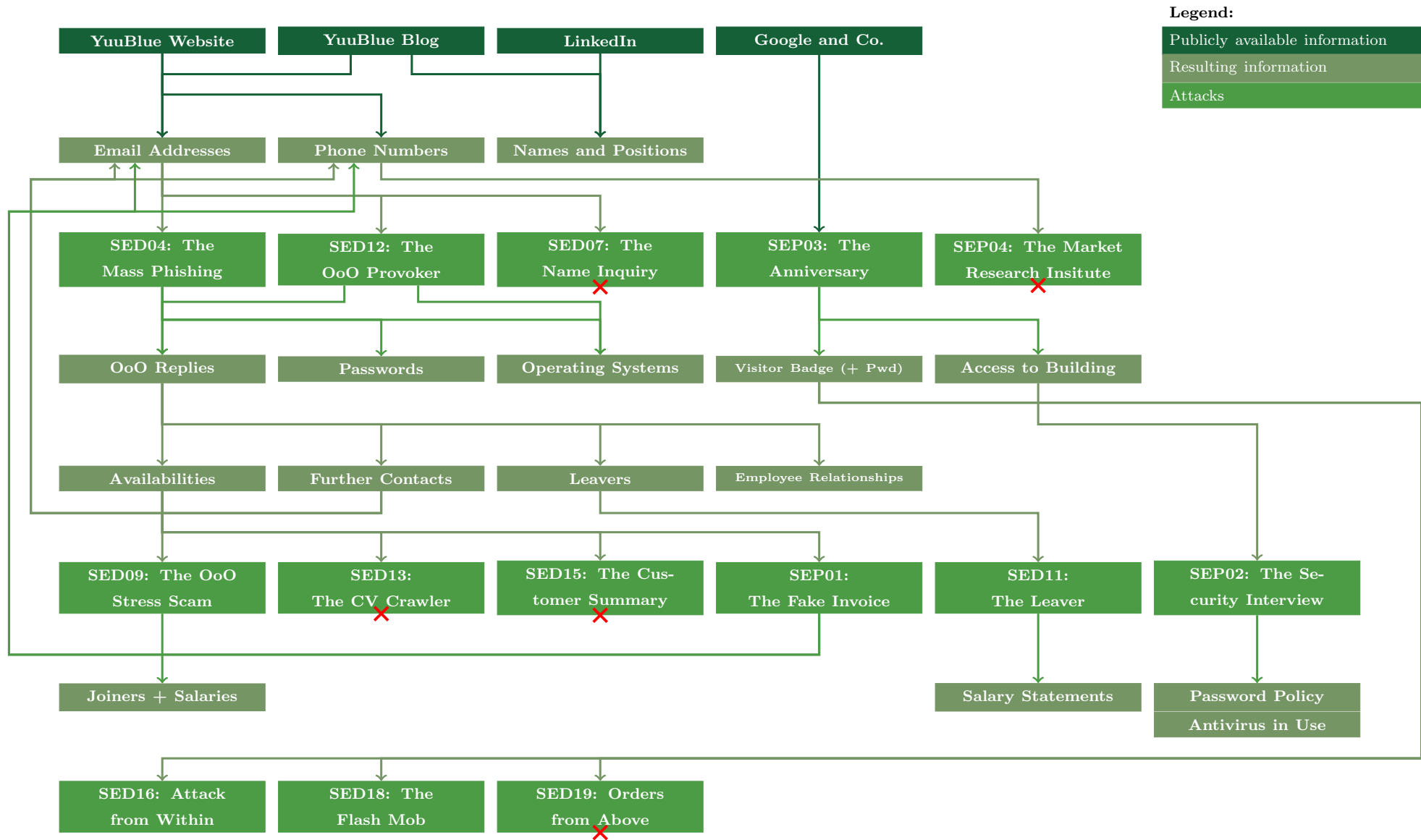


Figure 5.1: Attack Overview

5.1 Unexecuted Attacks

As the attentive reader might have noticed, the names of the schemes that are presented in the following chapters are not consecutive. During the course of this thesis, we had many ideas and planned many attacks. Some did not make it past a footnote or a quick bullet point, but others made it so far as to get a name and identification number.

Having a name and identification number means that a scheme or idea was sufficiently concrete and that enough planning had gone into it to be executed. Unfortunately, for various reasons presented below, we were not able to execute all of our plans. Hereafter, we provide a list of the attacks that were planned but never executed:

SED01: The Company Profile Spear-phishing attack with a malicious payload. Not executed because the payload was not fully undetectable and not fully adapted to the target’s operating system. More investigation would have been necessary.

SED02: The Thesis Proposal Spear-phishing attack with a malicious payload. Not executed because the payload was not fully undetectable and not fully adapted to the target’s operating system. More investigation would have been necessary.

SED03: The Speculative Application Spear-phishing attack with a malicious payload. Not executed because the payload was not fully undetectable and not fully adapted to the target’s operating system. More investigation would have been necessary.

SED05: The Printer Mail Send an email that looks like a scan of a local printer to various targets. The payload attached should have been a malicious PDF. Not executed because we were unable to produce a malicious PDF for newer versions of Adobe Reader and SEP03 (see [Chapter 18](#)) was aborted too early to send ourselves a sample email from one of the printers.

SED06: The Massive Distribution List Send an email to as many email addresses as possible, camouflaging it as a distribution list gone wrong. The numerous answers of employees hitting “reply all” to unsubscribe from the distribution list might have overwhelmed [YuuBlue](#)’s email system (for a definition of [Denial of Service \(DoS\)](#) see [Section 2.4](#)). Not executed because the damage would be too severe.

SED08: The Recruiter Horror Upload a malicious document to the recruiting platform of [YuuBlue](#). Not executed because we were unable to fabricate a malicious PDF for newer versions of Adobe Reader.

SED10: The Bloggers' Fanboy Spear-phishing attack with a malicious payload. Not executed because the payload was not fully undetectable. More investigation would have been necessary.

SED14: The Fake Recruiter Send a fake job recruitment email to specific targets and try to gain information about [YuuBlue](#) during the negotiation process. Not executed due to lack of time.

SED17: The Important Meeting Invite management to a fake meeting in the name of the CEO (similar to SED18, see [Chapter 14](#)). Not executed in order to not endanger the success of SED18 due to simultaneous execution.

We have not executed these and other attacks because of our inability to produce malicious payloads (PDF documents, Microsoft Word documents, or custom executables) that were undetectable by common antivirus software (and specifically by the antivirus software in use at [YuuBlue](#)). More investigation/research would have been necessary. Due to the limited time window of a Master's thesis, we decided to focus on other attacks instead of bothering with technical obstacles. [Section 3.3](#) and [Section 3.4](#) describe two of our attempts to create executables containing a Meterpreter payload (see [Section 2.7.1](#)). Besides using Meterpreter, we also tried to create a keylogger, with similar success.

In addition to the planned attacks above, many hardware-related ideas, like distributing malicious USB drives or installing cameras to spy on door access codes, crossed our minds but were ultimately discarded for time and budget reasons.

SED04: The Mass Phishing

In contrast to SED01-03 (see [Section 5.1](#)), SED04 targeted a large number of people simultaneously. It was a typical phishing attack whose goal was to extract login credentials from as many employees as possible by tricking them into entering their credentials into a legit-looking login mask which the employees use on a daily basis.

6.1 Preparation

There were a number of essential questions that had to be answered before the attack was ready to be launched:

Where and How Can We Procure Large Amounts of Email Addresses?

[Section 4.1](#) answers this question in detail.

The final attack only used addresses which the address crawler had found on the [YuuBlue](#) website. At the time, we had not yet crawled any LinkedIn addresses.

How Can We Procure a Legit-Looking Login Mask?

Acquiring a login mask can often be achieved by doing a DNS enumeration, as described in [Section 4.2](#).

The DNS enumeration of the [YuuBlue](#) domain yielded a login mask that was used to log into an email service.

Copying such a login mask is not difficult, requiring only to download the corresponding source files. By adding just three lines of JavaScript, the entered login credentials can be exfiltrated and can subsequently be sent to a PHP script for processing.

How Should We Transport and Store the Exfiltrated Passwords?

To guarantee the confidentiality of the exfiltrated passwords, we sent all data over HTTPS. The passwords were never stored in plain text. Instead of storing the passwords in plain text, a PHP script masked half of the characters of the password with placeholder characters and counted, for statistical purposes, how many characters of which type (upper characters, lower characters, special characters, and digits) the password contained.

How Can We Distinguish the Employees Who Click on the Link?

Tricking the employees into clicking on the link in the email required the email to be an HTML email, allowing the link to look like the main company domain when it was not:

```
<a href="https://evilbank.com">https://bank.com</a>
```

At the same time, we were able to send along the first 12 characters of the hash of the employee email address as a GET parameter, thereby personalizing the links. With the help of these hashes, the passwords were easily linkable to the corresponding email addresses and employees.

```
<a href="https://evilbank.com/?p=5d41402abc4b">  
    https://bank.com  
</a>
```

Simultaneously, every visit to the fake login page stored the exact timestamp of the visit, and the visitor's operating system, and the visitor's IP address.

Where Will We Host the Fake Login Page?

As described in Section 3.1, we ordered a VPS with corresponding domain `yuubluе-support.com`. This server hosted the fake login page under the path `yuubluе-support.com/it-center/services/pw_change`. The long path was chosen to arouse less suspicion than a short and direct `yuubluе-support.com/pw_change`.

Alternatively, the login page could have been hosted using a free web space provider like Awardspace (see SED12 in Chapter 10).

When Should We Send the Email?

We assumed that most of the employees were out-of-office over the weekend. On Mondays, they were probably recovering from the weekend (for example, answering emails that came in over the weekend) and on Fridays, they might be more interested in wrapping things up, than in starting anything new. Emails that come in on Friday afternoon might get forgotten over the weekend. These considerations left us with the prime candidates Tuesday, Wednesday, and Thursday for the phishing attack to have the largest impact.

Much consideration went into the time of day at which the phishing email should be sent.

If we sent the email in the morning, the staff would have the opportunity to talk about it at lunch, which might expose the attack.

If we were to send the email during a time where meetings usually take place, many employees would probably not have time to react to the email before countermeasures like blocking of the IP address are put in place.

To prevent the IT center from responding too quickly, we could send the email in the evening, when the IT center is not working anymore. However, the rest of the employees might not be working anymore either in the evening.

In the end, we settled for a Tuesday morning, ignoring meetings and hoping for the best.

Who Should Be the Sender of the Email?

Who has the authority within `<YuuBlue>` to send an email requesting the employees to log in? The IT center seemed like a good candidate, which is why we decided to use the email address `it-center@yuublue-support.com`.

How Can We Prevent the IT Center from Blocking the Server and Its IP Address?

After such a noisy phishing attack, it was likely that the server IP address and domain are going to be blocked by `<YuuBlue>`. As a result, nobody within `<YuuBlue>`'s network would then be able to access the phishing website anymore, which renders the execution of any further attacks using the same server, domain, or IP address impossible.

To prevent blocking of the IP address by the IT center of `<YuuBlue>`, we registered a second domain (`pizzeria.com`). This domain pointed to the VPS which displayed the *fake* website of a *real* Pizza place located in Zurich (the Pizza place was not informed about this which is why a `robots.txt` tried to prevent Google from crawling this website). Since the two domains pointed to the same webpage, we implemented a PHP switch that redirected visitors

for yuublue-support.com to the real yuublue.com website and let visitors for pizzeria.com pass.

The idea behind the second domain was that if the IT center did a reverse DNS query on the server IP, they would find two domains registered for this server. This would then hopefully trigger the investigator to think that the Pizza place's server got hacked which would lead to the Pizza place being informed (hopefully by email, which was controlled by the attacker) rather than to the entire domain being blocked.

However, if the investigators were to have an even more profound look, they would find that pizzeria.com was registered later than yuublue-support.com, which would be a sign that both websites were fake.

As it turned out, nobody has ever found the pizzeria.com website and the IT center blocked the server regardless of our countermeasures. Even if they had found the website, it is questionable if they had cared enough about a Pizza place in Zurich to not block their website.

6.2 Execution History

Timestamp 1

Wednesday, 2018/02/28:

We create the fake email account `<julius.caesar>@protonmail.com` and use it to buy a VPS (see [Section 3.1](#)) and to register the domain yuublue-support.com, all in the name of the CEO of `<YuuBlue>`, `<Julius Caesar>`.

Timestamp 2

Monday, 2018/03/26, 10:45:

We write a Python script (see [Appendix A](#)) that crawls the entire `<YuuBlue>` website for email addresses. It finishes after running for about 12 hours, yielding 305 email addresses (see [Section 4.1.1](#)).



Mitigation 1

Avoid Publishing Contact Details

Email addresses and other details, such as street addresses, phone numbers, or employee names and positions, are valuable information and should be treated as such.

On the company website, these details **should only be published if absolutely necessary**, to ensure that an attacker cannot gain arbitrary contact handles to arbitrary employees within the company. See [Sec-](#)

tion 4.1.1 for suggestions on how to cautiously display email addresses online.

Every employee whose contact details can be attained publicly needs to be informed about the dangers of social engineering to render the frontline of the company impenetrable (see [Section 21.1](#)).

Timestamp 3

Friday, 2018/03/30, 13:59:

A Python script (see [Appendix A](#)) that is capable of sending the same email to an extensive list of addresses is created and successfully tested. It runs on the VPS and sends a single email to every recipient in the crawled address list. It uses `sendmail` as the outgoing SMTP server, which is also running on the VPS.

Timestamp 4

Friday, 2018/03/30:

Via DNS enumeration, we find a login page used by [YuuBlue](#). The login page is cloned, and a slight change to its JavaScript functionality allows to exfiltrate the entered login credentials. After submitting the credentials, the page forwards to the real login page in the hope the users do not notice anything out of the ordinary. The login page sends the credentials with an AJAX request over HTTPS to a PHP script (see [Appendix A](#)) that masks the passwords and stores them in the database, along with a timestamp, the hash of the target's email address, the target's operating system, and the target's IP address.

Timestamp 5

Thursday, 2018/04/05:

We register the domain [pizzeria.com](#) (see [How Can We Prevent the IT Center from Blocking the Server and Its IP Address?](#)).

Timestamp 6

Monday, 2018/04/16, 15:35:

To make the email seem more trustworthy, we want to find an employee working in the IT center whose name can be used to sign the email. For this purpose, we create and execute SED07 (see [Chapter 7](#)). Unfortunately, [YuuBlue](#) discloses no name and the phishing email has to be sent using a generic sender name.

Timestamp 7

Tuesday, 2018/04/17, 10:00:

On the day of the attack, we extensively test our setup.

The Python script that is used to create and send the emails has to function smoothly. Can the script send 300 emails in a row? Can the script send emails to non-existing email addresses without crashing? How long does it take to send 300 emails? All of the tests are successful.

We set up a catch-all email address for the domain yuubblue-support.com so that no email sent to the server creates a bounce and thereby discloses information to the sender (and potential threat investigator).

We also create the sender email address it-center@yuubblue-support.com, so that all bounces, replies, and automatic out-of-office replies can be received.

Timestamp 8

Tuesday, 2018/04/17, 10:36:

We send the following email to 305 [YuuBlue](#) employees (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: Security Prompt – [YuuBlue Support](#)**

Dear Colleagues,

We just became aware of the fact that the passwords of some [YuuBlue](#) accounts have been stolen as the result of a phishing attack.

In order to prevent any further damage, we ask you to **immediately** change your password:

https://yuubblue.com/it-center/services/pw_change

Beware of phishing: Attachments in emails coming from external sources should only be opened in case the sender is known and trusted, and the email is expected.

Following these rules, future data leakages can be prevented.

We will inform you as soon as our investigations concerning the

stolen data are concluded.

Best regards,
Your IT Center

”

The link in the email does not lead to yuublue.com, but to yuublue-support.com, and is personalized so that the targets can be distinguished.



Mitigation 2

Spot a Phishing Email

Do not let yourself be fooled by perfect layout, logos, and execution. Fraudsters are professionals and are often just as good as the designers and IT specialists of the entities they are trying to impersonate. Always treat emails with a healthy load of suspicion and train yourself to be easily triggered on the following details [50]:

- Tip 1** When hovering a link in the email body, check the URL *before* clicking. If it looks suspicious or unexpected (e.g., https://yuublu.com/change_password, instead of https://yuublue.com/change_password), the email might be phishing.
- Tip 2** Most companies take a lot of care when they send a newsletter or message to you. If the email contains many (obvious) spelling mistakes or poor grammar, the email might be phishing.
- Tip 3** Often, companies address their customers with their real name. If an email addresses you with “Dear subscriber”, “Valued customer”, or another generic verbiage, the email might be phishing.
- Tip 4** No legitimate and professional organization will ever ask you for credentials or personal information in an email. If the sender still requests such or similar data, the email might be phishing.
- Tip 5** No legitimate and professional organization will ever threaten you. If the email contains threatening language insinuating urgency to do something (e.g., change your password), the email might be phishing.
- Tip 6** A legitimate and professional organization usually signs an email with secondary contact channels, like the name of a contact person and a phone number. If such information is lacking, the email might be phishing.

Tip 7 Because of public data breaches [51], you may receive an email containing correct information about yourself, or even passwords that you used in the past [52, 53]. Do not let this fact increase your trust in the sender. Change the passwords of all accounts that are still using the breached password and treat the email as you would treat any other phishing email, regardless of the threats that may accompany the phishing email.

If an email matches one or more of these points, it is possible that it is a phishing email. However, even emails matching none of the points may be phishing, and vice versa, a legitimate email could match multiple points and still be sincere.

Since the phishing authors never stand still and devise new phishing schemes on a daily basis, these rules may need to be adapted in the future.

The goal of phishing emails is often to steal login credentials. Using two-factor authentication (2FA), the number of phishing incidents can be significantly reduced [54].

Timestamp 9

Tuesday, 2018/04/17, 12:29:

The web server is shut down to prevent any further investigation by <YuuBlue>'s IT center, following large numbers of page visits by the same IP.

Of 305 emails, 60 (19.67%) could not be delivered (the email addresses may not have existed anymore) and 22 (7.21%) triggered an automatic out-of-office reply. The remaining 223 (73.11%) emails were delivered successfully and produced 47 (21.07% of the targets clicked on the link) unique visits to the login page and resulted in 23 unique passwords (48.93% of page visitors entered their passwords).

See [Figure 6.1](#) on page 59 for a graphical representation.

Some people logged in twice and others entered their email address and a fake password (nonetheless, thereby surrendering their operating system information which could potentially be used in further, targeted attacks).

If the 17 emails are subtracted that went to an English speaking branch of <YuuBlue> (they could not have understood the content of the phishing email) the percentage of people clicking on the phishing link rises to 22.81%.

Most of the clicks and logins happened within the first 15 minutes after the phishing email was sent.

 **Mitigation 3 Handle Links and Attachments Suspiciously**

Links and attachments in emails should always be treated carefully.

To evaluate if it is safe to click on a link or to open an attached document, you should consider the following points:

Yes Do you know and trust the sender of the email (if it is a generic sender address, such as `info@company.com`, or an external address with which you never had contact before, the answer is no)?

Expected Did you expect the email or is it coming unexpectedly?

As promised When hovering the mouse over the link, does the link lead to where it promises or is it a different address (pay close attention to the domain name)?

No Does the email prompt you to enter data after having clicked the link and/or to open a document?

If the answers to these questions deviate from the “safe” answers in front of the questions, then refrain from clicking any links or opening any documents.

If you are still uncertain, consider to **double-check the facts** mentioned by the sender or **ask a colleague for a second opinion**. If the email demands a login, a password change, or any other information, do not use the link provided in the email. Instead, open your browser and log in like you usually do.

While simply clicking the link does not do significant damage, the attacker still gains knowledge about your system which could be used in future attacks. Therefore, it is recommended to strictly avoid clicking phishing links.

See also [Section 21.2, How to Handle Links and Attachments in Emails?](#) on page 149.

 See also [Avoid or Limit Out-of-Office Replies](#) on page 135.

Timestamp 10

Wednesday, 2018/04/18, 15:03:

The `<YuuBlue>` IT center had notified the VPS provider that one of their root servers was being used to host phishing websites and that identity theft was being committed.

As a result, the VPS provider sends the following email to `<julius.caesar>` `@protonmail.com` (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: [Ticket#66279921] Phishing on Your Root Server

Dear Mr. `<Caesar>`,

We were informed that your system contains so-called phishing websites.

You can find more information about phishing here:
<http://de.wikipedia.org/wiki/Phishing>

The affected URL is
https://www.yuublue-support.com/it-center/services/pw_change

Because it is a root system, you are in charge of analyzing and eliminating the problem. However, please note that our terms and conditions exclude pages of this kind, and any further incidents could lead to termination without previous notice.

Moreover, we have been alerted about a possible identity abuse. That is why we ask you to send us a copy of an identity document or another proof of your identity.

Yours sincerely,

`<George Orwell>`
Customer Service

`<Address of Server Provider>`
`<Website of Server Provider>`

”

Timestamp 11

Monday, 2018/04/23, 16:13:

We explain the situation to the VPS provider and tell them to lie to whoever tipped them off (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: RE: [Ticket#66279921] Phishing on Your Root Server**

Dear Mr. [Orwell](#),

Please excuse the late reply.

I am impressed by how seriously you take the concerns of your customers. If all providers reacted as exemplary as you do, the Internet would be a more secure place.

Unfortunately, this is often not the case. For this reason, [YuuBlue](#) performs penetration tests at the moment, in coordination with the executive management (<https://yuublu.com/management>). To that end, the phishing website that was mentioned by you was uploaded to the system.

As you undoubtedly know, penetration tests of this kind only work if an absolute minimum number of employees know about them. The alert that you received was part of the (excellent) reaction of our IT center to the penetration test that took place.

The IP address of the server has been blocked, which is why we will not perform any further tests on this root system. Therefore, we ask you, full of confidence in your competent actions, to tell our IT center that you have handled the problem with the root server and that you have terminated the server. If you do not plan to give any feedback to our IT center, all the better.

If you happen to have any further questions in this matter, please contact our CISO [Albert Einstein](#) ([Albert's Email Address](#)), [Albert's Phone Number](#), [Albert's Workplace Address](#)). He is the only member of the IT center who knows

about the penetration tests. You can talk openly with him.

