universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

A Nexus of Vulnerabilities

Infested Code and the Prospects of Coordinated Vulnerability Disclosure

verfasst von / submitted by

Marc Čudlik, BA

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Arts (MA)

Wien, 2021 / Vienna, 2021

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet::	UA 066 906
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet::	Masterstudium Science-Technology-Society
Betreut von / Supervisor:	Univ. Prof. Sarah Davies, Bsc Msc PhD

It is at least my thesis,
so I want the spirit to speak.
[…]
The mysterious jinx and
the error in heavens' masterplan.

The Enigmatic Spirit - Vintersorg

Acknowledgments

What a beautiful tradition humankind has evolved with pages like these. How to make a selection of all the people accompanying one toward and until this point in time?

Thanks to my supervisor, Prof. Sarah Davies. For your always friendly support, accessibility and taking out the time when I needed it most. But also for your valuable insights and giving me the freedom I hoped for. Thank you very much for your understanding!

Thanks to the Deputy Head of Department of the STS department, Maximilian Fochler. You truly are the good and guiding spirit I needed while studying in this program. Thanks for keeping up with my flaws, providing an address regarding quite some concerns and being an ever good sport about it.

Thanks to my interview partners. Presupposing trust while never having heard of someone is quite a step. Especially in the field you're working in. Thank you for taking the time and giving me a glimpse in your lives and the world(s) you inhibit. Also, that has to be said, thank you for your readiness to take up the dreadful prospect of doing an online interview. Last but not least, thanks for all you do. Not just for me, for all of us.

Thanks to my superior at my day job, Pia. Your part in providing me the chance to study during work shall never be underestimated. Thanks for giving me this chance and enable me to follow my, in all aspects of the word, academic interests. Thanks for shifting around shifts. Quite shifty!

Thanks to all my family & friends. Keeping me sane in these strange times as well as providing the year long support of my academic career is nothing to sneeze at. Thank you all for pushing me, for your patience and, well, love.

Thanks to Maren. For proofreading, comments and providing general QA.

Last but not least…

… thanks to Veronika. For your relentless support, being the dump of my sorrows and stalwart foundation. For your understanding, opinions and affection. Without you, I probably wouldn't even have applied for this program again. Now see where we're at. You truly are a good *patpat*.

From the bottom of my heart, thanks to all of you.

Table of Contents

Acknowledgments ..... **v**

Table of Contents ..... **vii**

Introduction **1**

1. Context **4**

1.1 Three Dimensions of Vulnerability 4

1.2 Bug Bounty Programs & Coordinated Vulnerability Disclosure – A Specification 6

1.3 Parties, Protagonists and Organizations 9

1.4 A Complex Ecosystem 12

2. A Mobilization of Concepts **15**

2.1 On Science, Knowledge and Technology 15

2.1.1 Knowledge and the Processes of Science 15

2.1.2 Scientific Controversy, Closure and the Social 16

2.1.3 From the Laboratory to Inscription 16

2.2 Inscription, Agency and Ontology 17

2.2.1 Inscription and Politics 17

2.2.2 The Problematization of the Social 18

2.2.3 Assemblages and Ontology 19

2.3 Digital Technologies 20

2.3.1 Inscription and Transformation in Digital Devices 20

2.3.2 Performativity 21

2.3.3 From Matters of Concern to Matters of Care 22

3. Research Questions **25**

4. The Case Assembly: Methodology & Materials **27**

4.1 An Introduction to the Research Field 27

4.2 Methodology 29

4.2.1 Document Analysis 29

4.2.2 Qualitative Interviews 30

4.2.3 Finding Interview Partners 32

4.2.4 Ethical Considerations 33

5. Analysis....36
5.1 Intermezzo – A Need for Clarification....36
5.1.1 Databases, Weaknesses and Resources....36
5.1.2 Software Security, Cybercrime and Cyberwar....38
5.2 Before Bugs – Starting Locations....42
5.2.1 Expectations....42
5.2.2 Feelings....42
5.2.3 Intentions....43
5.3 Finding Vulnerabilities....44
5.3.1 Mindsets and Corporeal Engagements....44
5.3.2 Practice and Enactment....45
5.4 Contact....47
5.4.1 Finding a Point of Contact....47
5.4.2 Resources in Reporting....48
5.4.3 Judgment Calls....48
5.5 The Triage Process....50
5.5.1 Mediating Receivers....50
5.5.2 Ontology, CVSS and Actionability....50
5.5.3 Stabilization through Communication....51
5.6 Social Tokens....52
5.6.1 Supply Chains, Code Bases and Trust....52
5.6.2 Trust and Legal Issues....54
5.6.3 Politics....58
5.7 Morality, Incentives and Markets....60
5.7.1 Help and Incentives....60
5.7.2 Public disclosure/Non-disclosure....62
5.7.3 Reputation....63
5.7.4 Compliance....64
5.7.5 Markets....65
5.8 Maturity and Management....66
5.8.1 Commitment....66
5.8.2 Hierarchy of Maturity....68
5.8.3 Precedence and Guidelines....70
5.8.4 Future of Vulnerability Research....71

6. Conclusion....74
6.1 Software Vulnerabilities in CVD....75
6.2 CVD in ANT....75
6.3 Software Vulnerabilities in CVD....76
6.4 CVD as “Matter of Care”....76
6.5 Further Research....77

Bibliography....79

Appendix....86
Abstract (english)....86
Abstract (deutsch)....87

Introduction

Imagine a company responsible for 25% of the world's container transportation infrastructure. Thousand of vessels, terminals, harbors; millions of people, computers and singular container units with tons and tons of different industrial and consumer wares working in unison to produce a theater of what provides the basis of our daily lives. And then imagine a sudden halt to this well-oiled machine. The underlying grid of information, the logs and files where which container is supposed to go, when they will arrive and how many will go where, is suddenly gone. Locked behind a screen message which reads: "Ooops, your important files are encrypted." Everywhere you look, every single computer screen on the company's network shows the same message. You can't enter any inputs, you can't extract data or information, you can't look up anything, you can't contact anyone. You are alone with a useless piece of technology, now reduced to a pile of cables, chips, surfaces and buttons. The next four days: Ships and vessels dock in harbors, yet cannot attribute the wares to the trucks waiting to deliver them. The backlog grows longer, other companies and factories are waiting for their goods and raw materials to be delivered, new wares need to be distributed. All stands still.

This is what happened on a Tuesday, June 27 2017 to the Danish company A.P. Møller-Maersk. This incident lead to a financial loss for the company in the range of 250-300 million US Dollar and the need to re-install software on more than 45 000 computers and servers (Kovacs, 2018; World Economic Forum, 2018, min 2:53). What happened? A piece of software, more specifically a piece of malware called NotPetya, infected one singular company in the Ukraine. From there, it spread through all of their network and infecting all connected devices, making the jump in the wider internet and ultimately hitting hospitals, banks, postal services, food providers and construction companies all over the world alike, resulting in estimated damages of more than 10 billion US dollars (Greenberg, 2019, p. 199).

NotPetya is built from two essential component: Mimikatz, a tool built to obtain account credentials and their respective passwords to gain access to otherwise inaccessible parts of the computer. The second part is EternalBlue. This vulnerability was first discovered by the National Security Agency

(NSA) of the United Stated of America, before the technology was stolen and leaked by a hacker group called “The Shadow Brokers”. After the technology was stolen, the NSA told Microsoft about the vulnerability. Microsoft built a fix, rolled out a patch and, even though this patch was available at the time NotPetya came around, NotPetya caused a lot of damage (Schulze & Reinhold, 2018, p. 460).
Not all instances were software vulnerabilities show their potential disrupting global supply chains. However, if this is a *potential* effect of vulnerabilities in software products, it is worth the question of what can be done about them? Software products or things that rely on software are ubiquitous and our lives are deeply interconnected with the functioning of these things. Therefore, the question employed in this thesis rests on one of the the strategies employed to mitigate software vulnerabilities: Coordinated Vulnerability Disclosure. This process is commonly described as facilitating an orderly communication between the finders of software vulnerabilities and the respective companies or people deemed responsible for fixing those vulnerabilities. This process has several instances, going from the finding of the vulnerability to writing up a report to examining the contents of this report to finally accepting or declining the report. All of those steps, however, are quite complicated practices, bringing together not only people but technologies, intentions, perspectives and a whole bunch of documents, standards, laws, policies and decisions. To understand what resources are needed to maintain this mitigation strategy, I will take a closer look at the software vulnerabilities in the context of the CVD process.
The first section will introduce the context of this thesis, the basic principles of software vulnerabilities. This section will also introduce the CVD process, describing it with the help of a similar, yet distinct, mitigation strategy. Also, parties and actors involved in this process will be discussed. The final aspect to this section will be an overview of literature and documents which deal with software vulnerabilities.
The second section deals with the theoretical framing applied in this thesis. Since this thesis is written in Science-Technology-Studies, I will give a short overview over its historical roots. The basic concepts of Actor-Network Theory, the theory applied, will be introduced. Lastly, the theoretical concepts will be discussed in the context of the digital sphere.
The third section is the introduction of the very research question guiding this thesis. Also, some subquestions and the intention behind them will be explained.
In the fourth section I will discuss the broader research fields, the methods applied and materials gathered for this thesis. Also, an account of how I went about finding suitable interview partners as well as what ethical considerations I had will find their place here.
The fifth part, at last, will discuss the findings of my research. Starting with a clarification of two topics which didn’t fit particularly well in any other chapter, I will systematically go through my findings.

Lastly, in the sixth section, I will conclude this thesis by reviewing the steps taken, contextualize my findings and open up further possible pathways in researching or thinking about software vulnerabilities.
After the conclusion, of course, you will find the bibliography. A list of all the documents, books, standards, academic papers, videos, websites and other kinds of resources that went into the making of this thesis.

I hope you enjoy reading this thesis and find some new and interesting ideas.

1. Context

1.1 Three Dimensions of Vulnerability

When discussing software vulnerabilities in an academic setting, there immediately come to mind an avalanche of questions. The first and probably most fundamental ones are: What are “vulnerabilities”, what is “software” and why does this question even matter?

The simple question of what “software” is points towards the very basics of technological devices we encounter on a daily basis. Obviously, software is something that is inherent in every personal computer, mobile phone and internet-enabled device. Devices which carry software in them or are somehow reliant on software are ubiquitous. We may think of computers as the “classical” home computer, where one answers their e-mails, plays games or writes homework. We may think of them as the mobile phones we use for getting the latest news while on the road or watching memes in bed. But not only the digital devices themselves are ubiquitous, there is a concerted effort to be seen to integrate those digital devices into an interconnected network. Everything from our homes, our industries, hospitals, governments and other institutions, the very basis of what makes up our lives, make use of software, digital devices and the internet. There are the apparent layers of the internet such as big websites: Almost everyone is familiar with Facebook, Twitter, Google, YouTube, Airbnb or Netflix. And then there are the more hidden or “non-obvious” layers to the internet, servers which transport data, large storage facilities which serve as “the cloud” for our databases, streaming services or other high volume needs and a myriad of different devices necessary to sustain our lives.

A big part of the devices being part of the internet nowadays can be subsumed under what can be called the “Internet of Things” (IoT), the millions upon millions of distributed cameras, sensors, computers and machines which measure temperature, connect and distribute information, handle requests, guide packages and so on. Possibly the most famous of these devices are the so-called home assistants, Google’s home assistant software Nest, Amazon’s Alexa and Apple’s Siri. Also part of this distributed network are devices such as fridges that check its contents and possibly automatically order missing items, (surveillance) drones and even sex toys which are connected to the internet subsumed under the *terminus technicus* “teledildonics” (e.g. Dickson, 2020). Of great concern in recent years is also the upcoming trend of what is commonly called “Industry 4.0”, the digitalization of industrial

production processes. With that, small sensor arrays may control the amount of liquids traveling through pipes in a factory, measure the pressure materials are subject to in other industrial contexts or count the amount of products being collected in a packaging unit. Those sensors and machines are subsumed under the term “Industry Control Systems” (ICS).
Why is that a potential problem? Generally, software vulnerabilities (also called “bugs”) are described as being just “[…] a fact of life. You can’t opt out.” (HackerOne, 2020, p. 10) They can be found in every piece of software there is. How many there are is quite difficult to say. There are three dimensions to that problem:

1. The *code base*, the many pieces within *one* device. This first problem introduces the contemporary production practices of software and computers, pointing to the many parties involved in creating on device:

> “Modern computer chips are typically designed by one company, manufactured by another and then mounted on circuit boards built by third parties next to other chips from yet more firms. A further firm writes the lowest-level software necessary for the computer to function at all. The operating system that lets the machine run particular programs comes from someone else. The programs themselves from someone else again. A mistake at any stage, or in the links between any two stages, can leave the entire system faulty - or vulnerable to attack.”
> (The Economist, 2017b)

2. The amount of *connected* devices. This second problem relates to the emergence of the internet as “evocative structure” (Shah, 2012), which increases the connections between those devices exponentially. There are estimations that by this year (2021), there is will be 46 *billion* connected devices, an increase of 200% to 2016 and with an outlook to there being as much as 125 *billion* by 2030 (Galov, 2021).

3. The *density* of vulnerability. This third dimension is the question of density. Similar to Ozment & Schechter in their paper on “Milk Or Wine: Does Software Security Improve with Age?” (Ozment & Schechter, 2006, p. 1), Bruce Schneier in an article in The Atlantic opened this Pandora’s Box with their question if vulnerabilities in software are sparse or dense (Scheier, 2014).

What we can say for sure is that there are estimations such as “[…] that the average programme has at least 14 separate points of vulnerability” (Schaake et al., 2018, p. 1). Which is a big problem if we consider that something like for example the Windows operating system has millions and millions of lines of code (House, 2016, p. 26). Google is said to have 2 billion lines of code in their products (The Economist, 2017a). In general, we can say that “[s]security bugs will always exist as long as humans write software” (House, 2016, p. 26). Even though there are debates if machines themselves are

capable of writing code with software vulnerabilities in them as well (cf. Geer, 2014). Taken together, I can say that vulnerabilities are quite a mundane thing, they seem to be part of the world we live in, brought about by the very software that surrounds us. So, let me introduce, at last, three definitions as I have found them in the literature consulted on what vulnerabilities are:

> "[F]unctional behaviour of a product or service that violates an implicit or explicit security policy."
> (ISO/IEC, 2018, p. 7)

> "A weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability."
> (NIST n.d. n.p.)