For the further success of the penetration tests, it is imperative to keep the list of confidants as small as possible. The phishing page has been removed from the root server and I can promise you at this point, that no more phishing websites will be uploaded.

I thank you very much for your cooperation and I am sorry for the expense incurred.

Yours sincerely,
⟨Julius Caesar⟩

”

Note that we did not send along any proof of identity whatsoever in this email. As it turned out, the server provider never contacted the CISO about the claims in this email. Nor did they contact ⟨julius.caesar⟩@protonmail.com ever again.

After the end of the project, the IT center kindly made their side of the conversation with the domain provider of yuublue-support.com available (see [Findings and Post-Execution Insights](#) in [Section 6.3](#)), which makes for an exciting read.

Timestamp 12

Tuesday, 2018/04/24, 12:40:

The DNS entry for the domain pizzeria.com is removed to prevent any problems with the real Pizza place in case anyone finds the website.

We close SED04.

6.3 Findings and Post-Execution Insights

SED04 was the most successful scheme of the entire project. First and foremost, it provided us with passwords (see the exact numbers in [Timestamp 9](#) on page 49). It also provided insights into how many people clicked on the phishing link and in what time span. Even though some people knew the link was a phishing link, curiosity won and they clicked on it anyway, entering fake credentials to

make us angry. Simultaneously though, these people surrendered their operating system to the attackers who could now prepare targeted attacks against specific employees.

However, the most valuable information gained from SED04 were, as it turned out, the out-of-office replies. These contained further contact information and, most importantly, the exact dates when an employee will return to the office again. In some rare cases, insider-knowledge like the name of a project could be learned from them as well. The out-of-office replies enabled the schemes SED09 and SED11 to be successfully executed (see [Chapter 8](#) and [Chapter 9](#)).

SED04 caused much upheaval in the entire company. On the day of the attack, there was no other topic. The employees talked about it in breaks and over lunch, and they did so for the better part of a week, wondering who was behind it and how they did it, if it was an inside job or rather a disgruntled ex-employee. Apparently, the email had hit the precise tone that the IT center uses in their legitimate emails, which was a lucky punch since we executed SED04 *without* insider-knowledge.

Unsurprisingly, the IT center was having a rough week while investigating the attack and dealing with the aftermath. After the project was unveiled two months later, they kindly provided the following conversation with the provider of the domain yuublue-support.com:

Timestamp 13

Tuesday, 2018/04/17, 16:15:

The original text was written in German, it is translated here for purposes of reaching a broader audience:

**“ Subject Line: [KS#2018041710007948] Phishing with
yuublue-support.com**

Good Day,

Our company [YuuBlue](#) was the victim of a phishing attack today. Phishing emails were sent to over 250 employees with the request to change their passwords with the link provided in the email.

The provided link:

https://www.yuublue.com/it-center/services/pw_change

pointed to the this host in reality:

www.yuublue-support.com;http://www.yuublue-support.com;
[https://www.yuublue-support.com/it-center/services/
pw_change](https://www.yuublue-support.com/it-center/services/pw_change)

On this website, our login page has been recreated in order to get usernames and passwords.

The domain was registered in the name of our CEO [\(Julius Caesar\)](#), which is false.

Was [yuublue-support.com](#) registered at your company?

Could you help us to stop this attack? The phishing website is still active!

Kind regards,
[\(Thomas Edison\)](#)
Head of IT Center
[\(Contact Details of Thomas\)](#)

[\(YuuBlue Address\)](#)
[\(YuuBlue Website\)](#)

This e-mail is for the addressees only. The information it contains is confidential and may be legally privileged. If you are not an addressee you must not distribute, copy, disclose, use or rely on this e-mail or its contents and you must immediately notify the sender you are in receipt of this e-mail and delete all copies from your system. Any unauthorised use may be unlawful. ”

This email was sent in the afternoon, claiming that the phishing website was still online, which it was not (see [Timestamp 9](#)). In retrospect, it seems to have been an excellent choice to take the server offline, or the domain provider’s answer might have looked differently.

Timestamp 14

Wednesday, 2018/04/18, 11:19:

The domain provider forwarded the complaint to their reseller (the company at which we registered the VPS, see [Section 3.1](#)) and promised to get back to [\(YuuBlue\)](#) as soon as they knew more (the original text was written in German,

it is translated here for purposes of reaching a broader audience):

“ **Subject Line: RE: [KS#2018041710007948] Phishing with yuublue-support.com**

Dear Mr. <Edison>,

Thank you very much for your message.

The domain yuublue-support.com has been registered on our automatic system by our reseller <VPS Provider> for their customers.

We have examined the transmitted URL and found that it no longer resolves.

We have forwarded your complaint to our reseller for further processing.

As soon as we hear from them, we will get in touch with you again.

Best regards,

<Eva Peron>
Customer Support

<Eva's Contact Information>

<Domain Provider Address>

This e-mail and its attachments is intended only for the person to whom it is addressed. If an addressing or transmission error has misdirected this e-mail, please notify the author by replying to this e-mail or contacting us by telephone.

Furthermore it is not allowed to publish any content of this Email. If you are not the intended recipient you must not use, disclose, copy, print or rely on this e-mail.

”

The VPS provider reacted promptly, and only four hours later, the complaint had reached the inbox of <[julius.caesar](mailto:julius.caesar@protonmail.com)>@protonmail.com (see [Timestamp 10](#)).

Timestamp 15

Thursday, 2018/04/26, 07:09:

Three days after we had sent the appeasing email to the VPS provider (see [Timestamp 11](#)), the IT center received the following email from the domain provider:

“ Subject Line: RE: [KS#2018041710007948] Phishing with yuublue-support.com

Dear Mr. [⟨Edison⟩](#),

Our reseller has informed us that the matter has been resolved and the server has been terminated.

Best regards,

[⟨Eva Peron⟩](#)
Customer Support

[⟨Eva’s Contact Information⟩](#)

[⟨Domain Provider Address⟩](#)

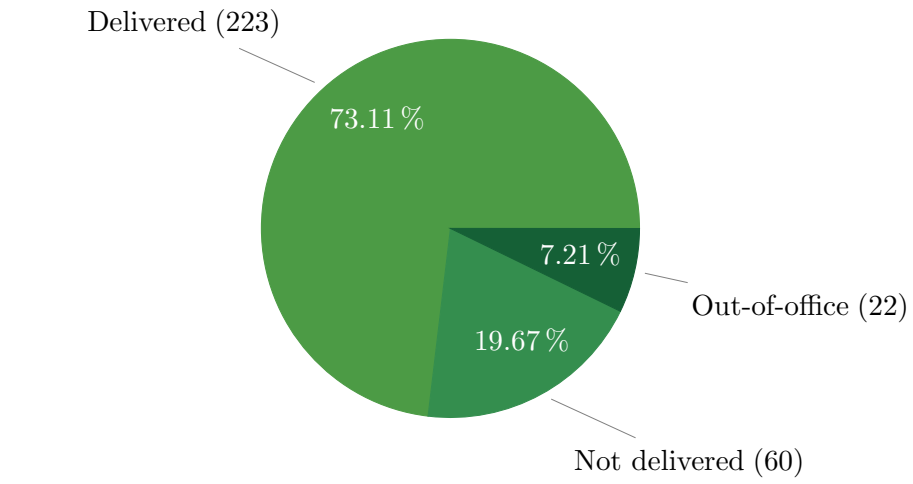
This e-mail and its attachments is intended only for the person to whom it is addressed. If an addressing or transmission error has misdirected this e-mail, please notify the author by replying to this e-mail or contacting us by telephone.

Furthermore it is not allowed to publish any content of this Email. If you are not the intended recipient you must not use, disclose, copy, print or rely on this e-mail.

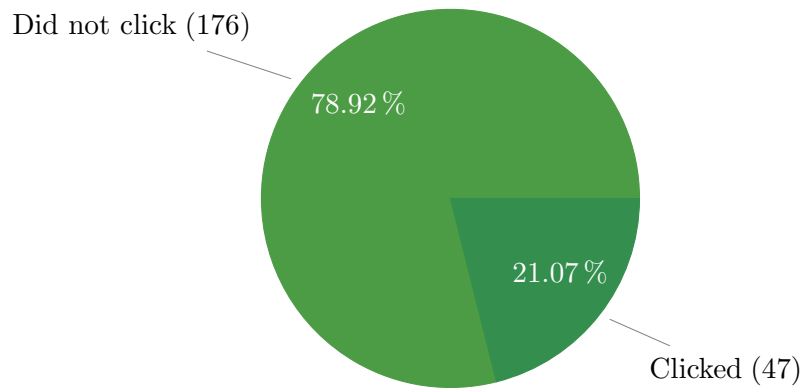
”

Even though the story that we told to the VPS provider in the appeasement email was true, it might just as well have been entirely made up. The VPS provider never double-checked the facts ([⟨Albert Einstein⟩](#), the CISO of [⟨YuuBlue⟩](#), told us that they had never contacted him) and blatantly lied to the domain provider that the server had been blocked. This wilful misinformation might be the most shocking finding of SED04. If a party stands to lose money from terminating a malevolent customer, they might not do it if they do not have to (not double-checking the facts gave them plausible deniability in case of

an investigation). What seems to be even worse is that they did not lie to the domain provider of their own accord. The appeasement email told them to do so, and they followed the instructions to the letter. The question arises what else we could have told them to do.

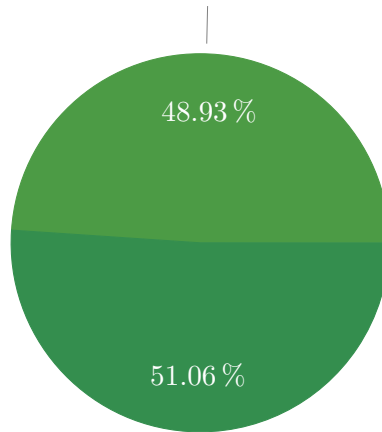


(a) Delivery Rate



(b) Click Rate

Clicked and Entered Password (23)



Only clicked (24)

(c) Password Rate

Figure 6.1: Statistics

SED07: The Name Inquiry

This scheme was part of SED04 (see [Chapter 6](#)). The goal of SED07 was to get the name of a person working in the IT center of [YuuBlue](#). We wanted to use that name to make the phishing email of SED04 seem more trustworthy.

Because the [YuuBlue](#) website did not provide such a name, another source of information had to be found. Frequently, it is enough to ask for information, and people give it up willingly to be of help.

7.1 Preparation

The important questions were:

Who Would Give the Name of an Employee to an Outsider?

It had to be somebody whose contact information was publicly available, for example on the website. HR and the reception desk seemed to be two good candidates because they often have legitimate contact to outsiders. Since there were other schemes planned that targeted HR, we decided to target the **reception desk**.

What Email Address Should We Use to Send the Inquiry?

Because the reception desk did not know Marlon Torben yet, we chose to reuse the email address `marlon.torben@outlook.com` that had initially been created for SED01 (see [Section 5.1](#)).

7.2 Execution History

Timestamp 1

Monday, 2018/04/16, 15:35:

We send the following email to `info@yuublue.com` (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: Open Position in the <YuuBlue> IT Center

Dear Sir or Madam,

My name is Marlon Torben. This summer I finish my apprenticeship in information technology and would like to ask you if there are any open positions in your IT center.

Could you possibly give me a contact within the IT center whom I can ask further technical questions before I submit an application on the official channels?

Yours sincerely,
Marlon Torben

”

Timestamp 2

Tuesday, 2018/04/17, 09:58:

Unfortunately, the <YuuBlue> reception desk is not inclined to give out any names or contact details. Instead, they offer to connect us to HR (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: Open Position in the <YuuBlue> IT Center

Dear Mr. Torben,

Thank you very much for your email.

In general, all our open positions can be viewed on our website,

sorted by country and city.

Any personal questions you can send to job@yuublue.com, along with your contact details. An employee of the recruiting team will then get in touch with you.

Kind regards,
 <Florence Nightingale>
 Head of Reception
 <Florence's Phone Number>

<YuuBlue Address>
 <YuuBlue Website>

This e-mail is for the addressees only. The information it contains is confidential and may be legally privileged. If you are not an addressee you must not distribute, copy, disclose, use or rely on this e-mail or its contents and you must immediately notify the sender you are in receipt of this e-mail and delete all copies from your system. Any unauthorised use may be unlawful. ”

Timestamp 3

Wednesday, 2018/04/18, 16:40:

Because <YuuBlue> has multiple international business locations, there is more than one reception desk that could potentially leak contact information. We send the following email to the reception desks of two other countries: <Berlin>@yuublue.com, <Vienna>@yuublue.com (the original text was written in German, it is translated here for purposes of reaching a broader audience).

“ **Subject Line: Open Position in the <YuuBlue> IT Center**

Dear Sir or Madam,

My name is Marlon Torben. This summer I finish my education in information technology and would like to ask you if there are any open positions in your IT center.

Could you possibly give me a contact within the IT center whom I can ask further technical questions before I submit an application on the official channels?

Yours sincerely,
Marlon Torben

”

To be on the safe side, we decided to send only one email per country, because it was hard to tell if the different offices of, for example, Germany, shared a single reception desk or not.

Unfortunately, the very next day, both reception desks answered negatively (see below).

Timestamp 4

Thursday, 2018/04/19, 09:16:

Austria answers our inquiry with the following text (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: RE: Open Position in the <YuuBlue> IT Center**

Dear Mr. Torben,

Thank you for your inquiry. Unfortunately, there are no suitable open positions at the moment.

I wish you all the best and remain with best regards,
<Sigmund Freud>
Human Resources Specialist

<YuuBlue Address in Austria>

This e-mail is for the addressees only. The information it contains is confidential and may be legally privileged. If you are not an addressee you must not distribute, copy, disclose, use or rely on this e-mail or its contents and you must immediately notify the sender you are in receipt of this e-mail and delete all copies from your system. Any unauthorised use may be unlawful.

”

Timestamp 5

Thursday, 2018/04/19, 12:37:

Germany replies with the following email (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: RE: Open Position in the <YuuBlue> IT Center**

Dear Mr. Torben,

Thank you very much for your email and your interest in employment at <YuuBlue>. At the moment, we can offer you only the following vacancy in our IT center:

<https://yuublue.com/cp?sid=1&ccid=133>

If you are interested, we would be happy to receive your application.

I would be happy to answer any questions you might have.

Many regards,
<Emily Dickinson>
Human Resources Specialist

<YuuBlue Address in Germany>

This e-mail is for the addressees only. The information it contains is confidential and may be legally privileged. If you are not an addressee you must not distribute, copy, disclose, use or rely on this e-mail or its contents and you must immediately notify the sender you are in receipt of this e-mail and delete all copies from your system. Any unauthorised use may be unlawful.

”

The link lead to a job offer on the <YuuBlue> website.

Friday, 2018/04/20:

Due to no success, we close SED07.

7.3 Findings and Post-Execution Insights

Even though finding a name of an employee in the IT center of `<YuuBlue>` was not possible, the scheme at least yielded three new names, email addresses, and phone numbers, belonging to the Swiss reception desk, the Austrian HR department, and the German HR department. Every handle an attacker gains is valuable and could be used in later, targeted attacks.

The reception desks and recipients of the emails sent to the generic company addresses (like `<Berlin>@yuublue.com`) seem to be well trained in avoiding to disclose any more information than they need to. This resistance is a good sign for `<YuuBlue>` and may be the result of educating the employees and preparing them for situations like this.

The answers Marlon received concentrated on his inquiry about a job opening. One could argue that the inquiry text was not aggressive enough in trying to obtain a handle in the IT center. On the other hand, being more aggressive might trigger an alarm at the receiving end.

SED09: The OoO Stress Scam

Among the out-of-office replies that we received after executing SED04 (see [Chapter 6](#)), one belonged to the [YuuBlue](#) head of HR: [Eleanor Roosevelt](#).

“ Thank you for your message. I am out of the office until 24 April 2018 and have limited access to my e-mails during that time. ”

[Eleanor](#), as head of HR, was among the few people who knew about project SOEP. Her employees, however, did not. It was the perfect opportunity to pretend to be her.

8.1 Preparation

The only question left to answer was:

To Whom Should [Eleanor](#) Write?

Thanks to the [YuuBlue](#) website, we knew that [Eleanor](#) was the head of HR and a member of management (see [Section 4.5](#)). Because we were working without insider-knowledge, we had no clue as to who was working under her.

Luckily, the [YuuBlue](#) website featured a blog, filled with articles written by its employees. The blog had a page where all the authors were listed with their complete name, email address, social media accounts like Twitter or LinkedIn, more often than not a picture, and their **current position within the company**.

It was a goldmine for any attacker. Not only did the blog disclose social media accounts and email addresses (most of the email addresses crawled in SED04 originated from this one page), it even exposed a picture of the person for

everyone to see. Even though pictures may be deceiving, they give the attackers an idea to whom they are talking to and what approaches might work.

With a simple, in-browser search via *Ctrl+F*, we found [〈Marie Curie〉](#) who was working in HR, along with her email address (which we could have also guessed from her name, see [Section 4.1.1](#)).

Later, LinkedIn served as a reliable source for job positions (see [Section 4.1.2](#)). However, at the time of the execution of SED09, we had not yet crawled LinkedIn for email addresses and data.

8.2 Execution History

Timestamp 1

Wednesday, 2018/04/18, 13:20:

We check the [〈YuuBlue〉](#) website for HR employees and find [〈Marie Curie〉](#), along with her email address.



See [Avoid Publishing Contact Details](#) on page 45.

Timestamp 2

Wednesday, 2018/04/18, 13:40:

We create the email address [〈eleanor.roosevelt〉@mail.com](#).

Timestamp 3

Wednesday, 2018/04/18, 16:36:

We send the following email to [〈Marie Curie〉](#). In the email, we mention the CEO of [〈YuuBlue〉](#), [〈Julius Caesar〉](#), to convince [〈Marie〉](#) of the urgency of the matter (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: Joiners March 2018

Hello [〈Marie〉](#),

I urgently need a list of the joiners for March 2018. The list I have with me seems incomplete and [〈Julius〉](#) has asked me to let him know as soon as possible. Since I'm not in the office, I

have no access to my emails (that's why I write from my private address).

Can you please send me the list briefly (Excel or Pdf, whichever is faster for you)? That would be nice.

Kind regards,

<Eleanor>

”

Note that we did not append any signature to the email. We hoped that since the email was coming from a private address, the lack of a signature would not be any more suspicious than the email itself was.



Mitigation 4

Mistrust Unusual Email Addresses

If an email is not sent from the company address or the address that is usually used, but from a “private” (different) address, the email should be treated with the utmost suspicion, even if the sender gives a seemingly valid reason for not using the usual email address.

Never trust such an email, because anyone could sit at the other end. Instead, **double-check the facts** via a second channel (for example, phone, VoIP, SMS, or text chats). If you are unsure about an email, **consult a colleague** for a second opinion.

Timestamp 4

Wednesday, 2018/04/18, 16:54:

Only twenty minutes later, we receive the following email (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: Joiners March 2018

Dear <Eleanor>,

Here is an overview of the joiners in March:

⟨Joining date, full name, work quota, position, team, and department of seven employees⟩

Is that fine?

Kind regards,
⟨Marie⟩
HR Operations Specialist
⟨Direct Phone Number⟩

⟨YuuBlue Address⟩
⟨YuuBlue Website⟩

This e-mail is for the addressees only. The information it contains is confidential and may be legally privileged. If you are not an addressee you must not distribute, copy, disclose, use or rely on this e-mail or its contents and you must immediately notify the sender you are in receipt of this e-mail and delete all copies from your system. Any unauthorised use may be unlawful. ”



Mitigation 5

Encrypt Data for Transmission

The simple mail transfer protocol (SMTP) is ancient and was not designed with security in mind. Anyone listening can see your email traffic.

As a consequence, sensitive data should never be sent in plaintext over the email protocol. Instead, **create an encrypted archive** and put the sensitive data you want to send into the archive.

Do *not* write the password for the archive in plaintext in the email. Instead, **drop a hint in the email** to which only you and the intended recipient can relate to (for example, the password is the name of the restaurant we had lunch at yesterday). Use a different password for every encrypted archive, and try to include digits as well.

If possible, do not mention the password in the email at all and relay it via a second channel.

Timestamp 5

Wednesday, 2018/04/18, 17:44:

Inspired by the success, we decide to be even bolder and ask for the salaries of the seven joiners (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: RE: Joiners March 2018

Hello [⟨Marie⟩](#),

Thank you for your quick reaction! [⟨Julius⟩](#) still needs the salaries for the coordination with [⟨Green15⟩](#). Can you please give me the salaries according to the contracts?

Please excuse the circumstances!

Thank you and kind regards,
[⟨Eleanor⟩](#)

”

Notable here is that we had no idea if [⟨YuuBlue⟩](#) had any business at all with [⟨Green15⟩](#) (in the original email, we chose the name of a large, reputable Swiss telecom company). We only dropped the name of the CEO (repeatedly, in this and the last email) and the name of a large company, in the hope to create a sense of urgency.

Timestamp 6

Wednesday, 2018/04/18, 17:44:

Again, about twenty minutes later, we get an email with the information requested (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: RE: RE: Joiners March 2018

Dear [⟨Eleanor⟩](#),

Here you go :) the salaries are on 100% basis.

Gotta go pick up the kid now.

Have a nice vacation!

Kind regards,
⟨Marie⟩

Sent: Wednesday, April 18, 2018 at 4:54 PM
From: "⟨Curie, Marie⟩" ⟨marie.curie⟩@yuublue.com
To: "⟨Eleanor Roosevelt⟩" ⟨eleanor.roosevelt⟩@mail.com
Subject: RE: Joiners March 2018

Dear ⟨Eleanor⟩,

Here is an overview of the joiners in March:

⟨Joining date, full name, work quota, position, team, department, **and salary** of seven employees⟩

Is that fine?

Kind regards,
⟨Marie⟩
HR Operations Specialist
⟨Direct Phone Number⟩

⟨YuuBlue Address⟩
⟨YuuBlue Website⟩

This e-mail is for the addressees only. The information it contains is confidential and may be legally privileged. If you are not an addressee you must not distribute, copy, disclose, use or rely on this e-mail or its contents and you must immediately notify the sender you are in receipt of this e-mail and delete all copies from your system. Any unauthorised use may be unlawful. ”

Sensitive data such as salaries should never be sent over an unsecured channel.