> "Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source."
> (Joint Task Force Interagency Working Group, 2020, p. 423)

That seems quite straightforward. As I have (hopefully) established by now, software vulnerabilities are ubiquitous, rely on technology which connects not only technical devices but are the basis for all our lives, and are trusted upon. They are everywhere and they can be exploited. When code is published in the form of products or services, however, there are only a few things a company can do to test and remedy potential vulnerabilities. Since the affected code segments are already "in the wild", the question changes from "Why are there vulnerabilities in the first place?" to *"*What can people do about them?".

1.2 Bug Bounty Programs & Coordinated Vulnerability Disclosure – A Specification

To understand what is commonly understood as CVD[1] it may be of value to contrast this strategy to other ways for companies to deal with software vulnerabilities. The pathways to find and remedy vulnerabilities are manifold. There are recommendations on the part of the developers so that "insecure" code doesn't even get published. These reach from implicit security requirements, practices of secure software engineering, secure software development cycle to respecting and introducing existing standards and good practices (ENISA, 2019a, p. 6 f.). Relevant here is that these recommendations are related to strategies a company can implement *in house*: "Code should be built, tested, integrated, maintained, and updated with security aspects in mind." (ENISA, 2019b, p. 15) This can mean teaching software engineers how to follow security aspects while coding, having feedback

1 Also called "Responsible Vulnerability Disclosure", how this is seen as ambiguous term, especially since it has a normative connotation of being 'responsible' and interpretation varies from stakeholder to stakeholder." (ENISA & RAND Europe, 2015, p. 24)

loops in place which ensure that mistakes made are being recognized and inform previous production segments to *having* standards in place, knowing what they are (cf. Votipka et al., 2018) and how to implement them in the respective products, code sections, databases, systems and so on many strategies can be applied to make code more secure or harden products.
A popular way for companies to open up the possibility for vulnerabilities to be found are so called Bug Bounty programs. In this case, to be clear, "programs" here refer to social institutions (similar to "university programs"), not programs in terms of software. These programs are either run by companies themselves or they are offered as a service by specialized companies. What makes bug bounty programs special in terms of vulnerability mitigation is that they...

- ...have a specific scope (e.g. website, program, product, database,…)
- ...employ a strict set of rules of what is considered worth of a reward
- ...pay a monetary reward.

The amount of money being paid for specific software vulnerabilities reflects their "criticality". This criticality is determined by the so-called "triage" process. These examination are often done by employing a standardized test, called the "Common Vulnerability Scoring System" (CVSS) (FIRST, n.d.). This metric seeks to relate information on the affected systems, known attack vectors (cf. Storm et al., 2020) and a time component into a single value between 1 and 10. "Attack vector" describes all the potential routes an attacker can take to compromise their target. These "incentives" for researchers, as the bounties are sometimes called, are not uniform across the industry and are subject to change over time (Finifter et al., 2002; Laszka et al., 2016; Zhao et al., 2015). This number is commonly understood as a pointer towards criticality, it serves as an indicator for how much a bug may be worth in bug bounties (Bugcrowd, n.d.), what possible payout there may is. The more critical, the higher the pay. What kind of information goes into and what should be considered relevant for this examination process, however, is sometimes not quite clear (Allodi et al., 2018). Relevant to add at this point are "0day" or zero-day vulnerabilities. These vulnerabilities are not (yet) publicly known and pose a significant challenge, since between the time they get known and the time needed to develop a mitigation strategy (patch them) these vulnerabilities may be exploited. This result in a situation whereas the developers of a patch are given zero days to respond, hence, zero days (this is just one definition among many, others may be found in Householder, 2015).
There are a few different models of bug bounty programs, depending on who is able or allowed to participate. As the bug bounty company HackerOne puts it:

> "A public program allows any hacker to participate for a chance at a bounty reward. A private program limits access to select hackers who are invited to participate. Focused programs can also be time-bound, or run as virtual or in-person live events."
> (HackerOne, 2020, p. 8)

Even though it is difficult to estimate the total amounts of programs being run by companies across the industry, an insight may be that one of the biggest companies, HackerOne[2], asserts they have had more than 2000 so far (HackerOne, 2021, p. 12).

To summarize, I can say that Bug Bounty programs invite people to look for vulnerabilities in a specific piece of software with the prospect of a monetary reward. In contrast to those, the CVD is slightly different. This process is characterized, as is found in a communication by the European Commission on Resilience, Deterrence and Defence in cyberspace, for example as follows:

> "Coordinated vulnerability disclosure is a form of cooperation which facilitates and enables security researchers to report vulnerabilities to the owner or vendor of the information system, allowing the organisation the opportunity to diagnose and remedy the vulnerability in a correct and timely fashion before detailed vulnerability information is disclosed to third parties or the public."
> (EC, 2017, p. 6 [footnote])

As is described here, the CVD process is a unilateral process, going from the finder of a vulnerability to the vendor, resulting and in a patch and thereby resolving the vulnerability. Generally, the contact possibility is not, as with the bug bounty programs, a company or defined program but to include a dedicated point of contact in the companies' public appearance (website). This can be an e-mail address (quite common are security@<companydomain> configurations) or a public text file in the domain. There are efforts to standardize this public appearance, as can be seen with the introduction of the ".../security.txt" format (Foudil & Shafranovich, 2021). Introduced in 2017, this public file format is currently in the 12th draft form *en route* to become a RFC, a "Request for Comment", which is an internet standard developed and maintained by the Internet Engineering Task Force (IETF).

The definition put forward by the EC, however, doesn't take a look at what vulnerabilities are (what is considered a vulnerability by vendors/researchers), what a reward could look like or why someone should report vulnerabilities in the first place. In my understanding, CVD has many different stages, actors and possible constellations between actors. Also, this process is not necessarily "unilateral" in the sense of reporting, having one reporter (or finder) and one recipient (or vendor). This process can take on quite a lot of forms, including many different parties and levels of involvement (FIRST, 2020). Also, what happens after the company rolls out a patch is subject to negotiation and debate. The two extremes are between full public disclosure, that is, making the vulnerability public on a website or

2 www.hackerone.com

forum post. Here is the potential problem that if not all affected devices or services are patched already, they are potentially endangered. Since the knowledge of the vulnerability in this instance is public, also the potential to misuse the vulnerability information is public. The other form is non-disclosure. There are two aspects to that. The first one being that the vulnerability doesn't get reported at all and is being kept secret, even from the company producing the device or offering a service. This may have negative consequences since there is always the chance that a vulnerability can be found by someone else at a later date. The other form is to report the vulnerability, but afterwards not disclosing the information about what was vulnerable at some point. That may be the case to not harm the reputation of a company through admitting that it had a security issue or because the information shouldn't be made public because of other reasons. To withhold this information then can't inform other researchers, vendors or producers that a specific practice may result in potentially vulnerable products, services or software.

Therefore, this definition somewhat leaves out a whole lot. A more comprehensible explanation of what goes into CVD and how complex this process is can be found in the introduction of a standard as is proposed by the International Organization for Standardization (better known with their abbreviation, ISO) together with the International Electrotechnical Commission (IEC) on CVD:

> "This document describes vulnerability disclosure: techniques and policies for vendors to receive vulnerability reports and publish remediation information. Vulnerability disclosure enables both the remediation of vulnerabilities and better-informed risk decisions. Vulnerability disclosure is a critical element of the support, maintenance, and operation of any product or service that is exposed to active threats. This includes practically any product or service that uses open networks such as the Internet. A vulnerability disclosure capability is an essential part of the development, acquisition, operation, and support of all products and services. Operating without vulnerability disclosure capability puts users at increased risk."
> (ISO/IEC, 2018, p. vii [Preface])

Comparing this definition with the previous one by the European Commission, we can clearly see that the focus is not on the parties involved and the quick remediation of a vulnerability, but the *handling* ("receive", "publish"), *impact* "("risk", "critical", "exposed", "threat", "capability", "essential") and *function* ("enables", "decisions", "support", "maintenance", "operation", "development", "acquisition", "support") of vulnerability disclosure processes.

1.3 Parties, Protagonists and Organizations

When researching the CVD process, one stumbles upon a lot of different definitions of the parties involved. In this chapter, I will introduce how these parties are talked about, what their relations are being presented as and how that may inform this thesis.

The first party I want to introduce are the people searching for vulnerabilities. In the literature consulted, they are called many names and are not easily subsumed. They may be called "software engineer", stemming from the very basic observation that they program, code or otherwise deal with software and its development, handling, tinkering or manipulation. However, the position in regards to CVD is not necessarily clear from this description alone since they may be part of a company and develop software or are the ones finding vulnerabilities when products or services are deployed. Similarly, (cyber-)security professional is a somewhat diffuse term. However, that term is quite often used in conjunction with people having some kind of education in security related topics. There are quite a lot of different certifications related to security. One of such certifications is, for example, the "Offensive Security Certified Professional" (OSCP), offered by the company Offensive Security[3]. Interestingly enough, they describe this certification as "[…] a penetration testing (or ethical hacking) training course designed for information security professionals." (Offensive Security, n.d.)

Which brings me to the next description, namely "hacker". This term, although quite well known and often used, sometimes eludes specific definitions. Sometimes, "hackers" are said to be "[…] independent security researchers, so-called white hats […]" (Laszka et al., 2016, p. 161) Some other "definitions" can be found in the literature consulted. They stretch, however, between "friendly hackers" (Elazari Bar On, 2019, p. 231), "*white hat researchers*" (Zhao et al., 2015, p. 1105 [emphasis i.o.]), "expert freelancers known as "white-hat hackers" (Votipka et al., 2018, p. 374), "enterprising security researchers and criminal hackers" (Weulen Kranenbarg et al., 2018, p. 1) to "adversaries" or "unauthorized individuals" (Woszczynski et al., 2020, p. 1). "White hat" refers to a commonly understood differentiation between "white hat hackers" and "black hat hackers". In this distinction, there is a moral judgment included, mostly related to the moral stance, application of knowledge about vulnerabilities/exploits or stance in regards to the law, as can be in the discussion surrounding "grey hats" (Kilovaty, 2017, p. 483). For this thesis I will therefore refrain from using that term as it clearly is too diffuse and may be misunderstood. In general, Matwyshyn et al. possibly state it best when they say "[t]he exact definition of vulnerability research and who counts as a 'vulnerability researcher' is subject of debate in the academic and business communities." (Matwyshyn et al., 2010, p. 67). So, for this thesis I will try to keep with "researcher", "finder" or "reporter", as those relate simply to their function in the CVD.

A second big group of actors are the affected parties. Here, the descriptions range in a similar fashion from "organisations" (ibid. p.2), "vendors", "owner" or "software developers" (Schulze & Reinhold, 2018, p. 454; Weulen Kranenbarg et al., 2018, p. 1), "companies", "defenders", "businesses", and "governments". Generally speaking, in this thesis I will refer to the affected parties mostly as "vendors" or "companies". There are two reasons for this. First, "affected parties" may be completely

3 https://www.offensive-security.com/

other entities than the actual "producers" of software (which would be more precisely defined by "vendor", if they sell their software). Here it is important to remember, however, that software may be "produced" by singular programmers in their leisure time, as a collective effort as "open source" projects or as the results of machines programming. Secondly, "companies" is a somewhat imprecise term in the sense that there exists "Cybercrime-as-a-service", as it is called in the European Security Union Strategy by the European Commission (EC, 2020, p. 3). Botnets, "hacks", vulnerabilities and many other products and services can be found online. This, in turn, somewhat muddies the waters in regards to who and what is subsumed under the mere term "company" (for further reading on "white", "gray" or "black" markets see Libicki et al., 2015, p. 44ff.). In my research, however, I will keep the term "company" as a receiver of vulnerabilities and "defending" entity.

This brings me to yet another relevant party in this ecosystem: The Information Technology (IT) departments of the affected parties, most often than not the receivers of vulnerability reports. Here, it is necessary to say that there exist the so-called Computer Emergency Response Teams (CERTS), sometimes Computer Emergency Readiness Teams (US CERT, n.d.). These teams are responsible for "[…] analyzing and reducing cyber threats and vulnerabilities, disseminating cyber threat warning information, and coordinating incident response activities" (ibid., p.1). Since they handle whatever "incident" there may be, they are sometimes also called Computer Security Incident Response Teams (CSIRT). There doesn't seem to be any organizational limits to where one may encounter such a team, as they "can be created for nation states or economies, governments, commercial organizations, educational institutions, and even non-profit entities" (Ruefle, 2007). As vulnerabilities are, as stated before, also found in products, some production companies have a distinct *Product* SIRT, which results in the acronym PSIRT. These units or organizational parts are sometimes themselves organized in even bigger structures. They then form, for example, "Information Sharing and Analysis Centers" (ISACs) or "National Cyber Response Coordination Group (NCRCG)" (US CERT, n.d.). ISAC and NCRCG are national organisations, yet there exist also *global* institutions such as, for example, the Forum of Incident Reponse and Security Teams[4] (FIRST).

The third actor group I actually have already introduced: The specific companies running the bug bounties. They decidedly form a distinct party as they, as previously discussed (see 1.2 Bug Bounty Programs & Coordinated Vulnerability Disclosure – A Specification) take on the function of a mediator between researchers and the vendors. Their function lies in the service of dealing with vulnerability reports, the communication with the finders/researchers and the handling of bounties. Since they often have established relations with researchers as shown with the "invitation" of trusted researchers to private programs as well as the need for researchers to register themselves on platforms, it is possible to get an idea on how big those platforms can get: HackerOne, as one of the bigger ones,

4 www.first.org

assert that "[s]ince the release of the 2019 Hacker Report two years ago, the HackerOne community has doubled in size to over one million registered hackers." (HackerOne, 2021, p. 2) Other big companies are for example BugCrowd[5] or Intigriti[6].

As we have already seen, governments and their departments are quite an important actor in this ecosystem as well. Not only the sources provided up until now point towards a strong involvement, but also just their function of issuing policy documents, guidelines and, more general, formulating laws.