See **Encrypt Data for Transmission** on page 69.

Timestamp 7

Thursday, 2018/04/19, 07:13:

At this point, we decide to wrap it up and thank [\(Marie\)](#) for her time.

“ **Subject Line: RE: RE: RE: RE: Joiners March 2018**

Thank you very much and greetings,
[\(Eleanor\)](#)

”

Timestamp 8

Thursday, 2018/04/19, 10:14:

We back up and encrypt the messages and delete the emails from the [mail.com](#) server.

We close SED09.

8.3 Findings and Post-Execution Insights

SED09 was a scheme that went extremely smoothly. In less than a day, we went from a harmless out-of-office reply to having dangerous data like the joining date, full name, work quota, position, team, department, *and* salary of seven employees. An attacker could now further investigate all of these seven employees.

Apart from the extensive knowledge we gained about these seven employees, we furthermore gained information about [\(Marie Curie\)](#). The attackers would now have her direct work phone number, enabling them to call her personally and potentially get even more information from her. The attackers even learned that [\(Marie\)](#) has a little son (the original German email specified the gender of the child). This kind of personal information is something she undoubtedly does not want any strangers to have about her. It makes her vulnerable to an attack, be it emotionally by calling her and playing a YouTube video of a crying child in the background, in order to try to make her sympathize with the attacker, or be it more sinister by threatening her's or the child's safety (regardless of whether the attacker actually has access to the child). The latter could very well escalate in extortion for any information the attacker might want on the company.

Similar evil plans could be executed on the seven employees above. If attackers were to threaten one or more of them, they would probably set it aside

as a bad joke. However, if the attackers were to tell them their exact salary, the threat suddenly becomes very real. Having data they can impossibly know creates a mystery about the attackers, making them dangerous and terrifying (“What else do they know about me?”).

Information is power, and even worse, it is a power that can be gained relatively quickly and easily, as SED09 demonstrated.

The worst part was that at this moment, nobody even knew what had happened. Once [Eleanor](#) would have returned to the office, the leakage of information would have been discovered if [Marie](#) had asked [Eleanor](#) if she had received the list. In any case, it would have been too late, the attackers had already received the information.

Because [Eleanor](#) knew about the project, we asked her to play along to spare [Marie](#) the stress of knowing that she did something wrong or even of fearing for her job (for which she had no reason to).

SED11: The Leaver

Many of our schemes involved the “private” email addresses of [YuuBlue](#) employees who were out-of-office. Because the goal of the entire project was to sharpen awareness to the subject of security, people eventually caught on to the threat as more and more of these attacks were executed (see the failed attempts to get data in SED13 and SED15, in [Chapter 11](#) and [Chapter 12](#)). The employees started to treat such emails like they should be; with a healthy load of suspicion.



See [Mistrust Unusual Email Addresses](#) on page 68.

However, there are cases, where private email addresses are legit and do not arouse immediate suspicion. One such case, for example, is the case of a former employee who writes to the company.

9.1 Preparation

Questions we had to answer involved:

Do We Know a Former Employee of [YuuBlue](#)?

As we have seen in previous schemes like SED07 (see [Chapter 7](#)), automatic out-of-office replies are an excellent source of information (see [Section 4.3](#)). We remembered that we had encountered a few emails that said that the sender was not employed by [YuuBlue](#) anymore. After sifting through the replies once more, we found three former employees. Of these three, we picked [William Tell](#), because his name, in the original form, sounded the most Swiss and we were confident that he had been employed in Switzerland and not in one of the offices abroad.

“ Good day,

I no longer work at [YuuBlue](#). Your emails will not be answered anymore.

Please contact [YuuBlue](#) directly ([YuuBlue Front Desk Phone Number](#)).

Kind regards,
[William Tell](#)”

Similar information can also be found on LinkedIn, where people can be filtered according to their former employers.

What Is [William](#)'s Goal?

For what reason might a former employee write to his or her former employer? The employee might need salary statements, a work certificate, or a pension fund statement.

Even though creativity knew no bounds when finding a reason why [William](#) would write to [YuuBlue](#), we eventually settled for the salary statements, because they sounded the most interesting to us. Also, companies often send work certificates via the postal service, which would not work for us since we did not have his residential address and it would have been highly illegal to intercept his mail.

To Whom Does [William](#) Send His Request?

When requesting documents from an employer, the HR department is usually the right place to start. However, the [YuuBlue](#) website only listed info@yuublue.com as an HR contact. We did not want to use this email address since; first, the reception desk had proven to be pretty good with this sort of emails (see [Chapter 7](#)), and second, a former employee likely has access to a more direct email address to initiate such a request.

While skimming through the out-of-office replies of SED04 (see [Chapter 6](#)) again, we came across the following reply:

“ Good day,

I work part-time and can be reached on Wednesdays and Thursdays. In the meantime, please forward your request to `<hrdirect>@yuubluе.com`.

Many thanks and kind regards
`<Marie Curie>`”

The attentive reader might recognize the author of this automatic reply. It is the lovely lady who provided us with the list of joiners in SED09 (see [Chapter 8](#)). Serendipitously, she now provided us with a direct contact handle to the HR department.

9.2 Execution History

Timestamp 1

Wednesday, 2018/04/25, 09:36:

We create the email address `<william.tell>@yahoo.com`.

Timestamp 2

Wednesday, 2018/04/25, 10:23:

We send the following email to `<hrdirect>@yuubluе.com` (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: Salary Statement**

Dear HR

After the end of my time at `<YuuBlue>`, I have a final request. For the sake of the completeness of my documents, I would like to ask you to send me a salary statement for my last three months at `<YuuBlue>`.

I recently moved near my new employer, so I ask you to send me the relevant documents by email.

Kind regards,
⟨William Tell⟩

”

Timestamp 3

Thursday, 2018/04/26, 07:46:

On the next day, we receive the full package (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: Salary Statement

Hello ⟨William⟩,

I'll send you your April, March, and February salary statements attached. May I ask you to give us your new address? We can then adjust it in our system and take it into account when sending the wage statement next year.

Thank you and best regards,
⟨Rosa⟩

⟨Rosa Parks⟩
HR Operations Specialist
⟨Direct Phone Number⟩

⟨YuuBlue Address⟩
⟨YuuBlue Website⟩

This e-mail is for the addressees only. The information it contains is confidential and may be legally privileged. If you are not an addressee you must not distribute, copy, disclose, use or rely on this e-mail or its contents and you must immediately notify the sender you are in receipt of this e-mail and delete all copies from your system. Any unauthorised use may be unlawful.

”

Attached to this email were ⟨William⟩'s salary statements for the months February, March, and April (see [Section B.1](#) for the salary statement of April 2018).

See **Mistrust Unusual Email Addresses** on page 68.

See **Encrypt Data for Transmission** on page 69.

Timestamp 4

Friday, 2018/04/27, 12:44:

We try to go a step further and demand money for unaccounted expenses (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: RE: RE: Salary Statement**

Dear **<Rosa>**,

Thank you very much for the statements.

Unfortunately, my current apartment is only temporary. I'll send you my new address as soon as I have a permanent place. Is that alright with you?

Alas, I still have a problem. During my last customer visit, I had to unexpectedly buy a hard disk on the way, which I could not account for in the expenses anymore (see attachment). Is it possible that you transfer the amount to me retrospectively (since my bank account has also changed with my move, please transfer it to **<IBAN>**)?

Kind regards,
<William>

”

Note that we deftly avoided giving **<Rosa>** an address. We could have given her a fake one, but then, **<William>**'s address would have changed in **<YuuBlue>**'s system, creating much irritation and potential damage, if data was subsequently sent to the wrong address.

To the above email, we attached a photo of the receipt for a hard drive that we had acquired a while back in January 2018. Having such a receipt ready was pure coincidence. However, if we had not had that receipt, we would have faked one. The receipt inspired the text of the email, not the other way round. One

could have also gone with expenses of a different kind, for example, a lunch with customers.

In SEP01 (see [Chapter 16](#)) we had failed to get money from the Finance department directly. This time, we wanted to test the link between the HR department and the Finance department of [YuuBlue](#). Was the link so strong that if HR required Finance to pay an invoice that they would do so without questioning the request?

Timestamp 5

Monday, 2018/04/30, 08:55:

As an answer, we receive the following, by all means, positive email to our insolent request (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: RE: RE: Salary Statement

Hello [William](#),

Thank you for the information regarding your new address, that works for us.

I'll be happy to have a look at it next week with Sandra (she's on vacation this week) and then give you feedback.

Kind regards,
[Rosa](#)

[Rosa Parks](#)
HR Operations Specialist
[Direct Phone Number](#)

[YuuBlue Address](#)
[YuuBlue Website](#)

This e-mail is for the addressees only. The information it contains is confidential and may be legally privileged. If you are not an addressee you must not distribute, copy, disclose, use or rely on this e-mail or its contents and you must immediately notify the sender you are in receipt of this e-mail and delete all copies from your system. Any unauthorised use may be unlawful.

”

Timestamp 6

Tuesday, 2018/05/01, 11:17:

We do not want to seem impolite and send our thanks to [⟨Rosa⟩](#) (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: RE: RE: RE: Salary Statement

Hello [⟨Rosa⟩](#),

Thank you very much! I look forward to hearing from you.

Kind regards,
[⟨William⟩](#)

”

Unfortunately, the week passed, and we never heard back from [⟨Rosa⟩](#). Only later, we found out what our mistake had been.

Timestamp 7

Wednesday, 2018/05/09:

[⟨Eleanor Roosevelt⟩](#), the head of HR who knows about project SOEP, tells us what we already surmised. The last part of the scheme had failed for reasons explained below.

We back up and encrypt all messages and documents and delete the emails from the [yahoo.com](#) server.

We close SED11.

9.3 Findings and Post-Execution Insights

The results of this scheme are incredible. The salary statements we received contained information such as [⟨William⟩](#)'s residential address, his personnel number, his salary, and how much bonus he received. We learned how much children's benefits he received and from that how many children he had (two!). We came to know how much expenses [⟨YuuBlue⟩](#) paid him and how high his social security

contributions were. Furthermore, [William](#)'s bank account and the name of his bank were printed on the statements (see [Section B.1](#)).

This incident is an extreme breach of privacy while the affected person was not even related to [YuuBlue](#) anymore. Every employer stores incredibly sensitive data about its employees. The employer knows where they live, how much money they make, and if they are married and have children. The employer also knows their nationality and work permit, and other sensitive data. Such data allows an attacker to better impersonate an individual when talking, for example, to a bank or another entity while attacking the individual personally. This kind of information can be dangerous and harmful in the wrong hands, and all it took to get it was a single email containing the right words.

In addition to the information learned from the salary statements, we could add another person to our inventory of internal handles: [Rosa Parks](#). Even though she had failed to recognize our request for salary statements as being fraudulent and had not double-checked the facts with the real [William Tell](#) over a second channel, she successfully spotted our request for money. It seems like [William](#) had not had any customers in the eastern part of Switzerland. The receipt, however, was issued by a store in St Gall (northeastern part of Switzerland), which [Rosa](#) deemed to be odd. The strange case of [William Tell](#) subsequently went up the ranks until [Eleanor Roosevelt](#) told us the story.

Because the scheme had already fallen apart in the HR department, it never reached Finances. As a result, the link between HR and Finances remains untested.

Together with [Eleanor](#), we decided not to let the real [William Tell](#) know what had happened. Since we were the attackers, no data had left the company.

SED12: The OoO Provoker

The automatic out-of-office replies of SED04 (see [Chapter 6](#)) provided us with so much valuable information that we decided to provoke another round of out-of-office replies.

SED04 was very noisy and caused some problems for us since the [YuuBlue](#) IT center started to investigate the matter (see [Section 6.3](#)). Since this time we were only after the out-of-office replies, we tried to be as quiet as possible.

10.1 Preparation

We had to find answers to the following questions:

How Can We Be Quiet and Still Provoke As Many Replies As Possible?

SED04 had only been so noisy because we had obviously and aggressively phished for login credentials. A large part of the recipients immediately perceived the email as a threat and an attack against themselves or [YuuBlue](#) and acted accordingly. The IT center was quickly informed and started to counter the attack by blocking the server (which we did not know for sure at that time) and by warning the employees.

What if this time, we sent an email that the recipients perceived as a mere nuisance, rather than a threat? Such an email would, if done right, land in the inbox and would subsequently be removed from there by mildly annoyed recipients. The IT center would never even know that something had happened.

Emails of this kind are commonly called spam, and we were going to spam the entire company without anybody noticing it.

How Can We Grow the List of Recipients?

In SED04, we sent 305 emails. This time, we wanted more. Thus, we raided LinkedIn for more employee names and addresses, as described in [Section 4.1.2](#).

Combining the address list used in SED04 and the one gained from LinkedIn, we had a list featuring 681 names and email addresses, which was a substantial percentage of the total number of employees of [YuuBlue](#). There were some duplicates, and a certain amount of addresses turned out to be invalid (see [Times-tamp 11](#) on page 89), but still, the list was quite impressive.

How Do We Write Spam That Is Not Recognized As Such?

This question was the hardest to answer. We did not know how good the [YuuBlue](#) email infrastructure was when it came to spam.

Since [YuuBlue](#) is an IT company, the idea was to send them something that could genuinely interest at least some of the recipients, for example, an event with an exciting, computer-related topic that takes place near where [YuuBlue](#) is based.

To find such an event we googled for “events [Bern](#)”, clicked on the first result (which was computer-related) and took the description of the event as the body of our email. Because it was a real event, it would stand against any background check an employee might perform, without us requiring to do anything.

We tried our best to write a message that did not look like spam outright, but ultimately, all we could do was to experiment live on the hot target (see [Execution History](#) in [Section 10.2](#)).

How Can We Tell If Somebody Takes the Email Seriously?

Just as in SED04, we wanted to have a record of who read the email. For this reason, we decided to again include a personalized link in the spam email. This link just sent along the first 12 characters of the hashed email address of the recipient as a GET parameter. We surmised that whoever would click on the link had probably read the email.

We were aware that a link in the email might boost the spam score of our email, meaning that the classification of our email as spam might become more likely. However, we decided that knowing how many employees clicked the link, plus the operating system of whoever did, was worth the risk.

Services	Total Available	Used
Disk Space	1 GB	12 KB
Traffic	5 GB	≈ 5 MB
Domains	1	0
Subdomains	3	1
MySQL DBs	1	1
Email accounts	1	0
File Size Limit	15 MB	
SMTP	ON	
PHP version	7.0.30	
MySQL version	5.7	

Table 10.1: Awardspace Hosting Account Statistics

Where Can We Host a Statistics Page?

For the personalized link to work, however, we needed to host a web page somewhere. After the SED04 incident, we had to assume that [YuuBlue](#)'s IT center had blocked our server's IP address and domain. Thus, we could not use the same server anymore.

Luckily, many platforms offer a small amount of free web space to everyone. We registered an account with Awardspace (see Table 10.1 for information on the web space) and uploaded a PHP script (see Appendix A) that stored the exact time of the page visit, the visitor's operating system and IP address, as well as the visitor's hash as an identification mechanism in a database.

After inserting this data into the database, the script immediately forwarded to the destination the visitor was expecting. On a moderately fast internet connection, it is almost impossible to see that a redirect takes place. Even if the users were to notice the redirect, they would not get overly suspicious, because redirects have become standard, since companies like Google use them extensively.

How Do We Send the Emails?

To send the emails, we reused the Python script we had written for SED04 (see [Appendix A](#)). `sendmail` again served as the outgoing SMTP server.

We figured that the IT center would only block our server's IP address for traffic on ports 80 (HTTP) and 443 (HTTPS). If we were wrong and our emails were blocked as well, we could still try to use another server to send the emails.

10.2 Execution History

Timestamp 1

**Wednesday, 2018/04/25, 08:29:**

We find an event (see [How Do We Write Spam That Is Not Recognized As Such?](#)) about learning Python, a beginners course, offered by [BlurRed](#), a large, world-wide known IT company.

We create the email address `<blurred>.python@mail.com`.

Timestamp 2

**Thursday, 2018/04/26, 16:30:**

We look for a free webspace and quickly find Awardspace, where we create an account and upload our PHP logging script (see [Appendix A](#)).

Timestamp 3

**Thursday, 2018/04/26, 17:00:**

We send an email with the following text and sign it with the name Maria Stadelmann, a fictional person (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: Learn Programming with Python @ [BlurRed](#)**

Dear subscriber,

Do you want to learn to program? Then you are in the right place! In the course, we will introduce you to the programming language Python. We explain the most important data structures and build a small, fully functional program that you can take home with you. This course teaches the basics of pro-

programming and will also help you if you want to learn another programming language later. All you need is to bring your own laptop to the course.

The number of seats is limited. An `<BlurRed>` ticket is obligatory to attend the course.

Registration ends on Thu 31.5.

The course is open to all, regardless of gender, age, and work experience. The course language is German, but the tutors also speak English.

The meeting point is the `<BlurRed>` Hall. More information about the event can be found at <https://www.events.com/learn-python-blurred>

Kind regards,
Maria Stadelmann
`<BlurRed>`
Event Manager

”

In the original email, the link was personalized and pointed to our logging script on Awardspace first and then redirected to <https://www.events.com/learn-python-blurred> (censored link).

Timestamp 4

Thursday, 2018/04/26, 17:05:

We do not receive a single bounce or out-of-office message. Since SED04 provoked many bounces (for example, because of invalid email addresses in the list), our spam email should have produced some too. From the absence of bounces, we conclude that our email was considered spam, skipped the inboxes, and **went directly into the spam folder**.

Timestamp 5

Thursday, 2018/04/26, 17:30:

We discuss possible reasons for our high spam score. We had sent the email using `<BlurRed>` in the subject line and the **From:** header. It may be too generic since the name of a large company like that might be used all too often by spammers.

We remove `<BlurRed>` from the subject line, set the **From:** header to Maria Stadelmann, and **send the email again.**

Timestamp 6

Thursday, 2018/04/26, 17:35:

Once more, we do not receive any bounces. We are getting nervous. Have they blocked our server's IP address not only for HTTP and HTTPS but do they also filter our emails?

Timestamp 7

Thursday, 2018/04/26, 18:00:

We decide to give it one last try and call it a day.

It may be the peculiar sender address (`<blurred>_python@mail.com`) that causes the emails to be classified as spam. We decide to use a Gmail address that we had planned to use in SED02 (see [Section 5.1](#)): `maja.donatelli@gmail.com`. We change the signature of the email to match the new email address, and we renew the subject line entirely, in case `<YuuBlue>`'s system is intelligent enough to link our slightly changed attempts of getting past the spam folder.

The result is the following email (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: Beginner Programming Course at `<BlurRed>`

Dear subscriber

Do you want to learn to program? Then you are in the right place! In the course, we will introduce you to the programming language Python. We explain the most important data structures and build a small, fully functional program that you can take home with you. This course teaches the basics of programming and will also help you if you want to learn another programming language later. All you need is to bring your own laptop to the course.

The number of seats is limited. An `<BlurRed>` ticket is obligatory to attend the course.

Registration ends on Thu 31.5.

The course is open to all, regardless of gender, age, and work experience. The course language is German, but the tutors also speak English.

The meeting point is the `<BlurRed>` Hall. More information about the event can be found at <https://www.events.com/learn-python-blurred>

Kind regards,
Maja Donatelli
`<BlurRed>` Event Manager

”

Timestamp 8

Thursday, 2018/04/26, 18:05:

Sadly, we still do not receive any bounces. We routinely check our email queue and find it not to be empty. All of our emails have produced the following message at the last attempt:

“ 65029817D0 4062 Thu Apr 26 18:05:18
maja.donatelli@gmail.com
(host mail.yuublue.com[xxx.xxx.xxx.xxx] said:
451 4.7.500 Server busy. Please try again later from
[yyy.yyy.yyy.yyy].

(AS77712200) [mail.yuublue.com] (in reply to end of DATA
command)) `<mahatma.gandhi>`@yuublue.com

”

`<Mahatma>` was the recipient of this particular email.

The messages in the queue meant that `<YuuBlue>`'s email server was talking to us and that our server's IP address had not been blocked (at least not on all ports).

What does the message mean, however?

Timestamp 9

Thursday, 2018/04/26, 18:35:

After half an hour of googling the error code 451 4.7.500, we are confident that the messages are caused by an implementation of greylisting (see [Section 2.13](#)). All we had to do now was to wait for `sendmail` to attempt to deliver our emails again automatically.

Timestamp 10

Thursday, 2018/04/26, 19:30:

It was indeed greylisting. All of our emails have been delivered, and we are getting the expected bounces and out-of-office replies.

Timestamp 11

Saturday, 2018/04/28, 11:19:

It is time to see what we got.

Of 681 emails, 83 (12.18%) could not be delivered, and 52 (7.63%) were out-of-office. For the remaining 546 emails (80.17%) that were delivered, our PHP script registered 37 unique clicks (6.77%).

See [Figure 10.1](#) on page 90 for a graphical representation.



See [Avoid or Limit Out-of-Office Replies](#) on page 135.

We close SED12.

10.3 Findings and Post-Execution Insights

The twice as long list of email addresses rewarded us with almost twice as many out-of-office replies. The informational value of these replies cannot be stressed enough. They provide an enormous amount of information (see [Section 4.3](#)) and enabled us to execute many schemes (for example, SED09, SED11, SED13, SED15, and SEP01). This time, four C-level executives were out-of-office. We had struck gold.

Even the bounces were not worthless. The email bounces allowed us to refine our list of email addresses, removing the invalid ones, resulting in an extensive and more and more accurate enumeration of all the employees of [YuuBlue](#).

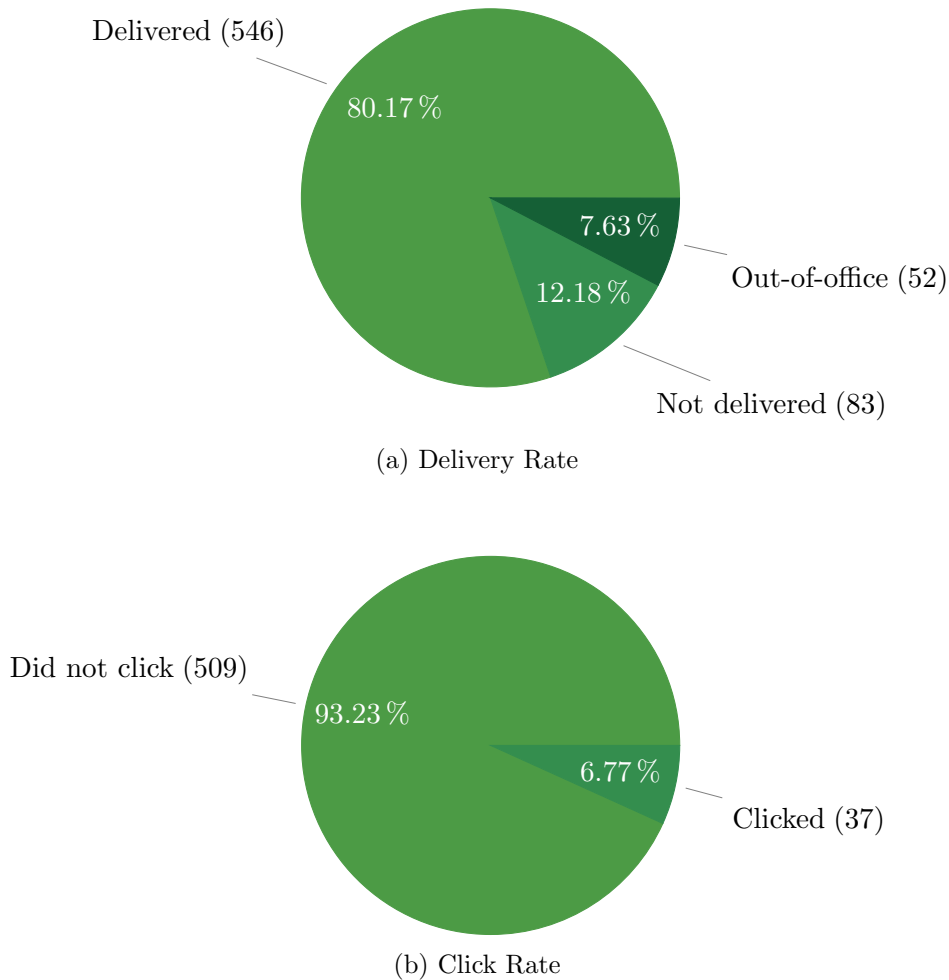


Figure 10.1: Statistics

The 37 employees who were curious enough to click the link are also worth mentioning. They provided us with information about their operating system, which could be used by an attacker in later, targeted attacks.

What is interesting about the course of this scheme is that we had a much harder time to get the email to the inbox than we had with our previous attempt in SED04 (see [Execution History](#) in [Section 6.2](#)). As we came to know later, greylisting was one of the measures that the [YuuBlue](#)'s IT center had implemented after the SED04 incident. It looked like our penetration tests had already borne fruit.

While SED04 had caused a commotion throughout [YuuBlue](#), we managed to execute SED12 entirely silent. No one thought of SED12 to be more than just another spam email. No one realized that it was a targeted attack, and no one was aware that more than two-thirds of the entire company just had received

the exact same email, or that beforehand, more than 1200 emails with the same content had gone to the spam folder.

We say this with confidence because, until the end of the project, nobody ever approached us with questions about SED12.

The stealthiness of this attack allows repeated execution whenever the attacker wants to know who is out-of-office. This way, **the attacker can “poll” the target company for information.**

SED13: The CV Crawler

The out-of-office replies resulting from SED04 (see [Chapter 6](#)) had enabled us to get information about salaries from current (see SED09 in [Chapter 8](#)) and former (see SED11 in [Chapter 9](#)) employees.

Since SED12 (see [Chapter 10](#)) had given us another round of out-of-office replies, we wanted to use these to get even more data. This time, we were after the CVs of applicants.

11.1 Preparation

The three important questions we had to answer were:

How Are We Going to Ask for CVs?

The idea was that the head of a department might need to see some CVs to hire someone new to enlarge the staff for a new project. Knowing this, we looked for suitable department heads.

Who Is Going to Be the Sender?

The out-of-office emails resulting from SED12 provided us with the out-of-office replies of eight department heads. One of them, [Galileo Galilei](#), seemed to be especially suitable to us, judging from his job description we had found on LinkedIn.

Whom Are We Going to Ask for CVs?

Applicants usually send their CVs to the HR department. Thus, we reused [hrdirect](#)@yuublue.com, the address that we had learned from [Marie Curie](#)'s out-of-office reply (see [To Whom Does William Send His Request?](#) in [Chapter 9](#)).

11.2 Execution History

Timestamp 1

Friday, 2018/04/27:

We choose `<Galileo Galilei>` as the “acting” person.

Timestamp 2

Wednesday, 2018/05/02, 10:00:

We create the email address `<galileo.galilei>@yahoo.com`.

Timestamp 3

Wednesday, 2018/05/02, 11:56:

We send the following email to `<hrdirect>@yuublue.com` (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: .NET Software Developer

Dear HR,

For an upcoming project, I am urgently looking for personnel reinforcement in the area of .NET.

The project is in the planning stage and will start soon, which is why I have to take care of it during the holidays (I don't have access to my `<YuuBlue>` emails at the moment).

Can you send me the CVs of the applicants in the area of .NET or software development in general?

Kind regards,
`<Galileo>`

”

See Mistrust Unusual Email Addresses on page 68.

Timestamp 4

Wednesday, 2018/05/09, 12:27:

Due to no response to the above email, we conclude that HR did not fall for the trick and close SED13.

11.3 Findings and Post-Execution Insights

We have not been able to extract any CVs from the HR department. The excellent reaction of the HR department is good for [YuuBlue](#) since, like in SED11 (see [Chapter 9](#)), data of people who were not (yet) related to the company was at stake.

The reason why HR had recognized the fraud as such was that it had been a “very strange request, coming from the wrong person”, as the person who had received the request told us later.

It seemed like we had picked not only the wrong person but also the wrong channel to make such a request. The HR team further told us that they had then proceeded with calling the real [Galileo Galilei](#) to double-check the facts.

Double-checking the facts is exemplary and precisely what one should do when in doubt about an email. It shows the effectiveness of this simple countermeasure. Double-checking the facts via a second channel that is not controlled by the attacker immediately exposes any fraud involving “private” email addresses.



See [Mistrust Unusual Email Addresses](#) on page 68.

SED15: The Customer Summary

Even though we were not able to extract any CVs in SED13 (see [Chapter 11](#)), we wanted to start one more “private” address scheme, involving a C-level executive.

So far we had tried to extract information about [YuuBlue](#)’s employees and applicants. In SED15, we wanted to get information about their customers and customer projects.

12.1 Preparation

Having already decided on a C-level executive [Winston Churchill](#) to place the request, the only questions left to ask were:

Who Is Going to Be the Recipient of [Winston](#)’s Request?

LinkedIn (see [Section 4.1.2](#)) proved once more to be an excellent source of information, allowing us to find [Winston](#)’s secretary [Marie Antoinette](#) by looking through the list of [YuuBlue](#) employees and their job descriptions.

What Is [Winston](#) Going to Request?

As mentioned above, we wanted to get information about [YuuBlue](#)’s customers and their projects. We were primarily interested in the scope and volume of the customers’ projects. That is why we decided to ask for a list of the ten most significant projects of 2017.

12.2 Execution History

Timestamp 1

Tuesday, 2018/05/01, 11:12:

We create the email address `<winston.churchill>@protonmail.com`.

Timestamp 2

Wednesday, 2018/05/02, 12:06:

We send the following email to `<marie.antoINETte>@yuublue.com` (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: Customer Projects**

Dear `<Marie>`,

Can you please put together a list of the ten top-selling customer projects for 2017? If possible, sorted by size and scope, as PDF or Excel. I urgently need the document for a web conference tomorrow, but unfortunately I do not have access to my `<YuuBlue>` emails (therefore I'm writing with my private email address).

That would be very nice of you.

Greetings,
`<Winston>`

”



See **Mistrust Unusual Email Addresses** on page 68.

Timestamp 3

Wednesday, 2018/05/02, 12:17:

Only ten minutes later, we receive the following, motivated answer from `<Marie>` (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: Customer Projects

Hello [Winston](#),

I'll have a look at it with [Florence](#) this afternoon. I was already at her's and [Niccolò M.](#)'s office. We can do that.

I'll get back to you afterward.

Many regards,
[Marie](#)

”

[Niccolò Machiavelli](#) was the CFO of [YuuBlue](#). He was running a very tight ship, with proper processes in place and well-trained employees, as SEP01 (see [Chapter 16](#)) had shown.

Putting together such a list takes time, and it made sense that [Marie](#) had to gather the data in cooperation with Finances. Even though we had no idea who [Florence](#) was, we surmised that she might be part of [Niccolò](#)'s team.

Timestamp 4

Wednesday, 2018/05/02, 14:58:

We send a polite answer and remind [Marie](#) that it is imperative to use the right email address, to instill a sense of urgency (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: RE: Customer Projects

Hello [Marie](#),

Thank you very much!

Kind regards,
[Winston](#)

PS: As I said, please send the list to this address because I have no access to my [YuuBlue](#) emails at the moment. Thank you!

”

Timestamp 5

Wednesday, 2018/05/09, 16:34:

Due to no response to the above email, we conclude that something had tipped [Marie](#) off. She may have double-checked the facts with the real [Winston Churchill](#).

We close SED15.

12.3 Findings and Post-Execution Insights

We were unable to extract information about [YuuBlue](#)'s customer projects. When we asked [Marie](#) for her side of the story after the project was finished, she told us that [Florence](#) was immediately suspicious of the request, especially since a private email address was involved. This suspicion had then caused her to double-check the facts with [Winston](#).

[Florence](#) was indeed part of [Niccolò](#)'s Finance team and was thus well-trained to spot frauds and scams.

It is possible that SED15 would have worked if we had asked a question that did not require anyone from Finance to be involved. [Marie](#) herself, namely, had not gotten suspicious and was planning to carry out the request.

Our urgent, almost desperate sounding request in the postscript of the second email (see [Timestamp 4](#)) might have been over the top and thus suspicious as well.

SED16: Attack from Within

SED16 itself was not an attack, but rather the preparation to SED18 and SED19 (see [Chapter 14](#) and [Chapter 15](#)).

We had been doing quite a few schemes involving the “private” addresses of employees who were out-of-office since it is a convenient way to impersonate somebody. However, since our last two schemes had failed (see SED13 in [Chapter 11](#) and SED15 in [Chapter 12](#)), it seemed like word had started to get around that trusting emails from external sources is not always advisable and can have severe implications if sensitive data is sent to the wrong person.

That is why we wanted to try a different approach. Since everybody was now suspiciously reading every email originating from external sources, we wanted to demonstrate that it was possible to send emails that seemingly came from within [YuuBlue](#), by merely faking the **From:** header of the email. The idea was that because everybody was so suspicious when it came to external emails now, they would blindly trust emails originating within [YuuBlue](#).

13.1 Preparation

Sending an email that was seemingly coming from within [YuuBlue](#), however, turned out to not be as easy as spoofing the **From:** header, because [YuuBlue](#) was employing a Sender Policy Framework (see [Section 2.14](#)). We learned about [YuuBlue](#)’s SPF by looking at the logs of our email server.

How Do We Get into the [YuuBlue](#) Network?

SPF meant for us that we could not send an email on behalf of the [yuublue.com](#) domain while being outside of their network. Lucky for us, we had obtained the wireless network password for one of [YuuBlue](#)’s wireless networks in SEP03 (see [Chapter 18](#)).

Where Do We Access the Wireless Network?

Another problem we had to solve was where we could get access to the wireless network. Outside of the main building, there was no connection possible because there was no signal.

⟨YuuBlue⟩ had, however, a second building that it shared with some other companies. Because the building was shared, it meant that the entrance hall, as well as the staircase, were publicly accessible. We used this public access to our advantage and climbed the staircase to one floor above the ⟨YuuBlue⟩ floor, where the signal was strong enough to log into the ⟨YuuBlue⟩ wireless network.

What Email Server Are We Going to Use?

Since we were equipped with a Windows machine, we used the *hMailServer*¹, a lightweight email server for Windows.

13.2 Execution History

Timestamp 1



Tuesday, 2018/05/08:

We install and configure the email server.

Using the Python email script created for SED04 (see [Appendix A](#)), we test the email server at home by sending a few emails to ourselves.

We test how strongly Gmail and other email providers react to different types of attachments and to changed Reply-to headers, to get an idea about how far we can go with the ⟨YuuBlue⟩ email server, without getting flagged as spam.

Timestamp 2



Wednesday, 2018/05/09, 14:00:

We arrive in ⟨Bern⟩ and try to get a network connection without entering the building. Unfortunately, the signal is not strong enough.

Timestamp 3



Wednesday, 2018/05/09, 14:15:

We enter the building with the public staircase and climb to the third floor, one floor above the ⟨YuuBlue⟩ floor. There, we have a signal strong enough to connect. We enter the password and are in the ⟨YuuBlue⟩ network.

¹<https://hmailserver.com>



See [Have the Courage to Address People](#) on page 137.

Timestamp 4

Wednesday, 2018/05/09, 14:30:

We start to send emails to non-existing [yuubluе.com](#) addresses, to test if the [YuuBlue](#) email server processes our emails. If they are processed, we should receive a bounce.

We send the emails to non-existing addresses so that no one will see them. If the IT center should have the time to look at the emails that created bounces, they would only see emails with spam-like texts and discard them as failed spam attempts, nothing more serious.

Timestamp 5

Wednesday, 2018/05/09, 15:30:

None of the emails we sent created any bounces. We have a look at the log and see that the sent emails never left our system.

We send another email to a Gmail account of ours. The email arrives, and we compare the log files.

From the log files we learn the reason why none of our emails leave the system: when we had configured the local email server, we had chosen [yuubluе.com](#) as the local domain (since we wanted to send emails on behalf of this domain). It turns out, though, that this had been a mistake, because the email server never sends an email intended for [yuubluе.com](#), since it believes to be [yuubluе.com](#) itself, meaning from the server's perspective, the email had already reached its destination.

Timestamp 6

Wednesday, 2018/05/09, 16:00:

We reconfigure the email server with a different domain and resend one of our spam emails to create a bounce.

It finally works. We can now send emails on behalf of [yuubluе.com](#) while being in the [YuuBlue](#) network.

Satisfied, we close SED16.

**Mitigation 6****Secure Your Domain**

The simple mail transfer protocol (SMTP) was not designed with security in mind and does not offer methods to verify authenticity or integrity of emails, making it easy for attackers to send emails on behalf of any domain, to change the content of emails (e.g., as part of a Man-in-the-Middle attack, see [Section 2.5](#)), or to simply read the emails intended only for the eyes of the recipient [43].

Over time, various protocols have been developed that can be used alongside SMTP and that allow authenticating the sender of an email [43]. Three famous ones are SPF, DKIM, and DMARC (see [Section 2.14](#)).

Use a combination of the protocols mentioned above. The correct configuration of the protocols is paramount. Guides on how to set up SPF and DKIM DNS entries can be found in the literature [55, 56].

Exceptions for IP addresses should only be used if necessary, and the number of IP addresses from which it is allowed to send emails on behalf of your domain should be kept at a minimum.

Test your implementation of SPF, DKIM, and DMARC by trying to abuse an internal email address from outside your network (computers in the guest network should not be allowed to send emails on your domain's behalf either, because too many people have access to the guest network).

Securing one's domain is a continuous process. Just as attack vectors change, email protocols may develop and change over time, requiring actions on the part of the users and providers.

13.3 Findings and Post-Execution Insights

When we were testing how strongly different email providers react to spam, Gmail was by far the most rigorous. It seemed that if we set a `Reply-to` header, Gmail always flagged an email as spam. The same went for attachments consisting of executables and executables in unencrypted archives. The other email providers reacted less rigorously, but as well did not accept any executables, be it in an archive or not.

Only encrypted archives had a chance of passing the email server. However, we decided that the chances of a target downloading the attachment, entering the password, executing the payload, *and* the payload not getting recognized by the installed antivirus, were close to zero, especially since Windows does not handle encrypted archives natively for all versions², which would require some users to install additional software first. It meant for us that we had to further concentrate on social-engineering schemes and forego the keylogger that we had written.

So far, all our schemes had tried to exfiltrate some information by provoking the target to answer an email. Because this time we were faking the sender address, an answer would go to the person we were impersonating. That is why we tested the `Reply-to` header, with which it would have been possible to set an alternative reply address. We decided not to use the header, because the target will see the alternative address when hitting “Reply”, which we deemed to be too suspicious.

This caveat meant for us that we had to come up with email texts that did not provoke an answer. We could only feed information, and we would not get any feedback. If the target sent an answer, the scheme would immediately be exposed because the real person would inform our target that the sender of the email at hand was somebody else.

Lastly, our problem-free, extended stay in the staircase of the side building surprised us. The next working space was only a (badge-protected) door away so that no employee in his or her right mind would sit in the cold and inconvenient staircase. Of all the people that passed us during the two hours that we spent in the staircase of that building, no one talked to us, and no one asked us what we were doing. We were sitting on the stairs with a computer on our lap: it does not get any more suspicious (the baseball caps that we were wearing to hide our faces from the camera in the entrance hall did not make it any better).

We believe that no one dared to talk to us because no one knew many people in the building. Several companies had rented space in the building, including a floor that was used exclusively by start-ups, where the set of faces changed regularly, due to the fast-pacing nature of start-ups.

²Windows 10 Home does, to our knowledge, not offer native encryption, while Windows 10 Pro offers Encrypting File System (EFS). We do not know if EFS can be used to encrypt archives.

SED18: The Flash Mob

In SED16 (see [Chapter 13](#)) we tested our capability to send emails on behalf of the [yuubluе.com](#) domain. For reasons shortly discussed in [Section 13.3](#), we could not exfiltrate any information this way. Thus, the goal was to send an email that provokes an action but does not require any further communication.

The idea was to invite people to a meeting that never takes place. If employees turn up to that meeting, it costs them and thus the company time. If an employee had been required elsewhere at the time of this meeting, we would have successfully executed a Denial of Service attack.

14.1 Preparation

The only two questions were:

Who Are We Going to Use As the Host?

As the host, we needed to use somebody who had enough credibility to convene a meeting. The [Yuubluе](#) CEO, [Julius Caesar](#), fulfilled this requirement.

Who Are We Going to Invite?

With each recipient of the invitation, the chances rose that somebody would reply and thus expose the scheme. That is why we limited the number of invitees to 16 department heads and C-level executives whose names and job positions we found on the [Yuubluе](#) website or in our extensive list of employees that we got from LinkedIn (see [Section 4.1.2](#)).

14.2 Execution History

Timestamp 1

Friday, 2018/05/11, 10:41:

We take the Python email script we had written for SED04 and rewrite it to fit SED18, namely to send only one email to multiple BCC recipients (see [Appendix A](#)).

Timestamp 2

Friday, 2018/05/11, 11:15:

We create the list of recipients, totaling in 16 employees.

Timestamp 3

Monday, 2018/05/14, 11:19:

We position ourselves in the staircase of the [YuuBlue](#) side building (just as we had done in SED16 in [Chapter 13](#), see [Timestamp 3](#) on page 100) and send the following email to all the recipients in the list, using [julius.caesar@yuublue.com](#) as the sender email address (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: Toast to New Challenges

Dear Friends,

I would like to invite you to come to the entrance hall of the main building and to toast with me.

After a few words from me, there will be a little surprise.

Afterward, for all those who like to, we can have lunch together in the [Canteen](#).

I look forward to numerous appearances.

Kind regards,
[Julius](#)

”

Initially, we had planned to set the meeting *in front* of the main building. Unfortunately, the weather was terrible on the 14th, which is why we settled to invite the targets to the entrance hall of the main building. We assumed that we would

still be able to see what was going on from outside, since the entrance hall was directly behind a large glass front.

The “little surprise” in the text was supposed to be that the CEO was not going to come to the meeting which he himself had convened.



See **Secure Your Domain** on page 101.

Timestamp 4

Monday, 2018/05/14, 11:25:

We race from the side building to the main building to observe, from outside, how many people followed the invitation.

The plan had been to position ourselves relatively inconspicuously on a side street. Unfortunately, due to the lighting situation, the glass front of the main building is too reflective, and we cannot see inside at that angle.

We start walking around to see if we can get a better angle.

Timestamp 5

Monday, 2018/05/14, 11:50:

With only ten minutes left, we improvise. On the opposite side of the street, there is an apartment block with a staircase that faces the entrance hall of the main building directly.

We position ourselves in front of the entrance to that block around the corner of the street and pretend to read the numerous mailbox labels. Sure enough, the first person entering the apartment block asks us if we would like to enter as well. We thank him profusely and tell him that a friend of ours had just recently moved here.

While the person is taking the elevator, we walk to the staircase and enjoy the view of the entrance hall of the main building.

Timestamp 6

Monday, 2018/05/14, 12:00:

Four people show up to toast with the CEO.

We take some photos and close SED18.

14.3 Findings and Post-Execution Insights

Even though we sent the email only 40 minutes before the actual meeting was supposed to take place, the IT center managed to warn everybody. As we came to know later, a member of management replied to the email, asking if it was real. This reply immediately alerted [Julius](#), who in turn notified the IT center, because he could not figure out all the recipients of the email himself since everyone had only received a BCC copy.

The IT center managed to recover all the recipients in record time and, just in time on 12 o'clock, sent the following email to all the recipients of the fake invitation (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: Fraud mail: Toast to New Challenges**

Hello everybody,

the email from [Julius Caesar](#) this morning (14/05/2018 09:19:43) is a false report.

Apparently, someone has managed to misuse [Julius](#)' email address.

[@Einstein, Albert](#) is informed to take further steps.

Best regards,
IT helpdesk

”

Note that, in a hurry, they had misspecified the time the email had supposedly been sent. [Albert Einstein](#) is the group-wide CISO of [YuuBlue](#). [Albert](#) was one of the few people who knew about project SOEP and did, therefore, not “take further steps”.