A last group of actors I will mention are universities, non-profit organizations and standardization organizations. Some of them we already encountered, such as the ISO/IEC (ISO/IEC, 2018) or the FIRST (FIRST, 2020). Both of them issue standards and guidelines. The same goes for university institutes and governments. Therefore, the distinction between how these entities come into play may overlap, interact or oppose each other. If this is the case in this thesis I will point it out.

1.4 A Complex Ecosystem

Having introduced the main context, the difference between CVD and Bug Bounties and a definitely by no means exhaustive list of actors, I can turn towards the sites, places and discourses we may encounter vulnerabilities in documents. As with probably every research, there is a plethora of various arenas, realms and topics connected.

The first of such topics that I want to mention in this research, however, concerns recent discussions of the implementation of CVD processes across a variety of countries. This relates to the timeliness of this study. Just a few months back, in September of 2020, the United States Department of Homeland Security issued a "Binding Operational Directive" to their agencies to implement a CVD within the following 180 days (U.S. Department of Homeland Security, 2020). This is an order to implement a process to receive potential vulnerabilities and has to be implemented. The Chinese government (the Ministry of Industry and Information Technology Network Security, MIITNS) followed in July of this year and released a similar provision (MIITNS, 2021).

Within the European Union there is currently no united effort to implement something similar, however, the European Telecommunications Standard Institute issued a draft document for the standardization of CVD (ETSI, 2021). But even before those direct commandments or instructions there have been many discussions and recommendation. The U.S. Department of Justice's (DoJ) Cybersecurity Unit formulated a framework already back in 2017 (DoJ, 2017). The European Union Agency for Network and Information Security (ENISA) also issued a "best practice guide" for CVD (ENISA & RAND Europe, 2015). As we can see, there is a lot of movement in the realm of policy

5 www.bugcrowd.com

6 www.intigriti.com

documents regarding CVD specifically. But these frameworks and guidelines are informed and are based in many cases on broader strategies in the realm of cyber security.

In the European Union, this can be observed within the context of the document "European Security Union Strategy" (EC, 2020), providing an umbrella for all aspects to life deemed relevant to security. Regarding the CVD, this is further specified in the cyber security strategy (EC, 2017). These strategies are in many cases informed by scientific studies (ENISA, 2018, 2019a, 2020; High Level Group of Scientific Advisors, 2017). Studies which interrogate experts on cyber security or vulnerability discovery in particular are also done by Bug Bounty programs themselves (HackerOne, 2020, 2021) or companies advising governments or businesses (ISC2, 2020; Libicki et al., 2015).

In regards to the specificity of CVD, there can be found studies on quite a variety of topics. Johnson et al. discuss the time between the discoveries of vulnerabilities (Johnson et al., 2016). This becomes relevant if someone wants to exploit a vulnerability in a system for whatever reason. If a vulnerability can be found by one person, it can as easily be found by someone else with potentially different interests as well. Woszczynski et al. argue the need for a comprehensive framework for CVD within the context of the U.S. Emergency Alert Systems because of previous misuse (Woszczynski et al., 2020). On the example of what would happen if there is a zombie apocalypse and no one gets alerted because of a system failure which could have been mitigated they discuss legal challenge in the processes of vulnerability disclosure.

Quite a few studies introduce models of the global vulnerability discovery ecosystems from a variety of different perspectives. Here, game theory models can be found (Schulze & Reinhold, 2018), system dynamic approaches (Lewis, 2017), models between attacker and defender motivations (Moore et al., 2010) or meta-studies on, for example, information sharing in cyber security (Pala & Zhuang, 2019). Game theory seems to be a very strongly investigated topic in this realm, pointing towards the different knowledges and expectations in this realm.

While all of these papers are present in the *ecosystem* of vulnerabilities, what g*overns* CVD in particular are primarily aspects of what is considered hacking, law and the relation between the public and the private sphere. Laws such as the Digital Millennium Copyright Act (DMCA) (Digital Millennium Copyright Act, 1998) or the Computer Fraud and Abuse Act (CFAA) (Computer Fraud Abuse Act, 1986) are prime examples, not only (still) governing these process but also informing important decisions made within. Such decisions may call for "safe harbour" provisions for vulnerability researchers to shield them from legal persecution (Elazari Bar On, 2019; Kilovaty, 2017; Weulen Kranenbarg et al., 2018).

There are other instances where the line between "legal" and "illegal" may be crossed and is debated within this realm. There are debates surrounding the relation between knowledge and security (Aradau, 2017) as well as knowledge, users and cybercrime (Klimburg-Witjes & Wentland, 2021).

Not only the legal aspects in terms of paragraphs, knowledges and persecution are of concern. The question of ethics in cyber space in general informs vulnerability research as well. Those debates go from vulnerability research in particular (Matwyshyn et al., 2010), broader realms such as the field of IoT (DeHondt, 2019) to the very debates what it needs to govern cyber space in general (Dickow et al., 2015; Stamatia, 2019).

The question of cyber space and what governs it ultimately leads to the problems associated with software vulnerabilities. Since they pose a threat to systems, they are oft talked about in the realm of "national security" (Ambastha, 2019; White House, 2017). Vulnerabilities can serve as the basis to produce exploits or software which is "weaponized", leading to questions of threats and warfare in cyber space (Caravelli & Jones, 2019; Daras, 2019; Geers & NATO Cooperative Cyber Defence Centre of Excellence, 2011; Prasad & Rohokale, 2020).

2. A Mobilization of Concepts

Where to start off with the investigation of software vulnerabilities? This is the question I will seek to answer in this chapter. First and foremost it has to be said that this case study is based in Science-Technology-Studies (STS), a (sub-)field in Sociology concerned with thinking about the (inter-/intra-)relations between technologies, scientific discovery and society. The concerns in this field are related to the understandings of technology as the consequence of debate, experiments, ideas and other instances of social interaction. This is also reflected in the fact this discipline is sometimes called Science-Technology-Society.

I will mobilize the insights gained from Actor-Network Theory to understand and examine software vulnerabilities. Rooted in a rich tradition of diverse influences, STS in general and ANT specifically provide many valuable resources for thinking about "the" digital, materiality and technology. Before I introduce some key concepts of this thesis, however, I will say in advance that Actor-Network Theory, or what is commonly understood as such, is in and of itself quite a contested term. In the introductory and possibly most comprehensibly written book on ANT, "Actor-Network Theory: Trials, Trails and Translations" by Mike Michael, they state that "[…] it is a complex, and oftentimes disparate, resource (closely aligned with a particular, evolving, set of sensibilities) that opens up a space for asking certain sorts of methodological, empirical, analytic and political questions about the processes of the (more-than-)social world." (Michael, 2016, p. 3) But let me start at where I see the beginnings of this thought.

2.1 On Science, Knowledge and Technology

2.1.1 Knowledge and the Processes of Science

One of the common starting points in understanding this discipline is arguably found in the Sociology of (Scientific) Knowledge (SSK) and its discussion in David Bloor's chapter "The Strong Programme in the Sociology of Knowledge" (Bloor, 1991). Here, Bloor argues that sociologists should concern themselves also with (scientific) knowledge itself as something to investigate, not as being self-explanatory. Knowledge is considered a form of culture, that is, a shared believe between individuals,

building collectives and relying on work to be done to form facts, disciplines as well as the differences between forms of knowledge. Bloor therefore takes up insights from Ludwig Fleck's discussions surrounding the emergence of scientific facts (Fleck et al., 2008) as well as Thomas Kuhn's ideas on "progress" in science. Kuhn understands "progress" as being a "revolutionary" shift in the common understandings, the paradigms, of the practitioners of science (Kuhn, 1962). Bloor's contribution to this discussion are the four tenets of causality, impartiality, symmetry and reflexivity (cf. Bloor, 1991, p. 7). In short, the idea behind these principles is that the sociology of knowledge should take into account not only the "states of knowledge" that prevailed but how those came to be and what other possibilities where discarded ("truth" & "falsity", "success" & "failure"). It should adhere to a "symmetry", the explanation should not only take into account the causality of a successful experiment and "fact" (that is, a knowledge or belief) but also the failures, missteps, the things being left out in and leading to certain explanations. The idea is that with the inclusion (or at least consideration) of everything that "went wrong" in the knowledge production process, the utterances and assertions made by scientific claims may come out stronger than before.

2.1.2 Scientific Controversy, Closure and the Social

A practical application of this view can be found in yet another "classical" text in STS, namely Pinch & Bijker's discussion on "The Social Construction of Facts and Artefacts: or How the Sociology of Science and the Sociology of Technology might Benefit Each Other" (Pinch & Bijker, 1984). In this study, they discuss the emergence of the bicycle over the "Penny-farthing". The Penny-farthing is the bike-like construction with a big front wheel and a very tiny rear wheel, where the driver is seated quite high above the front wheel, resulting in a high balance point. This discussion is part of the "Social Construction of Technology" (SCOT) approach, whereas "[…] the developmental process of a technological artefact is described as an alternation of variation and selection." (ibid., p. 411) In their study, the emergence of the bicycle as the hegemonic design is a result of a controversy between "social groups" with their own interests. Some of them concerned with their safety because of the high balance point, some of them in regards to how fast one can go, together they have different stances on for example the make-up of the streets etc. The "stabilization" of an artefact – in this case the prevalence of the bicycle – only comes about in a lengthy process of negotiation, trial, conflict and closure. Closure and stabilization in these cases mean the settling of disputes through the mobilization of arguments, resulting in one design being more common or accepted as the other(s).

2.1.3 From the Laboratory to Inscription

At a similar time, Latour and Woolgar published their influential book "Laboratory Life": The Construction of Scientific Facts" (Latour & Woolgar, 1986). In this influential work, Latour follows

scientists in a laboratory, observing their daily routines and describing in meticulous detail what processes are needed to end up with something that could be considered a "fact". Here, we can find the introduction of the term "inscription device" which Latour at this point in history describes as being "[…] any item of apparatus or particular configuration of such items which can transform a material substance into a figure or diagram […]" (Latour & Woolgar, 1986, p. 52). That is, the transformation from a material substance into some representation of the same. In a similar fashion to Latour's laboratory study, Karin Knorr-Cetina "followed the actors" in a laboratory setting, which resulted in their influential work "Epistemic Cultures". Here, Knorr-Cetina analyzed the many processes, mechanisms and amalgams that result in a description of "[…] how we know what we know" (cf. Knorr-Cetina, 1999, p. 1). This viewpoint will become relevant in the discussion of "ontology" (see 2.2.3 Assemblages and Ontology). For the moment I will disregard it, however, and concentrate on the inscriptions and transformations that take place.

The facts and knowledge coming out of a laboratory constitute "black boxes" in the making of a "social" world. As Callon & Latour define them, black boxes are everything "[…] that which no longer needs to be reconsidered, those things whose contents have become a matter of indifference." (Callon & Latour, 1981, p. 285) In this understanding, black boxes aren't open for discussion any more. Similar to the "closure" of debates as mentioned earlier, black boxes aren't something which is questioned or being understood as a source of conflict. As Mike Michael puts it "Now, knowledge is not only stabilized through these cascades of inscriptions but also becomes more and more resilient – there are greater and greater costs for those who would wish to problematize the representation at the end of a cascade of simplifications." (Michael, 2016, p. 40)

That is what Latour calls "matters of fact", the ready-made explanations of the world, the "[…] very polemical, very political renderings of matters of concern and only a subset of what could also be called *states of affairs*" (Latour, 2004, p. 232 [emphasis i.o.]). This quote also points toward what Langdon Winner meant when they asked "Do Artifacts Have Politics?" (Winner, 1980).

2.2 Inscription, Agency and Ontology

2.2.1 Inscription and Politics

Following "inscription" as a constitutive part of the "social" world, we can see the emergence of what will be called the "material turn". A first step in this direction may come from Madeleine Akrich in their slightly different reading of "inscription" than Latour:

> "The technical realization of the innovator's beliefs about the relationships between an object and its surrounding actors is thus an attempt to predetermine the settings that

> users are asked to imagine for a particular piece of technology and the pre-scriptions (notices, contracts, advice, etc.) that accompany it."
> (Akrich, 1992, p. 208)

The inscription of devices *not only* renders work being done invisible and is a translation from material things to facts, data or other representations of the physical realm as Latour understood them. They also constitute a *choice* made by the developers and engineers done long before anyone has ever interacted with the technology. It represents an imagination of what the technology should be used for. Being a choice also means, as Winner and Latour already hinted at, the result (the technology) is *political*, since it could be different as well:

> "This is why it makes sense to say that technical objects have political strength. They may change social relations, but they also stabilize, naturalize, depoliticize, and translate these into other media. After the event, the processes involved in building up technical objects are concealed. The causal links they established are naturalized. There was, or so it seems, never any possibility that it could have been otherwise."
> (ibid., p. 222)

A very demonstrative example of this nexus can be found in Johnson's description of a specific technological device in their text "Mixing Humans and Nonhumans Together: The Sociology of a Door-Closer" (Johnson, 1988). Here, they discuss the intents written into the object "door closer". In a quite graphic manner we follow the author through a door, a "hybrid", as they call it, between a wall and a hole. Since there is something which closes the door behind you automatically, there seems to be a translation (also called delegation) of intent from humans to non-humans. An inscription (and, simultaneously, a process of "black boxing") has occurred.

2.2.2 The Problematization of the Social

This quite clearly demonstrates that technologies aren't merely something "out there", cold and lifeless structures or entities which merely serve a purpose when we humans "do" something with them. Bijker and Law introduce this thought by stating that "[…] when we talk of the technological, we are not talking of the 'purely' technological - that no such beast exists. Rather we are saying that the technological is social." (Bijker & Law, 1992, p. 4) A critique on such an assertion, set out by Latour, is that "[…] the social has never explained anything; the social has to be explained instead. It's the very notion of a social explanation that has to be dealt with." (Latour, 2005, p. 97) Turning away from explanations of the "social world" as merely being the interests, ideologies and actions of humans (cf. ibid., 2005, p. 95), the "material turn" therefore introduces technologies and devices themselves as being an active, *constitutive* part of our (social) world. They have *agency*.