The question remains what would have happened if we had invited the entire company. We intentionally limited the number of recipients to minimize the risk of somebody alerting the real [Julius](#) (unfortunately, this happened despite our precautions). However, even though the IT center had been informed, four people had shown up. If we had sent the email to 160 people, instead of 16, then maybe 40 people instead of only four would have shown up. What had only required us to add more recipients to the list, would have resulted in a much more significant loss of time for [YuuBlue](#). We do not know if the IT center's process to recover the BCC recipients of the email was automated or if they had to do it manually. In the manual case, a ten-fold increase of recipients would have successfully prevented them from reacting in time and might have provoked

a company-wide warning email that would have irritated many employees.

Inviting some employees to a fake meeting is admittedly not a very imaginative or dangerous attack. However, from the beginning, SED18 only served to show the possibilities and to demonstrate that we were able to send emails on behalf of yuublue.com.

The power that we held in our hands was significant. Imagine we had sent the entire company an embarrassing text in the name of the CEO. Such an email can be profoundly defamatory, even if it is later discovered to be false. The damage is already done and hard to repair.

Alternatively, and even worse, we could have harmed the reputation of the entire company towards the outside world by sending an email to a news outlet on behalf of [YuuBlue](#), talking about a topic like environmental disregard or bankruptcy. We could have also, in any way, offended a customer. Since even the IP address is the right one, it is hard to prove that the email was fake, especially since a news agency may not want it to be fake.

We did not want to harm [YuuBlue](#) seriously. **A real attacker, however, is not bound by any contracts or a bad conscience.**

Aside from the results of our little ruse, we found it to be most interesting how easy it was getting into the apartment block on the opposite side of the road. Tailgating (see [Section 2.1.6](#)) works and is quite simple to pull off, even though it required some luck to encounter a person who entered the building in the few minutes that we needed it to happen. However, if time is not an issue, tailgating is a sure way to get inside most buildings.

SED19: Orders from Above

SED19 was our final attempt to trick [YuuBlue](#) into paying us money (the other two attempts being SED11 in [Chapter 9](#) and SEP01 in [Chapter 16](#)).

Since SED16 (see [Chapter 13](#)) had given us the power of sending emails on behalf of [yuubluе.com](#), we wanted to try to give a direct order to see an invoice paid.

15.1 Preparation

The list of questions for this scheme included two relatively easy questions and one hard one, in this order:

Who Would Give the Order?

Just as in SED18, the man with the most authority was, as we believed, the CEO of [YuuBlue](#): [Julius Caesar](#).

Who Would Execute [Julius](#)' Order?

[Julius](#) would probably send the order to pay an invoice to his secretary. Since [Marie Antoinette](#)'s (whom we know from SED15 in [Chapter 12](#)) job description in LinkedIn stated that she was the secretary for the entire management, she was the perfect person to be entrusted with our request.

What Kind of Invoice Is Plausible?

The hardest question to answer in this scheme was, what kind of invoice would come directly from the CEO. We were thinking about this for the better part of a day until we had the following idea:

⟨YuuBlue⟩ was well-known for a project they had done together with another well-known Swiss company. This project, ⟨Project X⟩, could be described as one of ⟨YuuBlue⟩’s prestige projects. As CEO, ⟨Julius⟩ had made significant contributions to the success of ⟨Project X⟩, which is why it must be quite valuable to him.

⟨Project X⟩, so we thought, was so crucial for ⟨YuuBlue⟩ and its CEO, that ⟨Julius⟩ had ordered the logo of ⟨Project X⟩ to be carved in stone and to be displayed in the front yard as a human-sized statue.

The statue may sound ridiculous, but the idea is not unprecedented to do this (see Google’s Android statues).

15.2 Execution History

Timestamp 1



Friday, 2018/05/11, 10:54:

We adjust the Python email script from SED04 to be able to send attachments (see [Appendix A](#)).

Timestamp 2



Saturday, 2018/05/12, 10:00:

We create an invoice for a first draft of the statue by a non-existing sculpture and add our bank account as the receiving account.

Timestamp 3



Monday, 2018/05/14, 11:21:

We send the following email in ⟨Julius⟩’ name to ⟨marie.antoINETte⟩@company.com (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: ⟨Project X⟩

Hello ⟨Marie⟩,

In order to present our prestige project, ⟨Project X⟩, even better, ⟨YuuBlue⟩ commissioned a sculptor to ”carve the ⟨Project X⟩ logo in stone”, so to speak.

Today, I have already received the artist’s first design. I am thrilled!

Can you please take care of the attached invoice and release it in the [Application for processing financial transactions](#)? However, please do not post it to the project budget, but as a marketing measure, as it is only for our employees.

Thank you very much!

Kind regards,
[Julius](#)

”

The [Application for processing financial transactions](#) was an internal tool used by [YuuBlue](#) to record invoices. We had learned of its existence during SEP01 (see [Chapter 16](#)).



See [Secure Your Domain](#) on page 101.

Timestamp 4

Tuesday, 2018/05/15:

Due to no response to the above email, we conclude that [Marie](#) did not fall for the trick and close SED19.

15.3 Findings and Post-Execution Insights

Since we never heard back, we assumed that the scheme had failed. We later talked to [Marie](#) and she said that when she told the real [Julius](#) about the [Project X](#) statue, the scheme was immediately exposed. Once more, double-checking the facts prevented an attack on the company.

Thus, we remain unable to extract any money from [YuuBlue](#).

SEP01: The Fake Invoice

SEP01 was our first attempt at having money paid to a bank account of ours.

SEP01 was based on the idea that for a company of a certain size, it may be more expensive to fact-check an invoice than to simply pay it if the amount charged was small enough.

16.1 Preparation

The following questions arose:

Who Could Send an Invoice to [YuuBlue](#), And for What Reason?

Since we did not know any specifics concerning the projects and orders that [YuuBlue](#) was working on, we had to write an invoice that was so generic, that it could belong to any project.

The invoice we ended up with (see [Section B.3](#)) charged [YuuBlue](#) with CHF 620 for four hours of “Project support”. We did not elaborate further on what kind of support was meant. We signed the invoice with the name of a fictional freelancer, Jonathan Steiger.


We did not include any contact details other than a postal address, because we wanted to make it as cumbersome as possible for [YuuBlue](#) to get any more information about the invoice, in the hope that they would pay it. Sending a letter via the postal service was their only option to make contact, and it meant that we could think twice about what we were going to answer (as opposed to, for example, a phone call, where we would have to react immediately).

What Postal Address Are We Going to Use?

For reasons of convenience, we added our residential address to the invoice. We added a label with the name of Jonathan Steiger to our mailbox to guarantee that we would get the reply letter that [YuuBlue](#) might send.

If anonymity is of importance, the own residential address might be problematic. Alternatively, we could have used the address (and the name) of a neighbor. However, choosing a different mailbox requires to intercept the postal service every day until the reply letter is received. Intercepting the postal service can either work by chatting with the mailman and “taking it from here”, or by poking around in the mailbox after the mailman has left and before the owner of the mailbox empties it. Both options are illegal and we did not want to involve any third parties.

16.2 Execution History

- 
- Timestamp 1 **Thursday, 2018/04/05, 10:20:**
We create the fake invoice, containing our residential address and our bank account.
- Timestamp 2 **Thursday, 2018/04/05, 11:30:**
We add a label “Jonathan Steiger” to our physical mailbox at home.
- Timestamp 3 **Thursday, 2018/04/05, 16:25:**
We send the fake invoice via the postal service to the address published as the primary address on the website of [YuuBlue](#).
- Timestamp 4 **Monday, 2018/04/16:**
The mailman inquires if Jonathan Steiger was now living with us. We had planned for questions like this and answer that Jonathan Steiger had moved away for a semester abroad and that we allowed him to use our address to receive letters.
The mailman does not like the answer at all and tells us to “remove the label if no Jonathan Steiger is living here”. Despite the mailman, however, and since

we did not yet receive an answer from [YuuBlue](#), we decide to leave the label there for at least another month, at the risk of upsetting the mailman.

Timestamp 5

Tuesday, 2018/04/17:

Fortunately, the very next day we receive a reply from [YuuBlue](#), signed by [Indira Gandhi](#) of the Finance department (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Invoice No. 1652-5463

Dear Mr. Steiger,

In the supplement you can find your invoice to [YuuBlue](#) in [Bern](#). Unfortunately, we could not find the client for your project support (altogether 4 hours) in our company.

Neither email address nor telephone number could be taken from your receipt. We ask you to send us an invoice with the full name of the client or to contact us by phone.

Kind regards,
[YuuBlue](#)

[Indira Gandhi](#)
[igh](#)@yuublu.e.com
[Direct Phone Number](#)

”

Sent along this inquiry was a copy of the fake invoice we had sent to [YuuBlue](#).

Timestamp 6

Thursday, 2018/04/19, 07:55:

Thanks to SED04 (see [Chapter 6](#)) which we executed in parallel to SEP01, we now know that [YuuBlue](#)'s CFO [Niccolò Machiavelli](#) is on vacation.

With the thought in mind that there might be **top-secret projects** of which only C-level executives might know, we create the email address [niccolo.machiavelli](#)@yahoo.com and send the following email to [igh](#)@yuublu.e.com (the original text was written in German, it is translated here for purposes of

reaching a broader audience):

“ Subject Line: Confidential Project

Hi [⟨Indira⟩](#),

Jonathan Steiger has contacted me regarding an invoice that you – correctly – rejected for lack of details. Since the invoice is about a strictly confidential project, I ask you to pay the bill within week’s end.

I will explain the exact circumstances as soon as I get back to the office (I don’t have access to my emails at the moment. Therefore, I use my private email address).

Kind regards,
[⟨Niccolò⟩](#)

”

By sending the email to [⟨igh⟩@yuublue.com](#) instead of [⟨indira.gandhi⟩@company.com](#), we try to establish a level of trust because we prove knowledge of [⟨Indira⟩](#)’s company-internal abbreviation, which no external entity can know (or so we assumed).

Because the CFO is so high up in the hierarchy, we thought that he probably does not need to specify to which project the invoice belongs, and because the project is “top-secret”, [⟨Indira⟩](#) is not likely to ask any further questions.



See [Mistrust Unusual Email Addresses](#) on page 68.

Timestamp 7

Thursday, 2018/04/19, 08:50:

Shortly afterward, we receive [⟨Indira⟩](#)’s answer (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ Subject Line: RE: Confidential Project

Hello [⟨Niccolò⟩](#),

Okay, I’ll take care of that. I have loaded the bill into

⟨Application for processing financial transactions⟩, but I have found no one to confirm the bill. I was going to contact this Jonathan by phone or email, but I had no information at all. The only way was to write him a letter.

In the meantime, I have removed the invoice from ⟨Application for processing financial transactions⟩. I will transfer the money with value date 2018/04/20.

Best regards,

⟨Indira⟩

”

Timestamp 8

Thursday, 2018/04/19, 11:16:

We politely reply with the following email (the original text was written in German, it is translated here for purposes of reaching a broader audience):

“ **Subject Line: RE: RE: Confidential Project**

Hello ⟨Indira⟩,

Thank you very much and see you next week.

Kind regards,
⟨Niccolò⟩

”

Timestamp 9

Tuesday, 2018/04/24:

Unfortunately, to this day the money has not arrived. We close SEP01.

16.3 Findings and Post-Execution Insights

The reply letter that we received from [YuuBlue](#) provided us with a handle in the Finance department. Every direct handle is valuable and could potentially be used for further, targeted attacks. What is more, the email address of [Indira Gandhi](#) gave away that [YuuBlue](#) used two kinds of email addresses: first, there are the email addresses with the usual `firstname.lastname@yuubluе.com` pattern (see [Section 4.1.1](#)), and second there are email addresses with a `shorthand@yuubluе.com` pattern. We surmised that the shorthand email addresses were predominantly used for internal email conversations. Knowing [Indira](#)'s shorthand may have given us a boost of trust when we wrote the email to her.

What is more, we learned about a program, system, or process that [YuuBlue](#) uses to process invoices. We used the knowledge of this system in [SED19](#) (see [Chapter 15](#)) in [Timestamp 3](#) on page 110, to gain the trust of the CEO's secretary.

SEP01 was the first of three attempts to exfiltrate money from [YuuBlue](#) (see [SED11](#) in [Chapter 9](#) and [SED19](#) in [Chapter 15](#), for the other two attempts). Unfortunately, the scheme failed even though we received a motivated and positive answer from [Indira](#) to our first email.

After we had been waiting for the money for nearly a week, we contacted the CFO of the company, the real [Niccolò Machiavelli](#), told him about project SOEP, and asked him what had happened. He told us that, in the Finance department, an extremely well-implemented four-eye principle forces every payment to be reviewed by a second person. It seems like this security measure had ultimately stopped the fraudulent payment because the second person to review our invoice and the email conversation between "[Niccolò](#)" and [Indira](#) became suspicious and informed the real [Niccolò](#) about the invoice, who naturally knew nothing about either the conversation or the invoice.



See [Mistrust Unusual Email Addresses](#) on page 68.

Even though we came incredibly close to being paid an invoice for absolutely no services, the Finance department's security measures prevented any money from being paid. These security measures, namely the **four-eye principle**, could serve as an example for other departments and companies.

SEP02: The Security Interview

In SEP02, we concentrated on asking employees questions about their passwords, without them knowing what we were after. SEP02 was part of SEP03 (see [Chapter 18](#)) and was executed while we were in [YuuBlue](#)'s main building.

The goal of the scheme was to find out how much information we could gain by simply asking people for it.

17.1 Preparation

The big question was:

What Questions Are We Going to Ask? How Are We Going to Ask These Questions Without Appearing Too Suspicious?

What interested us, in the end, were the passwords of the employees. However, we could not just ask them for their passwords; it would have been too obvious.

Instead, we thought about disguising our questions as a survey about computer security. Since [YuuBlue](#) is an IT company, we expected people to be interested in the topic and maybe spare a few minutes to answer our questions.

Because the best lies are those that are true at the core, we told the people we wanted to interview that we were doing a Master's thesis about computer security and were, therefore, surveying how people deal with the subject on a daily basis.

The final version of the survey can be found in [Section B.4](#). We started slowly by asking innocuous questions to build trust, and then, in the middle, we tried aggressively to obtain information about the interviewee's password.

17.2 Execution History

The exact date is redacted to protect the identity of the target company.

Timestamp 1

⟨Monday⟩, 2018/⟨01⟩/⟨01⟩:

The author is inside ⟨YuuBlue⟩'s main building and asks about ten people if they have time for a short survey.

Three of them say yes, and the author sits down to have the interview. *All* of the three employees answer *every* question to the last detail (see all questions in Section B.4).

After having interviewed the third person, we have to abort SEP03 (see [Timestamp 8](#) in [Chapter 18](#)), and consequently close SEP02.



Mitigation 7

Disclose Information Prudently

If someone asks you for information, think twice whether that person needs to have this information.

If you think that your conversation partner **is not entitled to specific information, politely tell him or her** that you would rather not disclose the information requested.

If you conclude that your conversation partner is in the right to have the requested information, think about how to **transmit it securely**. Encrypt it if the conversation takes place online, or take your conversation partner to a more quiet place with less prying ears and eyes.

17.3 Findings and Post-Execution Insights

While the initial goal had been to interview at least 30 people, we had to settle for only three interviewees because of the abortion of SEP03 (see [Timestamp 8](#) in [Chapter 18](#)).

Despite the small number of people questioned, we learned some interesting information.

- It seems like ⟨YuuBlue⟩ does not enforce any policies when it comes to how often the passwords of its employees have to be changed. This information is in so far interesting to us as it tells us that a lot of the employees probably

use the same password for an extended period and rarely change it, which means that any password we get our hands on may be valid for a long time.

- By asking questions of the operating systems and antivirus programs the employees use, we get an idea of what software is in use at [YuuBlue](#). This information allows for targeted attacks against specific employees.

Even though none of the interviewees write down their passwords, eventually, we maybe would have found somebody who does and could have subsequently tried to get such a password by searching in the right place.

The most exciting part of the survey, the part about the password mainly used at [YuuBlue](#), was answered in great detail by every single interviewee. We were quite surprised to hear people tell us the exact length of their passwords, the number of lower characters, upper characters, special characters, and digits, and what procedure they had used to come up with the password. With all of this information, the space of possible passwords is vastly decreased and may give enough clues to guess the password.

No one was in the least suspicious about these questions and was happy to answer. We do not know if the answers the interviewees gave are correct, but they did not sound like they were lying.

When it came to the questions about social engineering, one of the interviewees became restless as if it dawned on him that he was being social-engineered at the very moment. The other two, however, were not worried at all.

Furthermore, what we were excited to test was how the people would react to us asking them about their contact details at the end of the interview, despite having said at the beginning that the interview was anonymous. To reassure them of their anonymity, we ripped off the last page of the stapled survey and told them that their anonymity would be respected.

Every one of the questioned employees then consented to give us their email address, without hesitation. It seems like they did not realize (or did not care) that giving us their email address canceled their anonymity, even though we promised to respect it. They were trusting us, despite the fact that they had never before seen us.

Even though we only interviewed a small number of employees, SEP02 showed that people treat even important information like passwords quite liberally. They do not immediately (if at all) perceive a threat when being asked about such information and love to share information with strangers because they want to be of help, or because they enjoy that somebody takes an interest in them. Straight out asking for information can be the easiest way of obtaining data that is otherwise impossible to get.

SEP03: The Anniversary

The goal of SEP03 was to enter `<YuuBlue>`'s main building without a badge.

The entrance hall of the main building is located in a way so that we could observe it from the street. The entrance hall consists of a wardrobe and a waiting area where guests can wait for their contact person in the company. This waiting area is directly facing the reception desk, where three employees were working that day. At the far end of the entrance hall, there is a staircase next to the elevators.

To get into the building, we had to somehow pass the reception desk without any of the receptionists noticing us. As far as we could see, there were no gates that would require a badge.

18.1 Preparation

Even though we knew how the entrance hall is looking, the following questions remained unanswered:

How Do We Pass the Reception Desk Unobserved?

Passing the reception desk unobserved with the entrance hall being empty seemed to be impossible since three employees were working at the desk. What we needed was a crowd to blend in. Such crowds entering the building at once usually form in the morning, when everybody is commuting to work.

Thus, the plan was to walk into the building along with other employees.

On What Day Should We Enter the Building?

The more faces the reception desk sees during a day, the less likely it is that they remember a single one. We wanted to choose a day where there was more activity than usual.

Some time ago on [⟨PurpleGroup⟩](#), a social network where employees and applicants can write about their experience with companies (the name of the social network is not printed to protect the identity of [⟨YuuBlue⟩](#)), we had found a former employee of [⟨YuuBlue⟩](#) who talked about a so-called [⟨YuuBlue Day⟩](#), a regular meeting of all employees of [⟨YuuBlue⟩](#) (see [Section 4.6](#)):

On the $\langle n^{\text{th}} \rangle$ day of each month, so the former employee wrote, [⟨YuuBlue⟩](#) would arrange presentations, talks, and an aperitif. He went on to say that often on this day, he saw the faces of people who would otherwise not work at the [⟨YuuBlue⟩](#) main building, but instead were with customers. The [⟨YuuBlue Day⟩](#) seemed like an excellent opportunity for us to slip into the building because many less known faces could be seen on that day and thus we would arouse less suspicion.

However, then came along an even better opportunity. A few weeks earlier, we had created a fake account in the [⟨YuuBlue⟩](#) career pool, where applicants can upload their CVs and apply for a job. To the email address we had used at the registration, [⟨YuuBlue⟩](#) sent a newsletter on the occasion of its anniversary, saying that [⟨YuuBlue⟩](#) was founded exactly $\langle x \rangle$ years ago to the day. We thought that, on the anniversary, the main building might be even more crowded than on an average [⟨YuuBlue Day⟩](#). We spontaneously (within minutes of the anniversary email arriving) decided to attempt to start SEP03 a few days early (the anniversary took place a few days before the [⟨YuuBlue Day⟩](#)) and use the [⟨YuuBlue Day⟩](#) as a fallback, in case the anniversary did not generate enough tumult.

Where Do We Go after We Have Successfully Entered the Building?

We could only see the entrance hall while standing on the street in front of the main building. The person to be infiltrated did not know where to go once the hall was passed and the staircase reached.

While googling around to find a map of the building (which was impossible to find), we encountered a video on YouTube that was promoting a product that had been developed by [⟨YuuBlue⟩](#). The developers proudly presented the product using a camera-mounted drone which they started in front of the main building. They then proceeded by flying it into the entrance hall and up the staircase. They went on to scout different parts of the building with the drone (see [Section 4.4](#)).

Even though the entire video was a time-lapse, playing it five times slower was enough to give ourselves an idea of how the interior of the building looked like and where we could go. Orientation is a big help when trying to look like one belongs.

What Are We Going to Do If Somebody Asks Questions?

In case somebody gets suspicious of the person we infiltrated and approaches to ask questions, we wanted to be at least minimally prepared for the most likely questions they were going to ask. Since it is easiest to lie if at least part of the story (the core of the lie) is genuine, we prepared the following, minimal cover story:

The person we were going to send into the building was a student working on a Master's thesis on the topic of "Computer Security". His supervisor was [Amelia Earhart](#) from HR.

[Amelia Earhart](#) was the person with whom the author was first connected to when looking for this Master's thesis. Thus, [Amelia Earhart](#) was one of the few people who knew about project SOEP. However, we did not tell her about our plans to enter the building.

18.2 Execution History

The exact anniversary date is redacted to protect the identity of the target company. To keep a low profile, we only infiltrated a single person, namely the author of this thesis.

Timestamp 1

[Monday](#), 2018/01/01, 09:05:

The author steps out of the train and joins the commuters that are walking in the direction of the main building.

Timestamp 2

[Monday](#), 2018/01/01, 09:10:

Before entering the main building, the author pulls out his phone and pretends to be talking to a friend while following the employees into the building.

Another visitor is keeping the reception desk busy and the entrance hall is bustling, so the author slips by unnoticed and walks up the staircase together with the other employees. Nobody looks at him twice, and nobody asks any questions.



See [Counteract Tailgating](#) on page 139.

Timestamp 3

⟨Monday⟩, 2018/⟨01⟩/⟨01⟩, 09:15:

The author finds an area with some couches and sits down to look around. He is in the middle part of the building. To each side, a hallway behind a badge-protected door presumably leads to offices, workspaces, and meeting rooms. People come and go, and it would not be much of a problem to do another round of tailgating and slip in behind them.