What does it mean, then, that non-humans have agency? Generally speaking, there doesn't seem to be the one defining answer to that. One answer we can find in Andrew Pickering's "The Mangle of Practice", where agency is described as "the world […] continually *doing things*, things that bear upon us not as observation statements upon disembodied intellects but as forces upon material beings." (Pickering, 1995, p. 7) That is to say, every non-human entity, be it what we understand as technology or "naturally" occurring (weather, stones, trees…), in the interaction with other entities *do* something. One of the "classic" examples of agency be the "enrolment" of scallops sought to be collected by researchers in Callon's account of "Some Elements of a Sociology of Translation: domestication of the scallops and the fishermen of St Brieuc Bay". The scallops, Callon argues, aren't "just" collected. Callon introduces "enrolment", the "[…] multilateral negotiations, trials of strength and tricks […]" (Callon, 1984, p. 211) that go into this endeavor. On every stage of those negotiations there is the possibility that the collection attempt fails (the destruction of the netted bags through tidal waves, for example). The scallops, in this case, "decide" to not be enrolled, at least not in relation to the will of the researchers. The "translation" of the interests of the scientists was not successful. Therefore, ANT is also sometimes called the "Sociology of Translation".

2.2.3 Assemblages and Ontology

As I hopefully have explained, the technologies surrounding us are the result of (social) practices and represent the intentions of the designers, developers and engineers shaping them. The devices, technologies and entities resulting from these, in turn, have the "[…] capacity to operationalize associated discourses, fields, and practices" (Mutlu, 2012, p. 174) Since entities therefore act on humans and humans act on entities, they both constitute each other, form relations and networks. This understanding, which stands in contrast to the ready-made matters of fact mentioned earlier, points "[…] toward the ways in which 'things' are gathered or assembled together, that is composed out of a multitude of element, practices, 'interests' and so on." (Michael, 2016, p. 118) This multitude is referred to "assemblage", the networks of the world we live in.

As with the choices made that go in the *construction* of technology, the same is true for how we *encounter* the world. The interpretation of materials (non-humans, technologies) is always subject to our specific knowledges, viewpoints and analyses. This is what is understood as "ontology", the very different viewpoints that may be taken into account when investigating the world:

> "It [ontology, M.C.] refers to a potentially empirical investigation into the kinds of entities, the forms of being, or the structures of existence in an area. It is an interest that prompts one to look at the way the empirical universe happens to be configured into entities and properties."
> (Knorr-Cetina, 1999, p. 253)

This understanding casts aside the understanding that there is one viewpoint to be had regarding the material world. The efforts that go into the making of technologies, facts and other entities we assume to be "just there" are split up into a multitude of viewpoints. ANT, "[…] takes the semiotic insight, that of the relationality of entities, the notion that they are produced in relations, and applies this ruthlessly to all materials - and not simply to those that are linguistic." (Law, 1999, p. 4)

There are nowadays quite a lot of studies dealing and introducing this reading of the world. Between accounts such as Annemarie Mol's study "The Body Multiple" (Mol, 2002) and Puig de la Bellacasa's "Matters of Care" (Puig de la Bellacasa, 2017), there is an "[…] ontological understanding of the world as a 'world of becoming', a world in process, unfolding toward the 'not-as-yet'." (Michael, 2016, p. 116). This is what is called "ontological multiplicity" in ANT: The many potential worlds-that-are, the networks being constructed by the onlooker, the practitioner, the reader, the maker, the tinkerer. The world is "more than human" (cf. Bastian, 2017; Puig de la Bellacasa, 2017) , "[…] more than one, but less than many" (Mol, 2002, p. 55).

2.3 Digital Technologies

2.3.1 Inscription and Transformation in Digital Devices

As I've only talked about the "material world" until now, the problem of the digital unfolds. How can we think, when we speak about software vulnerabilities, about a "material" realm?

The main concern of this thesis is located, in contrast, in quite an ephemeral realm. Although there can be found many different approaches to technological *devices*, the *digital* sphere somewhat eludes historical as well as contemporary accounts of sociological investigation. The very programs, the code, the software, the electronics, bits, standards and protocols running *on* the machines are mostly hidden, obscured or just assumed, seldom being the focus of investigation. Some views on this are hinted at in the literature discussed already, yet I think it is worth pointing towards some specific studies and perspectives.

There are some approaches to understand the digital sphere. One of the most obvious and perhaps most discussed ones is the emergence of what we call "social networks" as "mappings" of social interactions. This is quite intriguingly discussed in a paper by Ruppert et al. (2013) where they state that "[…] Social worlds are thus saturated, being done and materialized by digital devices and what is increasingly being understood as 'big data' of various kinds." (Ruppert et al., 2013, p. 23). These approaches are mostly renderings of the relations between users, links and the transactions happening *on* platforms themselves. Also part of investigations are is the potential to extrapolate future events through data, be they insurance claims or terror attacks (cf. Amoore, 2013). In the realm of code and

coding practices there is worth mentioning a fascinating account of "Algorithms as Culture" by Nick Seaver (Seaver, 2017), which introduces the "stabilized" entity of "algorithms" as something contested, multiple and worth investigating how they come to be in their own right. Similarly, Stéphane Couture investigates "source code" as being a matter of concern (Couture, 2019). Jean-François Blanchette offers an insight into "A Material History of Bits" (Blanchette, 2011), arguing that the supposed "immateriality" of information cannot and should not be separated from its material basis, the atoms and bits information ultimately is based upon. In a response, David Ribes sees this as being an "ontological assertion":

> "Instead of reinterpreting the world as one set of fundamental materials, a material methodology gives the tools to recognize the situated and specifically textured nature of reductions and generations, as well as the importance of material agencies when they are encountered. In this sense, rather than casting materiality as an ontological assertion to be enacted across the board, materiality is an additional sensitizing concept along with those that draw our attention to the processuality of, say, practice, documents and archives, collaboration, power, and so on.""
> (Ribes, 2019, p. 54)

So, instead of reducing the digital sphere down to its (physical?) components, everyone investigating this field may be better of not only describing those components but including them in a broader view, how they interact with other "processes". This "materiality" is not defined by one ontological state, but is *enacted*, a *process* and a *practice*.

2.3.2 Performativity

We encountered similar notions before while talking about technology, ontology and inscriptions (see 2.2 Inscription, Agency and Ontology). Many scholars in STS, some of which mentioned, many not, understand the "social" world, everything they do within and what comes to *be* from these doings as enactments or practices. We can see similar notions in the inscription of technologies as seen in Madeleine Akrich's accounts of the developers inscribing ideas about their product or technology as well as in the various understandings of assemblages in ANT. John Law talked about the relation between "performativity" and ANT in the text "After ANT: compexity, naming and topology", where they state that ANT is a strategy to see "[h]ow it is that things get performed (and perform themselves) into relations that are relatively stable and stay in place. How it is that they make distributions between high and low, big and small, or human and non-human."
(Law, 1999, p. 4) Performativity, in this account, relates strongly to the practice in relation to something else. A very similar notion can be found by Annemarie Mol:

> "It is possible to say that in practices objects are *enacted*. […] It also suggests that in the act, and only then and there, something is - being enacted."
> (Mol, 2002, p. 32)

Technologies as something which is being enacted may seem strange if encountered first. But if we think about all the things that are being needed for interacting with technology, how many steps went into making technology and how our bodies are needed for doing all of those things, I think that takes us closer to an understanding of what I'm trying to accomplish in this thesis. Think about the many configurations that are needed for writing. You will need a functioning (all the parts!) computer, some way to enter the things you want to say (e.g. keyboard), some kind of information feedback so you know that you've tipped the correct thing (e.g. monitor), you are bound to the place where this interaction happens and so on and so forth. This, while very much being related to the notion of a "cyborg" (cf. Haraway, 1985), may also be found at the very core of ANT thought:

> "In any event, ANT has roots in a lineage of microsociologies which place emphasis on the analysis of discrete occasions of local interaction as a way of grasping the production of social order."
> (Michael, 2016, p. 24)

There are many "material" technologies that STS has built a vast knowledge base on over the years. This knowledge base was further broadened with the introduction of what came to be known as "matters of care".