Timestamp 4

⟨Monday⟩, 2018/⟨01⟩/⟨01⟩, 09:20:

The author starts to execute SEP02 (see [Chapter 17](#)). The very first interviewee suddenly starts to get nervous. The author finishes the interview, and the person leaves.

The author hopes that he has misinterpreted the interviewee's body language and proceeds to interview two more employees.

As it turned out later, this very interviewee was a member of the IT center and was thus the exact wrong person to ask questions about computer security without a badge (see [Section 18.3](#) below).

Timestamp 5

⟨Monday⟩, 2018/⟨01⟩/⟨01⟩, 10:00:

A person approaches the author. He gets ready to ask her if she wanted to do a short interview when she asks him who he was and what he was doing.

Luckily, the author has his cover story ready and tells her his name and that he is working on his Master's thesis on the topic of "Computer Security".

She nods and asks who his supervisor is, and the author gives her the name of [⟨Amelia Earhart⟩](#).

The woman tells the author that he cannot be inside the building without a badge. She asks him to follow her.

Timestamp 6

⟨Monday⟩, 2018/⟨01⟩/⟨01⟩, 10:05:

The author firmly expects to be kicked out of the building. However, against the odds, the woman leads him to the reception desk and hands him a visitor's badge. She smiles at him, and he is free to go back inside.

**Mitigation 8****Apply a Badge and Visitor Policy**

Do not hand out badges without having **double-checked the facts** on the story that the visitor is telling.

When handing out a badge, **always request identification** via passport or driver's license first.

Do not let visitors roam the premises alone. They must always be accompanied by an employee.

Timestamp 7

<Monday>, 2018/<01>/<01>, 10:10:

After going back up and calming himself down, the author immediately takes a photograph of the badge with the idea in mind to maybe copy it later.

An additional bonus: the badge shows the **<YuuBlue>** wireless network password. (The photo of the badge is omitted for the sake of protecting the identity of the company.)

**Mitigation 9****Secure the Wireless Network for Guests**

A wireless network for guests is always risky because an uncontrollable amount of people are going to have access to it.

Consider the following points:

- Make sure that the guest network does **not grant access to sensitive data or internal platforms**.
- **Change the password** of your networks every once in a while (too many changes, however, might exasperate your regular guests and your employees).
- **Do not print the password** for everyone to see. Surrender the password only on demand and use, if possible, time-expiring, individual passwords.
- **Log** visitors who received the wireless network password.

⟨Monday⟩, 2018/⟨01⟩/⟨01⟩, 10:25:

The author receives a phone call from his *real* supervisor that it is time to abort the operation for reasons he would not know until later (see [Section 18.3](#) below).

The author acts his way out of the building and closes SEP03.

18.3 Findings and Post-Execution Insights

SEP03 was one of the most exciting schemes. Physically entering a building is entirely different from sitting behind a computer screen and talk to people via email. Being nervous seemed like a big problem at first, but the nerve-fluttering soon subsided once inside the building.

Even though the entrance hall was filled with people, it was not immediately discernable if this had anything to do with the anniversary or if it was normal. In any case, we used the busy entrance hall to our advantage. As it turned out later, there were indeed festivities in progress that day and the ⟨YuuBlue Day⟩ had been moved to this day as well, robbing us of our fallback solution.

It is regrettable that we had to abort the scheme after only one hour, but we were successful in entering the building and starting to execute SEP02. Sneaking into a building and then starting to do interviews with random people (the social-engineering equivalent of screaming loudly at school when not trying to get picked for an exercise) is arguably not very wise. By starting to interview people, we maximized the contact to the employees, and it was clear that, sooner or later, we would come across someone who would ask too many questions or who would not take kindly to us being here without a visible badge.

However, SEP03 was not supposed to be more than a proof of concept. We had silently entered the building and could have gone wherever we wanted. Nobody would have been suspicious of us if we had not started to ask these bizarre questions as part of the interview of SEP02.

We could have done much damage once we were inside the building. We could have placed USB sticks or other hardware on computers. We could have photographed prototypes, whiteboards, or other data if we had encountered such by chance. We could have installed a camera, a microphone, or even a fake wireless access point. This list could be extensively expanded, because there is hardly a limit to what attackers can do once they have established physical access.

What surprised us the most was that instead of getting kicked out after being

discovered without a badge, the reception gave us a visitor's badge¹. With the badge, our movements within the building would have been greatly facilitated, because now more than ever, no one would have dared to ask us what we were doing or who we were.

Furthermore, the password to one of *YuuBlue*'s wireless networks was printed on the badge. Having access to a network opens all kinds of doors to the attackers. Once they have found and infected a vulnerable machine on the network, the attackers can move laterally to other machines, until they have what they want. The resulting costs for the company can be soaring, as a recent example showed [57].

As we came to know later, the first person that we interviewed had alerted the reception desk to our presence, after which an employee came to look for us. After our brief chat with the lady from the reception desk and after we had received the badge, the reception desk had immediately tried to contact *Amelia*, presumably to double-check the facts or to tell her that we were in the building. Fortunately for us, they could not reach *Amelia* at that moment as she was not yet in the office.

Despite our cover story, we had given the lady from the reception desk our real names when she had asked for them. This mistake was not planned and only happened because of our nervousness at that moment. However, no one asked us to present any identification, meaning we could have told her any name.

After the reception desk had given us the badge, they must have informed the facility manager of our presence, who was in a meeting at this point. The facility manager was not at all amused by the fact that a badge-less visitor roamed the building unattended by an employee and reacted enraged. He is ultimately the reason why we aborted SEP03. We did not want to cause any unnecessary commotion since we had already proven that it was possible to enter the building unnoticed.

The facility manager told us later that he had been furious and that we had been lucky that the reception desk had found us first. He said *he* would have immediately called the police, which would have caused much turmoil and would have possibly endangered all further schemes that we had planned.

¹We believe this happened because *YuuBlue* and its employees try to offer exceptional customer service at all times. Exceptional customer service requires a high level of friendliness and politeness, prohibiting the employees from removing visitors (even suspicious ones) from the premises. It is a practical example of the tradeoff between security and politeness.

SEP04: The Market Research Institute

The objective of SEP04 was similar to the one of SEP02 (see [Chapter 17](#)). We wanted to try to get information from people by asking them straightforward. While during SEP02 we asked the employees in person, the goal of SEP04 was to acquire information about people's income **over the phone**.

19.1 Preparation

Asking a person about his or her income is considered to be extremely rude in large parts of the world, Switzerland included. Doing it over the phone is even more questionable. To not immediately scare our targets away, we had to answer the following question:

How Do We Politely Ask for a Person's Salary?

This question was harder to answer than we had anticipated. We searched wide and far, but it seems like there is no polite way to ask such a question, especially if it is out of context. Since we did not have time to keep our target on the phone for a pleasant and long chat to carefully set the situation and context for such a question (they did not know who we were, so such a conversation would have been bizarre and utterly unthinkable), we had to think of something else.

We ended up with the idea to camouflage ourselves as the fictional market research institute [YellowFellow](#). The original name of our fictional market research institute contained the name of a reputable financial company known in Switzerland. This way, we were hoping to gain some trust from our targets. The idea was to innocuously tell the targets that we were making an anonymous data acquisition on the topic of salaries in the greater Zurich area.

How Do We Hide Our Phone Number?

A short search online revealed that anyone can hide the number of his or her phone easily for outgoing calls by changing the corresponding setting.

19.2 Execution History

Timestamp 1

Monday, 2018/04/30, 12:14:

We manually look through the entire [YuuBlue](#) website and sift through all our notes and results from previous schemes (including automatic out-of-office replies, email signatures, and letters) to find as many phone numbers as possible.

We manage to gather 22 direct phone numbers, not counting generic phone numbers that connect to the front desk or similar.

Timestamp 2

Friday, 2018/05/04, 10:00:

We hide our phone number and execute the calls with the following script:

“ *If the wrong person answers to phone or if the call is forwarded:*
Hello, is this ” [Dental Practice](#)”?

Oh, I’m sorry, I must have dialed the wrong number. Have a nice day. *End call.*

If target answers the phone: Hello, this is Marcus Lanz on behalf of the [YellowFellow](#) market research institute.

May I ask you a few questions on the topic of salaries in the greater Zurich area? It will only take one or two minutes.

If yes:

This data acquisition is anonymous, and the call is not recorded.

- In which industry are you active?
- What is your job title?
- Where do you work?

- What is your work quota?
- We are aware that the following question is a sensitive one, but may I still ask you **what your annual salary is?**
If no answer:
May we classify your income according to a range? *Ask ranges until you know the salary \pm CHF 10000.*

That's it! Thank you for your time and goodbye!

”

Marcus Lanz was a fictional person that we invented solely for SEP04.

Timestamp 3

Wednesday, 2018/05/09, 14:00:

Since the first time around, hardly any targets picked up their phones, we execute the calls a second time, with similar success (see below).

Due to a lack of results, we close SEP04.

19.3 Findings and Post-Execution Insights

SEP04 was hard to execute. Call centers do not enjoy tremendous popularity among the general population, especially if they ask nosy questions about salaries.

Even though we had planned a script (see [Timestamp 2](#)), every call was different and open to change should the situation require it.

The first difficulty that presented itself was forwarding. All too often, we were forwarded to the reception desk. Our exit strategy works only once and is quite suspicious if used multiple times on the same person. Luckily, we quickly figured out that a relatively short “crack” in the connection meant that we were being forwarded, giving us time to end the call before somebody could pick up the phone.

However, the far more significant problem was that most of the numbers we called went straight to voicemail. It may have been the case that all of our targets were in a meeting the first time around (we called at 10 o'clock), but at the very

latest when we called all of them a second time some days later (this time in the afternoon) it seemed more likely to us that they had blocked suppressed phone numbers.

After calling all of the 22 targets twice, we had only reached five people:

- Two of the five “did not have [the] time” or “did not want to answer any questions” and hung up after that.
- One of them was in a meeting (but, interestingly enough, still picked up the phone) and politely declined.
- Another one reacted quite strangely by asking for the name of our market research institute again. After hearing it, the person immediately hung up on us.
- The last one referred us to HR for such questions and hung up too.

The rest of the calls were forwarded to the reception desk or voicemail.

As it turned out, we never even got the chance to ask a single question. Getting information from people over the phone seems not to be as easy as we had anticipated. However, it may very well have to do with the fact that a market research institute calling busy people at work may not have been the best plan of ours.

Attacks over the phone might be more effective when targetting a single person with particular texts and goals in mind.

Part III

Ad Meliora

Towards Better Things

Summary of Findings

This chapter summarizes the most important findings and the data we were able to collect. We briefly comment on each of the main findings.

More detailed descriptions of the findings of each attack can be found in the “Findings and Post-Execution Insights” sections of each chapter in [Part II](#).

20.1 Data

Starting without inside knowledge about [YuuBlue](#), we managed to acquire the following information in a relatively short time (about two months):

- Large numbers of email addresses as well as corresponding names and job positions.
- Passwords and corresponding account names, email addresses, and operating systems.
- Project names, inter-employee relationships, work quotas, and availabilities of specific employees.
- Joining date, full name, work quota, position, team names, department names, and the salaries of all employees who started working in March 2018.
- Salary statements of a member of management, containing the residential address, the bank name and bank account number, the personnel number, the salary, the annual bonus, the social benefits, and the number of children.
- The login gate, a web portal to sign-in to the company’s internal applications and services (email, document management system, VPN, IT services including password change functionality)¹.

¹The visual appearance of the login portal was particularly useful for creating a legitimate-looking fake login site to enable the phishing attack in SED04 (see [Chapter 6](#)).

- The password policy of [YuuBlue](#) and the antivirus software in use.
- The password of the [YuuBlue](#) wireless network.
- The name of the software system used to process invoices.
- The layout of [YuuBlue](#)'s main building, including location of the canteen, offices, meeting rooms, and workspaces.
- Knowledge about existence or lack of physical security measures when entering the main building.
- Access to employees in familiar surroundings where they were more comfortable to divulge sensitive information.

Generally, every piece of information, however small or insignificant it may seem, can help attackers to advance their cause. Every name and every phone number gained is a step in the direction of finding the weakest link into the system of a company, and a single password may be enough to compromise the entire system.

Some of the information we acquired was extremely sensitive and valuable in itself. Information like salary statements put employees and even their families at risk of being extorted, threatened, or even robbed.

Other information, like the extensive list of email addresses, helped us to execute further attacks, while they were of no immediate danger to the individuals.

20.2 General Findings and Interpretation

This Master's thesis proves that no experience or formal education in the field of social engineering is required to attack and breach a company successfully. Criminal energy alone suffices, making improved defenses and security procedures a must-have for every organization, big and small.

In the following sections, we want to emphasize the most exciting findings of the thesis:

Identity Theft

Using the identity of the CEO of [YuuBlue](#) to register a domain and a server was not difficult at all (see [Virtual Private Server](#) in [Section 3.1](#)). Even when we were prompted for identification after the attacks have been reported, we found a way to keep both the server and the domain online, by making the VPS provider lie to [YuuBlue](#) on our behalf. Social engineering has proven useful on all layers.

The unchecked registration of domains has been a problem for a long time and continues to do so. Criminals register thousands of domains for phishing purposes and to feed their botnets and malware with commands. Better identity checks need to be put in place by the responsible companies and authorities.

**Mitigation 10****Terminate Malevolent Customers**

If a domain, server, or customer is suspected to be involved in criminal activities like phishing or identity theft, the domain and server providers should investigate the matter diligently and should, if proven guilty, terminate malevolent customers, even if the providers stand to lose money from the termination. If the circumstances are uncertain, the providers should double-check the facts and establish direct contact with the parties concerned. Often, criminal customers will not be reachable and should thus be treated guilty (i.e., their servers or domains should be terminated).

Out-of-Office Replies

The out-of-office replies of the employees of [YuuBlue](#) proved time and again to be extremely dangerous. Out-of-office replies are packed with information that can be used to attack a company (see [Section 4.3](#)). What is even worse is that this information can be polled by attackers whenever they are in need of new availability information (see SED12 in [Chapter 10](#)).

**Mitigation 11****Avoid or Limit Out-of-Office Replies**

Every company should define clear rules as to what information may or may not be disclosed in out-of-office replies. Email addresses of former employees should never return an out-of-office reply.

Considering the danger these out-of-office replies pose, it may be appropriate to give up on them altogether. Instead, a new system may be put in place that forwards the emails of employees who are out-of-office to an individually designatable and trustworthy colleague (a so-called *out-of-office proxy*) who can then decide if an answer is required (e.g., to a customer), or if the email may be dealt with when the employee is back in the office (e.g., spam). This way, companies can inform people selectively on a case-per-case basis about the availability of employees and expose information much more sparingly and safely.

Enumeration of the Labor Force

Thanks to the [YuuBlue](#) website and social networks such as LinkedIn, it was simple and efficient to obtain large numbers of employee names and email addresses (see [Section 4.1.1](#)). Especially LinkedIn, with its features to sort for current employers and work location, facilitates the enumeration of a good part of a company's labor force, complete with names, email addresses, and job descriptions.

The obtained list of employees can be further refined by polling the email addresses for bounces (as demonstrated in SED12 in [Chapter 10](#)). The bounces likely correspond to employees who no longer work at the company. These employees can subsequently be removed from the list.

We expect the enumeration of the labor force of companies to become even more straightforward if social networks such as LinkedIn continue to gain popularity.

Social networks are advised to take this matter seriously² and should consider turning off or further restricting their extensive sorting features.



Mitigation 12

Educate and Train All Employees

Since companies cannot control what their employees share on social and business networks, they must assume that every single employee is potentially exposed to attacks. Consequently, companies should regularly educate and train *all* of their employees to be able to withstand social-engineering attacks (see [Section 21.3](#)).

What companies *can* control is their website, on which they should share as little contact information as possible. If in doubt, use a contact form instead of an email address.



See also [Avoid Publishing Contact Details](#) on page 45.

Money and Information

In our attacks, we were able to exfiltrate sensitive information with moderate effort, but even though we tried three times, in three different ways to extract financial assets, we remained to be unable to do so.

²LinkedIn *does* take the matter seriously, as the restriction of our account described in [Section 4.1.2](#) proved.

It seems like [YuuBlue](#) defends money much more fiercely than information. Is money so much more valuable than information? In an age where information is omnipresent and where companies are collecting more data than anyone could ever use, information has enormous power and can decide if a company is going to make it or not. In light of this power, we argue that information may be of even more value than money.

Even if one does not believe in the inherent value of information, a look at the costs of data breaches, both money-wise and reputation-wise, might convince a skeptic.



Mitigation 13

Protect Information

Companies should defend sensitive information about their projects, processes, products, buildings, plans, employees, applicants, and especially customers diligently. Protecting sensitive information can be done in a similar way as protecting money: by putting 2nd-factor-authentication-like procedures in place (see, for example, the four-eye principle in the Findings of SEP01 in [Section 16.3](#)).

Information access should be tightly controlled and logged. Rules should dictate what information may be shared and accessed when and by whom (see [Section 21.2](#)).

Since the HR department is a hub of sensitive information and thus a prime target, special care must be devoted to educating employees and to preparing them for emergency situations (see [Section 21.3](#)).

Courage to Address People

To send emails on behalf of the [yuublue.com](#) domain, we were sitting in a public staircase for an extended period of time (see [Timestamp 3](#) and following of SED16 in [Chapter 13](#)). Waiting in a staircase may not be in itself suspicious, but doing so with a computer might be. Despite the suspicious nature of our little adventure, nobody had the courage to address us and to question our intentions.



Mitigation 14

Have the Courage to Address People

Companies should establish a culture of security awareness (see [Section 21.1](#)) among their employees so that employees find the courage to ask people what they are doing. Asking does not hurt and depending on

the plausibility of the answer, actions can be taken.

Such a culture of security awareness can only exist if the employees can trust their company blindly and if they know that they will be backed by their employer unconditionally.

Tailgating

Tailgating (see [Section 2.1.6](#)) has worked for us twice (see SED18 in [Chapter 14](#) and SEP03 in [Chapter 18](#)). It is a simple technique that only works because of our social nature and society's rules of politeness. Despite the simplicity of the technique, tailgating can cause substantial damage to a company, because physical access is given to the attacker.

Physical access allows an attacker to install hardware (e.g., leave USB sticks with malware, install MitM³ hardware or software, install cameras or microphones, install backdoors, both digitally and physically⁴) or to steal information (e.g., laptops or hard disks, photos of whiteboards, plans, and other interesting information laying around).

Physical access to a building is dangerous, since there is no telling what the attackers might have done in the time they were inside. While attacks on the wire leave traces in logs or elsewhere, having physical access potentially allows to silently wreak chaos (if one knows how to avoid the occasional camera). Cables can be unplugged or reconnected differently, servers can be rebooted manually, hard disks can be removed and stolen, doors or drawers can be broken, and sensitive information may be laying around openly because people carelessly think that nobody is going to break into the building. These scenarios are but a short list of examples and are by no means exhaustive.

More often than not, physical access has destructive effects (broken doors, locks, harddisks, or computers, leading to Denial-of-Service situations (see [Section 2.4](#))) and has its limits in that it relies to a certain degree on luck to encounter unencrypted harddisks, heedlessly stashed, sensitive information, or unlocked computers.

Furthermore, physical access comes with the substantial risk of getting caught and having to face various charges that might be more severe than charges invoked by hard-to-prove, digital misdeeds. Another reason physical access is not more often exploited is that digital access is much simpler and less expensive. Digital access can be achieved remotely, and in case of detection, there is bet-

³Man-in-the-Middle (see [Section 2.5](#)).

⁴A skilled attacker might dress as a locksmith and renew or reprogram the locks of a literal backdoor.

ter anonymity. Furthermore, physical access is of much less use if a distributed system of many computers or IoT devices that are locally separated needs to be compromised, since one cannot be at multiple locations at the same time.



Mitigation 15

Counteract Tailgating

The culture of security awareness introduced in [Section 21.1](#) can help to avoid tailgating or piggybacking events to a certain degree by encouraging employees not to let *anybody* pass directly behind them. Employees should always wait for doors to close behind them and point out to potential tailgaters that they must identify themselves with a badge or similar.

If a would-be tailgater fails to present identification or if an employee suspects that a tailgating incident has taken place, the responsible facility manager must be informed immediately. The facility manager must then retrace the steps of the trespassers to find out, what they have done to prevent further damage down the road. This damage assessment is a difficult task and may even be impossible. For unauthorized accesses, like for diseases, prevention is better than a cure.

Companies must find a way to enforce *quick* identification even of *groups* of people so that a tailgater cannot just blend in with a group, as we have done in SEP03 (see [Chapter 18](#)). The badge system of today works, but is not fast or convenient enough to deter groups of people from entering together. The future may bring better solutions.

Antivirus

Even though social engineering is a dangerous threat, conventional security measures must not be neglected. If [YuuBlue](#) had not employed proper antivirus software, we could have caused a lot more damage using keyloggers and Meterpreter payloads, with minimal effort.

Antivirus software (see [Section 2.11](#)) and intrusion prevention systems (see [Section 2.12](#)) are still integral pieces of a security system and protect the employees and their hardware — if not from everyone, then at least from inexperienced malware developers such as ourselves.

20.3 Weaknesses

⟨YuuBlue⟩ does its best to defend its assets and data. The company employs an entire team dedicated to running their IT infrastructure. This team proved to be able to react quickly in the case of an attack (see SED04 in [Chapter 6](#) and SED18 in [Chapter 14](#)). A well-working IT team is an excellent defense against many attacks.

Hacking, and social engineering in particular, however, are all about finding the weakest link in the chain. Since the IT team is rarely the weakest link, it is of limited use when it comes to defending against social-engineering attacks. All links in the security chain need to be continuously honed and reinforced since a single weak link potentially suffices to breach a system or to cause significant damage. It puts any defender at a disadvantage compared to the attacker since the attacker needs to find only one weak link, while the defender needs to reinforce all of them.

An analysis of our attacks (see [Part II](#)) identified **the following main weaknesses** of ⟨YuuBlue⟩ that enabled us to successfully execute our attacks:

Information Overflow in Out-of-Office Replies

Out-of-office replies played an essential role in many of our attacks (SED09, SED11, SED13, SED15, SEP01, see [Part II](#)) since they provided us with the availability (see [Section 4.3](#)) of specific employees, which we could subsequently use in attacks utilizing private email addresses.

As mentioned before, it may be prudent to cease using out-of-office replies in the way they are used today.



See [Avoid or Limit Out-of-Office Replies](#) on page 135.

Inability to Recognize Phishing Mails

One of our most dangerous attacks was SED04 (see [Chapter 6](#)), where we were able to extract passwords using a simple phishing scheme. Phishing (see [Section 2.1.1](#)) has been around for years and continues to cause substantial damage because people keep falling for it.

We recommend that *all* employees receive regular training on how to detect phishing attacks (see [Section 21.3](#)).

Apart from that, we recommend using two-factor authentication (2FA) wherever possible. Google, for example, reduced the number of phishing incidents

among their numerous employees to zero, after it introduced 2FA as a mandatory phishing countermeasure in 2017 [54].



See [Spot a Phishing Email](#) on page 48.



See [Handle Links and Attachments Suspiciously](#) on page 49.

Trusting Private Email Addresses

In the schemes SED09 (see [Chapter 8](#)), SED13 (see [Chapter 11](#)), and SED15 (see [Chapter 12](#)), we created email addresses with the intention of impersonating the colleagues or superiors of our targets. We then told the targets made-up stories (see [Pretexting](#) in [Section 2.1.3](#)) and hoped that these stories sounded plausible enough so that the targets would fall for it. Most of the data that we illegitimately received as a result of these stories was sent to a “private” email address by unsuspecting employees.

We recommend generally mistrusting any email or text message coming from a hitherto unknown email address or phone number (see also [Section 21.2, How to Handle Sensitive Information?](#) on page 147), even if the sender pretends to be a person one knows well. If in doubt, the person should be contacted via a second channel to validate the facts.

We furthermore recommend exposing employees to such situations regularly to evaluate the ability of the company to defend against social-engineering attacks (see [Section 21.3](#)).



See also [Mistrust Unusual Email Addresses](#) on page 68.

Published Contact Details

[YuuBlue](#) has an information-rich website with many pages. These pages not only inform the visitors about [YuuBlue](#)’s projects, events, and news, they also provide contact details to many of [YuuBlue](#)’s employees. Especially the list of authors of the blog written by [YuuBlue](#) employees hands an extensive list of email addresses and job descriptions on a silver platter to the attackers.

Contact details such as names, email addresses, job descriptions, and social media handles are valuable to attackers. They allow the attackers to get to know a company in more depth by enumerating its workforce, by finding suitable targets for an attack, and by drawing a relationship chart between employees,

thereby enabling the attackers to abuse the authority a superior might have over another employee.

In our case, published contact details directly enabled the execution of SED04 (see [Chapter 6](#)) and SED12 (see [Chapter 10](#)), and helped in finding a target for SED09 (see [Chapter 8](#)).

Even though contact details alone do not make an attack successful (the targets still need to fall for the attackers' tricks), we generally recommend to publish as few contact details as possible. The few employees whose contact details are required to be published should receive extensive, regular training against social-engineering attacks (see [Section 21.3](#)).



See also [Avoid Publishing Contact Details](#) on page 45.

Unencrypted Information on Insecure Channels

During schemes like SED09 (see [Chapter 8](#)) or SED11 (see [Chapter 9](#)), we received *sensitive* data over the *insecure* email protocol. Ideally, this data should have never left [YuuBlue](#) in the first place, but it did, and, adding insult to injury, it did so in an unencrypted manner. Anybody listening in on the connection could now have the same data as we obtained.

When transmitting sensitive data, or any data at all, one should do so in a safe manner (see also [Section 21.2, How to Handle Sensitive Information?](#) on page 147). Encrypting data for transmission is not difficult and can be done quite fast using native or third-party programs, depending on the operating system.

We recommend that all employees should receive regular training on how to securely transmit data (see [Section 21.3](#)). In these training sessions, the dangers of unencrypted transmission of sensitive data and the ease with which attackers can listen in on a connection should be brought to the employees' attention by explaining theory as well as showing practical examples.

Furthermore, not only digital data needs to be handled carefully. When talking about project details, one should be aware of one's environment to protect against eavesdroppers.



See [Encrypt Data for Transmission](#) on page 69.

Failing to Implement SPF Securely

Implemented correctly, the Sender Policy Framework (see [Section 2.14](#)) allows email receivers to check if the email has been sent by a legitimate email server

belonging to the domain in question. [YuuBlue](#) had implemented SPF so that we could not send emails on behalf of [yuublue.com](#). However, [YuuBlue](#) also implemented an exception, which enabled any computer that was in the company's wireless network, and thus using its IP address, to legitimately send an email on behalf of [yuublue.com](#).

This exception would have been fine if it did only work within the company's primary network used by its employees. Unfortunately though (or by design?), the exception extended to another one of [YuuBlue](#)'s wireless networks, whose password was distributed to all guests receiving a badge. The extension to this second network network allowed the execution of SED16 (see [Chapter 13](#)), SED18 (see [Chapter 14](#)), and SED19 (see [Chapter 15](#)).

We recommend the implementation of SPF to be tested thoroughly to make sure that exceptions do not have any unintended side effects for guest networks. We also recommend to use SPF in combination with DKIM (see [Section 2.14](#)) and DMARC (see [Section 2.14](#)).



See [Secure Your Domain](#) on page 101.

Wireless Password Easily Obtainable

[YuuBlue](#) prints the password to one of its wireless network directly on each visitor's badge, enabling us to execute SED16 (see [Chapter 13](#)), SED18 (see [Chapter 14](#)), and SED19 (see [Chapter 15](#)). Because the number of guests who receive a visitor's badge and who thus have access to [YuuBlue](#)'s wireless network accumulates quickly, there is soon no way to know precisely how many people have access to the network. Consequently, a company's guest network takes on similar traits as a public network that can be accessed by anyone.

Therefore, we recommend taking special care of any guest networks and make sure that no guest can access sensitive data stores or penetrate further into the system from that network.

We further recommend not printing the password anywhere and only surrender it on demand. That way, the reception has more control over who receives a password and who does not. We recommend logging everyone who has received the password and we recommend changing the password regularly.

If possible, individual passwords with automatic expiration times (in combination with extensive logging) should be employed.



See [Secure the Wireless Network for Guests](#) on page 125.

Liberal Handling of Sensitive Data

In SEP02 (see [Chapter 17](#)), we asked people precise and intruding questions about their passwords and habits when it comes to computer security. All of them answered all of our questions in great detail and seemed to be happy to comply.

Even though it is hard to do and against our nature, we would do well to ask ourselves from time to time, whether our conversation partner (be it online or offline) has the right to have or obtain certain information. If in doubt, it may be prudent to withhold the information.

Furthermore, there is information to which nobody has any access right. Among this information are passwords (see also [Section 21.2, How to Handle Passwords?](#) on page 148). Passwords should *never* be shared or written down. They should only exist in one's head or in encrypted form (e.g., password manager). Every (well-planned and -implemented) system is designed to be accessible by an administration password. Thus, *no* system technician, administrator, or "IT guy" will *ever* have to ask *anybody* for a password.



See also [Disclose Information Prudently](#) on page 119.

Insufficient Visitor Processes

When prompted for our names in SEP03 (see [Timestamp 5](#) in [Chapter 18](#)), we were handed out a badge *without* having to show any identification. We were sent back into the building *without* an escort and *before* our story was fact-checked.

We recommend tightening the process around visitors so that a badge is only handed out if valid identification can be presented. This way, the reception desk always knows who is in the building. Furthermore, no guests should be allowed to be unattended while on the premises and should always be accompanied by an employee (see also [Section 21.2, How to Deal with Visitors?](#) on page 151).



See also [Apply a Badge and Visitor Policy](#) on page 124.

Insufficient Physical Security Checks

In SEP03 (see [Chapter 18](#)), we were able to enter [YuuBlue](#)'s main building effortlessly. Hiding among a group of commuters, we slipped by the reception desk and climbed the stairs. So far, no badge was required. The first badge-protected door was out-of-sight of the reception desk, meaning that nobody was

paying attention anymore to who was coming and going.

Overall, the physical security checks were insufficient. We recommend requiring every employee (and thus potential tailgater and visitor) to present a badge at a door that is within sight of the reception desk.



See also **Counteract Tailgating** on page 139.

Being Too Nice

This point may seem to be unfair since being nice is the way to go in most situations of life, but the openness and politeness of [YuuBlue](#) facilitated attacks like SEP02 and SEP03 (see [Chapter 17](#) and [Chapter 18](#)) significantly. We are not advocating to be rude or impolite, but we suggest to nurse a culture of security awareness (see [Section 21.1](#)), including harboring a healthy suspicion if unusual events take place (like a badgeless visitor).



See also **Have the Courage to Address People** on page 137.

Mitigation and Countermeasures

Social-engineering attacks are challenging but not impossible to defend against. Setting up a social-engineering defense is not a one-time event, but rather a continuous process of education.

In this chapter, we present suggestions on how a company can prepare their employees for social-engineering incidents.

21.1 A Culture of Security Awareness

In the best position to defend their company against social-engineering attacks, such as phishing or tailgating, are the company's employees. That is why every company would do well to build a **culture of security awareness** among their employees. This culture needs to be exemplified top-down, starting with the C-level executives.

If all employees were aware of social engineering and its schemes, a lot of damage could be prevented and confidential business data would be more secure. However, just as cybercrime is continuously morphing, education and defense training should never stop, and penetration tests should be repeated every once in a while (at least once a year).

The ultimate goal of a company is to develop a culture of security awareness that is lived by every single employee. This culture encourages the employees to think critically about what they do online and offline regarding security. It rewards successful thwarting of attacks and responsible actions, but does never, under no circumstances, penalize the inability to detect or prevent an attack.

Security awareness can be established by educating employees about how social-engineering attacks are usually carried out and what dangerous effects they can have. Examples and hands-on exercises should, if possible, accompany these educational sessions to give the employees an idea of how easy it is to

execute such an attack in practice and how one can recognize it.

21.2 Plans, Protocols, and Procedures

The company, *together* with its employees, should improve plans, protocols, and procedures for specific (attack) situations. These plans, protocols, and procedures should be taught to the employees regularly (at least once a year) with the idea in mind, that people will act more security-aware if they know the *whys* behind the rules.

Fixed processes also make sure that the employees can react to an attack fast, without having to overthink in a stressful situation. Processes may need to be updated from time to time, adapting to new threats or insights. The topics for which rules and procedures may need to be defined include, but are not limited to:

How to Handle Sensitive Information?

As a first step, the company needs to define what information is sensitive and what information is not. Generally, any information that is not public knowledge should be treated as sensitive. If in doubt, always treat information as sensitive.

As a second step, the risk of sensitive information being exposed in different departments and areas of the company should be gauged, and rules should be put in place on how this information is to be treated, according to its exposure risk level.

These rules may include encryption enforcement for all sensitive information (as mentioned before in [Section 20.3, Unencrypted Information on Insecure Channels](#) on page 142) and conditions on when high-risk information may be accessed and for what reasons. For example, accessing of high-risk information should always be preceded by a double-check of the facts via a second channel and must not be sent to external email addresses (as discussed in [Section 20.3, Trusting Private Email Addresses](#) on page 141). Some information may only be accessed but must not be sent to anyone at all.

Similar rules should be put in place for other levels of exposure risk.



See also [Mistrust Unusual Email Addresses](#) on page 68.



See also [Encrypt Data for Transmission](#) on page 69.

The principle of **must-know** should be enforced so that only those who need

it can access information. Access rights should be managed on an individual level (not on a group level), and it may make sense to automatically revoke access rights after a certain amount of time in which a resource has not been accessed (temporary access rights). The responsible administrators should tightly and regularly control access rights, and logs should capture every resource access.

These rules should be accompanied by regular educational sessions (once a year) for the employees that visualize why the rules are in place. Knowing the reasoning behind the rules may encourage employees to respect them.

Furthermore, paper waste is a source of information for attackers that should not be underestimated (see [Dumpster Diving](#) in [Section 2.1.7](#)).



Mitigation 16

Prevent Dumpster Diving

Trash often contains valuable information, such as sales numbers, passwords, names, salaries, and other sensitive data that has been carelessly thrown away.

The following measures may help to prevent *dumpster diving* (see [Section 2.1.7](#)), the act of stealing information out of trash cans:

- **Offer separate trash cans** for sensitive information. Documents thrown into these cans should regularly be destroyed. If possible, *all* paper waste, hard disks, and CDs should be shredded.
- While trash waits to be collected by the responsible authorities in unsecured places (e.g., back alley), **a lock should secure the dumpsters.**

How to Handle Passwords?

While it may not be necessary to enforce regular password changes¹, all employees should be regularly (once a year) educated on best practices regarding password handling:

Secure passwords that are easy to remember can be created in many ways. A password should be reasonably long and not one of the standard, top-100 most used passwords. It makes sense to change important passwords regularly and never to use default credentials.

Passwords should not be written on post-its or other physical or digital

¹Enforced regular password changes often lead to people only incrementing parts of their password which makes the enforcement useless.

notepads. Passwords should only exist in a user's mind or encrypted form (e.g., a password manager). The use of password managers can significantly facilitate the simultaneous use of large amounts of different passwords for different services. The educational sessions should include guides on how to use password managers and the advantages and disadvantages of the various managers around.

If users decide not to use a password manager, different ways of how easy-to-remember passwords can be created should be presented.

An essential and prominent part of the educational session should emphasize that passwords must never be given to *anybody* and must never be entered on request (e.g., in a phishing mail), especially if urgency is pointed out. Every service is built with the case in mind, that the support needs to access certain parts without a user password. Consequently, there is no need for an administrator to know a user's password. Thus, no legitimate service will ever request a login or a password change² (as discussed before in [Section 20.3, Liberal Handling of Sensitive Data](#) on page 144).



See also [Disclose Information Prudently](#) on page 119.

The educational sessions may, depending on the level of expertise of the attendants, include more or less in-depth explanations of how the login process works on a website. The purpose of it being that if people know, for example, that stolen cookies work just as well as passwords to get access to an account, or that passwords entered on websites that are not protected by SSL/TLS³ can be captured and seen by everyone on the network (again, hands-on examples will work much better than simple explanations), they might start to log out after having used a service and they might make sure that a green lock is next to the address in their browser before they enter their passwords. Their actions may inspire others to do the same, and eventually, the entire company might browse more safely and conscientiously.

How to Handle Links and Attachments in Emails?

Many attacks start with an email containing a link or an attachment. The email as an attack vector is extremely popular and dangerous since, still, many people fall for it (see discussion in [Section 20.3, Inability to Recognize Phishing Mails](#) on page 140).

²If a service does so, it acts unprofessionally and is actively undermining the effort of establishing a culture of security awareness. In such a case, do not use any links provided (they may be wrong) and log in in the usual way.

³Secure Sockets Layer/Transport Layer Security (used for authenticating and encrypting the payload between browser and server).

All employees are potential targets and should be regularly (once a year) educated on how to treat emails with links and attachments.

As a general rule of thumb, links and attachments should only be clicked or opened if the email they came in and the data they represent were expected or requested and if the sender is known and trusted. Exceptions include attachments in the form of executables which should *never* be downloaded and executed.



See [Handle Links and Attachments Suspiciously](#) on page 49.



See also [Spot a Phishing Email](#) on page 48.

The educational sessions should emphasize the importance of patching and updating one's system by showing hands-on exploits of PDF readers to encourage the employees always to update their systems and applications as fast as possible.

Hands-on demonstrations of Microsoft Word Macro malware and their inner workings can also be used to show the importance of security awareness. Employees should be advised never to enable macros of documents that were attached to emails from unknown sender addresses. If possible and not needed, macros should be turned off for good without notification in the Microsoft Word settings.

Attachments in the form of executables in emails (be it expected or unexpected) should *never* be downloaded and executed, even if they come from known addresses. Attachments in the form of archives should be treated with suspicion as well and should only be opened if the sender address is known and trusted and the email is expected.

Unexpected emails from known sender addresses with suspicious attachments should be investigated via a second channel (e.g., calling the sender) before the attachment is opened.

If it is necessary to open attachments from an unknown sender address, it should be done in a safe environment (e.g., a virtual machine). If an employee does not have the necessary expertise to open a suspicious document, the responsible security professional of a company should be tasked with helping the employee.

Links pose a less immediate danger to a company than attachments. Even though clicking a link might surrender information to an attacker, clicking the link is in itself not dangerous. However, if the link does not lead where it promises (i.e., when the link displayed is not the same link that the browser displays when hovering it), if the link directly downloads a file, if the email requests credentials, or if the domain is a look-alike domain (see [Section 2.6](#)), it should not be clicked. Although, there is also the chance of hitherto unknown zero-day vulnerabilities

(see [Section 2.9](#)) being used, making clicking a link just as dangerous as executing attachments. That is why it is generally recommended not to click suspicious links in emails.

Furthermore, credentials and other personal or sensitive data should never be entered on request (see [How to Handle Passwords?](#) above and [Section 20.3, Liberal Handling of Sensitive Data](#) on page 144).

How to Deal with Visitors?

Despite having already discussed the topic in [Section 20.3, Insufficient Visitor Processes](#) on page 144, we describe more detailed procedures here.

No matter if a visitor wears a uniform or not, any visitor may only enter the building if announced by an employee beforehand. The visitor should always be required to show an identification card or passport before receiving a visitor's badge. These identification cards should be scanned or copied to identify the visitor later if anything should go wrong.

The employee who invited and announced the visitor should accompany the visitor everywhere and never leave him or her alone. If the visitor needs access to a server room, a control room, or a similar sensitive location, an employee of the corresponding department should be involved who is knowledgeable enough to ensure that the visitor does not engage in sabotage or that the visitor does not steal any data.

All employees and visitors should be required to wear their badge visibly at all times. Any person who does not wear a badge should be politely asked to present one. If the person is not able to do so, the employee needs to immediately accompany the trespasser to the reception desk or exit where the situation needs to be investigated and cleared.

Because of the friendly and open nature of many employees and the fear of retribution in case a superior should be asked for identification, many employees will hesitate to ask a person they do not know for a badge. It is therefore crucial that the company supports the security-aware, courageous actions of their employees unconditionally, even if a high-value customer feels offended after having been accompanied back to the exit. **A culture of security awareness can only be established and work, if absolute trust prevails between the company, its employees, and their superiors.**



See [Apply a Badge and Visitor Policy](#) on page 124.

How to Handle Ongoing or Successful Attacks Against the Company?

If despite all the precautionary measures an attack is ongoing or has already been successfully executed, plans and procedures should be in place that can be followed even by a new employee who has been with the company for less than a week.

These plans need to include the following five points, ordered by priority:

First, defensive measures to get the attackers out of the system need to be taken. Even though every second may count in an ongoing attack, it is often prudent to sit back and think about the problem for a minute, rather than taking action hastily and possibly making the situation even worse.

Second, the damage that was done should be assessed and, if possible, repaired. Any remains of the attack, both physical and digital, should be removed from all systems (except for logs and other evidence needed in further investigations).

Third, all afflicted third parties or, if necessary, the public, need to be informed of the attack so that they can prepare defenses and take actions to prevent further damage, depending on the data that was stolen.

Fourth, the attack should be analyzed, and rules and procedures should be updated to prevent similar attacks in the future. Under no circumstances, however, should a single individual or a group of individuals be declared to be guilty publicly or internally, under no circumstances should negative consequences be enacted on the individuals, and under no circumstances should anybody be made an example of, even if having enabled the attack in the first place. Instead, the attack should be evaluated together with all involved parties. Reasons for the success of the attack should be investigated, and solutions to prevent such an attack in the future should be developed.

On the contrary, if somebody acted well during an attack and prevented further damage, these actions should be communicated and rewarded publicly to create an incentive for employees to strive for higher levels of security awareness.

Fifth, legal actions against the attackers may be considered. Legal actions do not repair any damage, trust-, or reputation-losses. Therefore, they may be considered the least important part of the response to the attack.

Failing to act responsibly and sincerely in points three and four breaks the trust of customers, employees, and the public in a company. With broken trust, a culture of security awareness cannot exist. Trust is hard to repair and, once broken, may never reach the same level again. Trust may very well be the single most important value when it comes to business and security.

21.3 Educate, Drill, Repeat

The key to a successful defense against social engineering is security awareness. Security awareness can be established by **educating the employees regularly** about the most common social-engineering attacks and by teaching them rules and procedures for different situations, as described in the previous section.

However, theory and practice are different, which is why we recommend **executing regular, unannounced social-engineering attacks**, comparable to fire drills. These social-engineering drills can be executed by an internal or external penetration tester.

The goal of these drills is not to find, expose, or embarrass individual employees (just as a fire drill's goal is not to embarrass the slowest employee). Instead, it is a way for the company to measure the success of their security awareness program. It is a feedback loop for the company and allows it to educate its employees selectively and systematically.

Closing Remarks

It is distressing that only a moderate amount of effort is necessary to to exfiltrate all kinds of sensitive information. As long as there are humans somewhere along the security chain, data remains at risk of being stolen, computers remain at risk of being infected with malware, and companies remain at risk of losing money and reputation.

As demonstrated, experience or specialized training is not required to execute social-engineering attacks. Social engineering comes naturally to many of us since we use our social skills every day. However, even though we can learn about and even apply the techniques ourselves, we remain susceptible to attacks. It is simply the human nature. Experience and concentration is required to spot and successfully thwart social-engineering attacks in their tracks, and it is difficult to overcome the societal dogmas of sheer politeness and hierarchy.

While organizations all over the world invest staggering amounts of money in technical security systems such as cameras, gates, scanners, firewalls, antivirus software, and intrusion detection and prevention systems, the education of their workforce was neglected.

The threat emanating from social engineering is not going to become smaller in the future. If we want to come out on top in the battle against criminal entities, we need to start educating the workforces, the public, and our children. We need to hang the apples so high and become targets so small that social engineering ceases to be profitable.

A Look Ahead

The ever-changing landscape of criminal activity in the digital world needs to be countered with at least as much creativity as the other side is displaying. The goal should not only be to keep up in the fast-paced arms race but to overtake the other side eventually, an admittedly ambitious goal.

A first step in this direction is the improved education of the workforces that

should be embraced by companies. All employees need to be taught how the most common attacks work, how one can fend them off, and how to behave in a safe and cautious manner in everyday situations, both online and offline.

The industry as a whole needs to work towards safer processes. Identity thefts should not be possible any more in the future. Browsers should attempt to mark phishing websites with obvious visual cues about the certificates in use, and about the address visited (e.g., “did you mean [yuublu.com](#), instead of [yuublu.com](#)?”, if [yuublu.com](#) has been visited before). Imaginable is a global warning system, where trusted security experts mark websites as phishing sites. A browser could subsequently display a warning if such a website is visited.

The research community can help where companies cannot, namely in presenting detailed findings of penetration tests or data breach investigations without fearing the loss of trust, respect, or money. The research community can help organizations to develop impermeable procedures and guidelines to prevent, react to, and deal with social-engineering attacks. Aside from shedding light on obscure data breaches and from improving existing procedures, the research community can work on more secure protocols and architectures¹. Furthermore and most importantly, students of all fields of study need to be sensitized to (and taught about) social engineering, so that they are equipped when they start working in the industry.

The private sector, which is already huge in the cybersecurity field, will further grow in the area of social engineering. Companies will continue to offer education against social engineering as a service, an area of business that will grow further. We will see better technology to prevent tailgating, replacing the then obsolete badge system of today. Artificial intelligence will soon be in use on both sides of the battlefield, not only for impersonating humans, but likewise for detecting fraud, crime, and social engineering.

Last but not least, we as the end-users around the world need to be willing to forego certain amenities and convenience for increased security. Security has a price and we need to be willing to pay that price.

¹See, for example, the SCION Internet Architecture [58].

Glossary

Some of the following definitions are inspired by Wikipedia. Most of the definitions echo those in [Chapter 2](#), where sources are stated, if applicable.

Antivirus Software Antivirus software is the most widely used malware countermeasure (often offering prevention, detection, and removal of malicious software). [16](#)

Baiting Baiting is a technique where social engineers leave removable storage media devices such as USB flash drives as baits in strategic places. [10](#)

Blacklisting A blacklist is a list of all the entities who should be blocked and who are not allowed to perform a particular action. [20](#)

Bot A bot is part of a larger botnet and allows the botmaster (the owner of the botnet) to use the infected machine for malicious purposes by giving the botmaster access to every part of the machine. [12](#)

Denial of Service A denial-of-service attack tries to overwhelm a system with many requests until eventually, the system cannot serve legitimate requests anymore, often due to a lack of resources. [13](#)

Domain-Based Message Authentication, Reporting, and Conformance Domain-based Message Authentication, Reporting and Conformance is based on Sender Policy Framework and DomainKeys Identified Mail and is used to authenticate emails and to detect and prevent the spoofing of email sender addresses. [21](#)

DomainKeys Identified Mail DomainKeys Identified Mail is a method to authenticate emails and to detect the spoofing of email sender addresses. [21](#)

Dumpster Diving In the context of social engineering, dumpster diving describes the act of searching trash bins and dumpsters for information such as contracts, business numbers, internal memos, emails, salaries, or passwords. [11](#)

Greylisting Greylisting is a method of defending email users against spam. A message is initially held back and is only delivered to the target mailbox when delivery is attempted a second time. [20](#)

- Intrusion Detection System** Intrusion detection systems monitor the traffic to a computer and try to identify malicious software before it reaches the target computer. [19](#)
- Keylogger** Keyloggers are pieces of software that record the keys pressed on a keyboard attached to the target machine. [12](#)
- Look-Alike Domain** A look-alike domain is a domain that resembles a well-known domain. [13](#)
- Malware** Malware is software that was designed to interfere with a computer's normal functioning. [12](#)
- Man-in-the-Middle** In a Man-in-the-Middle attack, an attacker is secretly relaying messages between two parties who think that they are directly communicating with each other. [13](#)
- Metasploit Framework** The open-source Metasploit Framework is a penetration testing framework. [14](#)
- Meterpreter** Meterpreter is an advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime. [14](#)
- Open-Source Intelligence** Open-source intelligence denotes intelligence that can be gathered freely and publicly from social networks and similar sources. [11](#)
- Phishing** The term phishing denotes the procedure of tricking a target into opening a link or a document by posing as a trusted entity, in order to gain further information. [8](#)
- Piggybacking** See **Tailgating**. [10](#)
- Pretexting** While pretexting, social engineers act out a story in front of a target and thereby convince the target of the truthfulness of a scenario, increasing the chance that the victim will disclose information or perform an action, which would be unlikely under usual circumstances. [9](#)
- Ransomware** Ransomware is software that encrypts parts of a hard disk on the target machine and subsequently demands a ransom for the decryption key. [12](#)
- Regular Expression** A regular expression is a string that defines a search pattern. [16](#)

- Rootkit** A rootkit is a piece of malicious software that is activated during boot and can thus effectively hide from and control the operating system. 12
- Sender Policy Framework** Sender Policy Framework is a protocol that allows the owner of a domain to specify which servers are allowed to send emails on behalf of that domain. 20
- Shell** A shell is a command-line interface for access to the services of an operating system. 15
- Shoulder Surfing** Shoulder surfing is the act of looking over the target's shoulder while the target is entering a password or code. 11
- Social Engineering** Social engineering is the act of manipulating an entity into doing something the attacker wants the entity to do, but which may be detrimental to the entity itself or any entities associated with the attacked entity. 5
- Tailgating** Tailgating refers to the technique of entering a restricted area without authorization by following another employee through a door. 10
- Trojan** A trojan is a program that seemingly fulfills a legitimate purpose but secretly executes malicious code. 12
- Virtual Private Server** A virtual private server is a virtual machine that can be bought or rented as a service. 23
- Vishing** Vishing is a combination of the terms “voice” and “phishing” and describes an attack that works similar to phishing, but over the phone. 8
- Vulnerability** In the context of computer software, a vulnerability is a weakness in the code that allows attackers to perform unauthorized actions. 15
- Water Holing** Water holing is a technique whereby social engineers take advantage of the trust users have in websites they visit often. 9
- Whitelisting** A whitelist is a list of all the entities who are allowed to perform a particular action and blocks everyone else automatically. 20
- Worm** A worm is a self-replicating piece of software whose goal it is to spread as fast and wide as possible. 12
- Zero-Day Vulnerability** A zero-day vulnerability is a vulnerability that is unknown to the parties interested in patching the vulnerability. 15

List of Mitigations

1	Avoid Publishing Contact Details	45
2	Spot a Phishing Email	48
3	Handle Links and Attachments Suspiciously	49
4	Mistrust Unusual Email Addresses	68
5	Encrypt Data for Transmission	69
6	Secure Your Domain	101
7	Disclose Information Prudently	119
8	Apply a Badge and Visitor Policy	124
9	Secure the Wireless Network for Guests	125
10	Terminate Malevolent Customers	135
11	Avoid or Limit Out-of-Office Replies	135
12	Educate and Train All Employees	136
13	Protect Information	137
14	Have the Courage to Address People	137
15	Counteract Tailgating	139
16	Prevent Dumpster Diving	148

Bibliography

- [1] Symantec: Internet Security Threat Report. **22** (2017) 10
- [2] IDC: IDC-Studie: 67 Prozent der Unternehmen in Deutschland von IT Sicherheitsvorfällen betroffen. <https://idc.de/de/ueber-idc/press-center/65936-idc-studie-67-prozent-der-unternehmen-in-deutschland-von-it-sicherheitsvorfaellen-betroffen> (July 2018) [Online; accessed 2018-07-24].
- [3] US Airforce: Hack the Airforce. <http://www.af.mil/News/Article-Display/Article/1163923/do-you-have-what-it-takes-to-hack-the-air-force/> [Online; accessed 2018-06-13].
- [4] Pentagon: Hack the Pentagon. <https://www.defense.gov/News/Article/Article/710033/hack-the-pentagon-pilot-program-opens-for-registration/> [Online; accessed 2018-06-13].
- [5] Jeremiah Grossman: Hack Yourself First: National and Economic Security. <https://www.whitehatsec.com/blog/hack-yourself-first/> (December 2014) [Online; accessed 2018-06-13].
- [6] Samani, R.: Hacking the Human OS. Presentation
- [7] Cialdini, R.: Influence: The Psychology of Persuasion. Collins Business Essentials. HarperCollins (2009)
- [8] Wikipedia contributors: Social Engineering (Security). [https://en.wikipedia.org/wiki/Social_engineering_\(security\)](https://en.wikipedia.org/wiki/Social_engineering_(security)) [Online; accessed 2018-06-12].
- [9] KOYUN, A., Al Janabi, E.: Social Engineering Attacks
- [10] Lineberry, S.: The Human Element: The Weakest Link in Information Security. *Journal of Accountancy* **204**(5) (2007) 44
- [11] Sasse, M.A., Brostoff, S., Weirich, D.: Transforming the ‘Weakest Link’ — a Human/Computer Interaction Approach to Usable and Effective Security. *BT Technology Journal* **19**(3) (Jul 2001) 122–131

- [12] Nick Ismail: Why Employees Are a Business' Weakest Link - and How to Remedy That. <https://www.information-age.com/employees-businesses-weakest-link-123470435/> (January 2018) [Online; accessed 2018-07-31].
- [13] Arce, I.: The Weakest Link Revisited [Information Security]. IEEE Security Privacy **1**(2) (Mar 2003) 72–76
- [14] Fette, I., Sadeh, N., Tomasic, A.: Learning to Detect Phishing Emails. In: Proceedings of the 16th International Conference on World Wide Web. WWW '07, New York, NY, USA, ACM (2007) 649–656
- [15] Dhamija, R., Tygar, J.D., Hearst, M.: Why Phishing Works. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '06, New York, NY, USA, ACM (2006) 581–590
- [16] Krombholz, K., Hobel, H., Huber, M., Weippl, E.: Advanced Social Engineering Attacks. Journal of Information Security and Applications **22** (2015) 113 – 122 Special Issue on Security of Information and Networks.
- [17] Wikipedia contributors: Voice Phishing. https://en.wikipedia.org/wiki/Voice_phishing [Online; accessed 2018-06-12].
- [18] Yaniv Leviathan and Yossi Matias: Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone. <https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html> (May 2018) [Online; accessed 2018-07-17].
- [19] Google Duplex: Google Duplex Demo from Google IO 2018. <https://www.youtube.com/watch?v=bd1mEm2Fy08> (May 2018) [Online; accessed 2018-07-17].
- [20] Lyrebird: We Create the Most Realistic Artificial Voices in the World. <https://lyrebird.ai/> [Online; accessed 2018-07-17].
- [21] Wikipedia contributors: Watering Hole Attack. https://en.wikipedia.org/wiki/Watering_hole_attack [Online; accessed 2018-06-12].
- [22] Tischer, M., Durumeric, Z., Foster, S., Duan, S., Mori, A., Bursztein, E., Bailey, M.: Users Really Do Plug in USB Drives They Find. In: Security and Privacy, IEEE (2016)
- [23] Wikipedia contributors: Piggybacking (Security). [https://en.wikipedia.org/wiki/Piggybacking_\(security\)](https://en.wikipedia.org/wiki/Piggybacking_(security)) [Online; accessed 2018-06-12].

- [24] Merriam-Webster Online: Merriam-Webster Online Dictionary.
<http://www.merriam-webster.com> (2018)
- [25] Symantec: Internet Security Threat Report. **22** (2017) 11
- [26] Wikipedia contributors: Denial-of-Service Attack.
https://en.wikipedia.org/wiki/Denial-of-service_attack [Online; accessed 2018-06-12].
- [27] Wikipedia contributors: Man-in-the-Middle Attack.
https://en.wikipedia.org/wiki/Man-in-the-middle_attack [Online; accessed 2018-06-19].
- [28] Rapid7: Metasploit. <https://www.metasploit.com/> [Online; accessed 2018-06-12].
- [29] Offensive Security: Metasploit Unleashed.
<https://www.offensive-security.com/metasploit-unleashed> [Online; accessed 2018-03-14].
- [30] Wikipedia contributors: Metasploit Project.
https://en.wikipedia.org/wiki/Metasploit_Project [Online; accessed 2018-06-12].
- [31] Offensive Security: About the Metasploit Meterpreter.
<https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/> [Online; accessed 2018-06-12].
- [32] Offensive Security: Meterpreter Basic Commands. <https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/> [Online; accessed 2018-06-12].
- [33] Interference Security: ICMP Reverse Shell.
<http://resources.infosecinstitute.com/icmp-reverse-shell/> [Online; accessed 2018-03-15].
- [34] Wikipedia contributors: Vulnerability (Computing).
[https://en.wikipedia.org/wiki/Vulnerability_\(computing\)](https://en.wikipedia.org/wiki/Vulnerability_(computing)) [Online; accessed 2018-06-12].
- [35] Wikipedia contributors: Zero-day (Computing).
[https://en.wikipedia.org/wiki/Zero-day_\(computing\)](https://en.wikipedia.org/wiki/Zero-day_(computing)) [Online; accessed 2018-06-12].
- [36] Dübendorfer, T.: Lecture Slides of Malware, Network Security (October 2016)

- [37] Sukwong, O., Kim, H., Hoe, J.: Commercial Antivirus Software Effectiveness: An Empirical Study. *Computer* **44**(3) (March 2011) 63–70
- [38] Frei, S.: Lecture Slides of Botnets and Malware Development, Network Security (October 2016)
- [39] Frei, S., Dübendorfer, T., Many, D., Plattner, B., Perrig, A.: Lecture slides of Firewalls, NAT, and Intrusion Detection and Prevention Systems (IDS), Network Security (September 2016)
- [40] Wikipedia contributors: Greylisting.
<https://en.wikipedia.org/wiki/Greylisting> [Online; accessed 2018-06-12].
- [41] Levine, J.R.: Experiences with Greylisting. In: CEAS. (2005)
- [42] openspf.org: Sender Policy Framework - Introduction.
<http://www.openspf.org/Introduction> [Online; accessed 2018-06-02].
- [43] Eleanore Young: Securing Email of Your Own Domain.
<https://www.scip.ch/en/?labs.20171109> [Online; accessed 2018-07-20].
- [44] Matt Moorehead: How to Explain DKIM in Plain English.
<https://blog.returnpath.com/how-to-explain-dkim-in-plain-english-2/>
[Online; accessed 2018-07-20].
- [45] Matt Moorehead: How to Explain DMARC in Plain English.
<https://blog.returnpath.com/how-to-explain-dmarc-in-plain-english/>
[Online; accessed 2018-07-20].
- [46] swissinfo: Kriminelle telefonieren weiterhin anonym.
<https://www.swissinfo.ch/ger/kriminelle-telefonieren-weiterhin-anonym/6039442> [Online; accessed 2018-06-13].
- [47] Microsoft: ADV170021 — Microsoft Office Defense in Depth Update.
<https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/ADV170021> [Online; accessed 2018-03-15].
- [48] Thomas von Mengden: 15 Möglichkeiten, die E-Mail-Adresse geschützt darzustellen. <https://www.hosteurope.de/blog/15-moeglichkeiten-die-e-mail-adresse-geschuetzt-darzustellen/> [Online; accessed 2018-06-25].
- [49] LinkedIn: LinkedIn User Agreement.
<https://www.linkedin.com/legal/user-agreement> [Online; accessed 2018-05-23].

- [50] Estelle Derouet: 10 Tips on How to Identify a Phishing or Spoofing Email. <https://blog.returnpath.com/10-tips-on-how-to-identify-a-phishing-or-spoofing-email-v2/> [Online; accessed 2018-07-11].
- [51] Pierluigi Paganini: Swisscom Data Breach Hits 800000 Customers, 10% of Swiss Population. <https://securityaffairs.co/wordpress/68854/data-breach/swisscom-data-breach.html> [Online; accessed 2018-07-21].
- [52] Danny Bradbury: Sextortion Scam Knows Your Password, But Don't Fall for It. <https://nakedsecurity.sophos.com/2018/07/13/sextortion-scam-knows-your-password-but-dont-fall-for-it/> [Online; accessed 2018-07-21].
- [53] Ronald Eikenberg: Passwort-Mail: Erpresser machen schnelle Kasse. https://www.heise.de/security/meldung/Passwort-Mail-Erpresser-machen-schnelle-Kasse-4117115.html?wt_mc=nl.heisec-summary.2018-07-23 (July 2018) [Online; accessed 2018-07-24].
- [54] Brian Krebs: Google: Security Keys Neutralized Employee Phishing. <https://krebsonsecurity.com/2018/07/google-security-keys-neutralized-employee-phishing/> (July 2018) [Online; accessed 2018-07-24].
- [55] Dale Langley: Protecting Your Brand From Phishing: How to Create your SPF Record. <https://blog.returnpath.com/protecting-your-brand-from-phishing-how-to-create-your-spf-record/> [Online; accessed 2018-07-20].
- [56] Dale Langley: Protecting Your Brand From Phishing: How to Create a DKIM Record. <https://blog.returnpath.com/protecting-your-brand-from-phishing-how-to-create-a-dkim-record/> [Online; accessed 2018-07-20].
- [57] Tara Seals: Regional Virginia Bank Falls Victim to Coordinated \$2.4M ATM Heist. <https://threatpost.com/regional-virginia-bank-falls-victim-to-coordinated-2-4m-atm-heist/134425/> (July 2018) [Online; accessed 2018-07-31].
- [58] Perrig, A., Szalachowski, P., Reischuk, R.M., Chuat, L.: SCION: a secure Internet architecture. Springer (2017)

APPENDIX A

Code

Due to the length of the thesis and the limited space available, we do not print any code. However, those who wish to study the code are encouraged to request the files from the author.

Visit <https://yuublue.com>.

PDF Documents

On the following pages, the reader can find the PDF documents referenced in the text.

The original text was written in German, it is translated here for purposes of reaching a broader audience.

B.1 SED11: The Leaver

Name of Company	
Company Address	Personal-Nr. Personnel Number
Residential Address	

Lohnabrechnung April 2018

25.04.2018

Lohnposition	Faktor	Einheit	Ansatz	Einheit	Betrag
Bruttolohn					
Monatslohn					Gross Salary
EBT Management					Management Bonus
Bonus					Bonus
Total Bonus					Total Salary
Spesen und Zulagen					
Spesen pauschal					Expenses
Kinderzulage	Number of Children				Children's Benefits
Total Spesen u. Zulagen					Total Expenses and Benefits
Sozialabzüge					
AHV/IV/EO-Beitrag	Total Social Benefits				
ALV-Beitrag					
ALVZ-Beitrag					
NBUV-Beitrag AN					
KTG-Vers.-Beitrag					
BVG-Beitrag fix					
Total Sozialabzüge					Total Social Benefits
Total Auszahlungsbetrag					Net Salary

Der Auszahlungsbetrag wurde wie folgt überwiesen:

Zahlungsverbindung	IBAN / Konto	Betrag
Name of Bank	Bank Account Number	Net Salary

B.2 SED19: Orders from Above

Sculpture Brugger

Röntgenstrasse 16
8005 Zurich

Billing date	05/14/2018
Customer No.	1742/2018
Due on	06/14/2018

Invoice

Amount	Description	Unit Price	Total Price
5h	Erstentwurf Papier «Projext X»	CHF 90.00	CHF 450.00

Total CHF 450.00

Thank you for your trust.
Please transfer the amount to the account indicated below.

Kind regards,

John Brugger

Bank Account

B.3 SEP01: The Fake Invoice

Helping Businesses through Freelancing

Address of
Jonathan Steiger

Address of Target
Company

Invoice

Invoice No. 1652-5463

Customer No. 1652

Datum: 08/06/2018

Pos	Description	Single Price	No.	Price
1	Project Support	CHF 155.00	4 Stunden	CHF 620.00

The total amount has to be paid until
05/04/2018 to the account listed below.

Total (incl. VAT):

CHF 620.00

I thank you for your trust and I am looking forward to support you again.

Sincerely,
Jonathan Steiger

Address of
Jonathan Steiger

Bank Account

B.4 SEP02: The Security Interview

Survey on «Computer Security»

The survey is anonymous and will be used in my Master thesis. If for any reason you do not want to or cannot answer a question, please just say so.

How many e-mail addresses, online accounts, banking accounts, etc. do you have? _____

Do you use different passwords for different accounts, or do you use the same password in several places?

Different passwords

The same password

Which operating system do you use at work? _____

Which operating system do you prefer to use privately? _____

Do you regularly install updates for your programs, operating systems, etc.? Yes No

Do you use an antivirus at work? Yes No If yes, which one? _____

Do you use an antivirus privately? Yes No If yes, which one? _____

Are you concerned about privacy in social networks? Yes No

On a scale from 1 to 10, how liberal do you handle your data on social networks? _____

What do you say about 2-Factor-Authentication? Good Appropriate Too annoying

If possible/privately, do you use 2-Factor-Authentication? Yes No

Please think of the password now that you mainly use here at [\(Target Company\)](#).

Do you also use your company password with other accounts outside the company?

Yes

No

How do you like the company's password policy? Too strong Appropriate Too weak

On a scale of 1 to 10, how secure do you rate your password? _____

How many characters does your password contain? _____

Does your password contain lowercase letters? Yes No If yes, how many? _____

Does your password contain capital letters? Yes No If yes, how many? _____

Does your password contain numbers? Yes No If yes, how many? _____

Does your password contain special characters? Yes No If yes, how many? _____

How often do you change your password? _____

Do you change your password because it is mandatory or because you want to?

Mandatory

Because I want to

Do you save your passwords in your browser? Yes No

Do you use a password manager? Yes No

If yes, which one? _____

Do you write down your passwords? Yes No

If yes, where? _____

How do you invent your passwords? Do you use a scheme (e.g. first letter of each word in a sentence), a generator, or something else?

Scheme Generator Other

Have any of your passwords ever been stolen/abused? Yes No

How did this happen? _____

Finally, a few questions about social engineering.

Do you know the term "social engineering"? Yes No

Have you ever been the victim of a social engineering attack? Yes No

On a scale of 1 to 10, how resistant to social engineering do you rate yourself? _____

How do you protect yourself against social engineering? _____

In your opinion, who is responsible for fending off hacker attacks (e.g. phishing) and protecting sensitive data?

Employer Employee Both

With this survey, I want to raise awareness for digital security. I would like to send you the same questionnaire online in three months to see if your behaviour towards cybersecurity has changed. Would you like to join?

If yes, contact e-mail address: _____

If not, can I send you the results of the survey in three months?

If yes, contact e-mail address: _____