2.3.3 From Matters of Concern to Matters of Care

Latour's notion of matters of concern signify a shift away from the ready-made science and facts. The focus on matters of concerns brought to view the many hidden layers of inscription and labour that went into the making of these facts. However, over the years, this notion was developed even further. In particular Annemarie Mol's work on ontology and the many interpretations of the body brought to bear a further development of ANT thought: Matters of Care. In this particular notion of how technologies and (non-)humans come to be, the many processes and resources needed to develop, build and, most of all, *sustain* them. There is work needed to keep them as they are, to prevent them from decay or just about to keep the *status quo*:

> "This version of caring for technology carries well the double significance of care as an everyday labor of maintenance that conveys ethical obligation: we must take care of things in order to remain responsible for their becomings. Recent work that foregrounds the importance of repair and maintenance of technology infrastructures as practices of care supports this case and has expanded it, making a great difference in how objects, devices, and technological infrastructures and the more or less invisible

> agencies involved in their continuation (Star 1999; Star and Bowker 2007a) are conceived."
> (Puig de la Bellacasa, 2017, p. 43)

With this shift in understanding it is possible to not only understand the emergence of networks, actors and the agencies with them, but opens up yet another dimension: The dimension of time and its influence on "matters that matter". The continuous making, re-making and developing of the worlds around us are brought into focus and, first and foremost, made *understandable*. Making them visible helps in understanding them, shifting the attention away from the "grand questions" of the social localizes the matter at hand:

> "This work changes the focus on the 'robustness' of sociotechnical assemblages, on solid and successful networks or black boxes, by drawing attention to the constant need for repair and maintenance (Jackson 2014; Jackson and Kang 2014), the stakes of their "vulnerable" status (Denis and Pontille 2014)."
> (Puig de la Bellacasa, 2017, p. 43 f.)

The notions of repair and maintenance couldn't be more fitting for the topic at hand. Technologies and their continuous demand for attention in the form of a diverse set of practices lies at the core of this thesis. Therefore, the notion of "matters of care" should be kept in mind.
The need for care, as understood by Puig de la Bellacase, stems from interdependencies we are part of. Our bodies, our lives, our resources and our technologies are rooted within the world, therefore we can't neglect the dependencies that come with them. The conclusion is that "[i]nterdependency is not a contract, nor a moral ideal - it is a *condition.* Care is therefore concomitant to the continuation of life for many living beings in more than human entanglements - not forced upon them by a moral order, and not necessarily a rewarding obligation." (Puig de la Bellacasa, 2017, p. 70 [emphasis i.o.]).
Interdependencies we encountered in the very first chapter of this thesis. The connected devices of the internet as well as all the things we built upon this structure speaks to this notion. So is the relation between devices and their lifespan for example one of sustainability:

> "Many of these smart systems and devices (refrigerators, medical devices and cars) are expected to be operational for many years or even decades with a minimum of intervention. They also make extensive use of third-party libraries in integrated products, which act as a black box whose security is difficult to analyse. Therefore, industry, government and researchers should start thinking of how to effectively merge safety with security to ensure sustainability in software and in the supporting tool-chains."
> (Schaake et al., 2018, p. 2)

Others noted the need for care as is introduced here in relation to even more abstract sites, namely the relation between attacker and defender in the realm of cyber security:

> "We wanted to position cyber security as a contest between the defender and attacker. This relationship, in which the notion of complete security is impossible, is characterized by move and countermove. Rather cyber security continues to be an enduring effort, a dynamic that needs constant attention. Central to it is the ongoing motivation and adaptive processes of actors trying to maintain control of their assets while others are trying to deny, degrade, disrupt, destroy, and steal."
> (Jones, 2019, p. 175)

Generally speaking, the notion of care will follow along in all my thinking about technologies in this thesis. This introduction to the history of STS, ANT and quite a lot theoretical concepts discussed in this realm will inform my research questions. This theoretical basis is quite broad, therefore I will follow up with a breakdown of the most important concepts.

3. Research Questions

The main research question in this thesis is located in one minuscule location of the interaction between the "social" and "technology": the Coordinated Vulnerability Disclosure process. This process is meant to mitigate harmful effects of computer programs. Following the ideas put forward by Actor-Network Theory as an approach within the field of Science-Technology-Studies, the formation of the "social" world rests upon the smallest interactions of actors, whereby actors can be human and non-human alike. Only through the processes of transformation, enrolment of interests and stabilization bigger structures may emerge. The proposed method to investigate such processes is to "follow the actors" (Callon, 1984, p. 201). In this case, the actors to follow will be the software vulnerabilities as they are discussed in the context of CVD. Since the discussions in the CVD are in most cases not public, I had to approach and interview practitioners of vulnerability research as well as the people receiving the reports about found vulnerabilities. Their insights, framing and experiences, combined with relevant literature, will serve as the basis for the research.
The hypothesis is that only through the interaction between a variety of actors a software vulnerability comes into existence.

MQ: How do software vulnerabilities inform the formation of social structures?

My motivation in asking this question is to gain a deeper knowledge of how the diverse material devices and objects inform the formation of social structures. This main research question may answer how the complex things "software vulnerabilities" are formed, are subject to change and negotiation while traversing through through different stages. These stages will be opened up with the three sub-questions. Ultimately, the question should shift the focus toward the capabilities of the entity "software vulnerabilities" as an actor in the world and how it shapes its surroundings.

SQ1: What elements constitute software vulnerability research?

The SQ1 seeks to find answer on what knowledge(s) and/or practices are necessary to understand the potential of vulnerabilities, what are markers to recognize them and how they come to be as something that can be acted upon. This question also encompasses what resources are mobilized in engaging in information exchange, the expectations of doing so and what considerations are employed.

SQ2: How are software vulnerabilities assembled in the CVD process?

The SQ2 follows the SQ1 in the sense that withing the CVD there have to be followed certain steps. The processes of reporting as well as triage are considered constitutional elements of the CVD. Therefore the question is what part do software vulnerabilities play in this process? The SQ2 investigates what considerations go into these processes, what are the constituting parts and when is a vulnerability acted upon?

SQ3: How does the CVD process inform the emergence of social structures?

SQ3 deals with the question of how the implementation of a CVD process as well as “successful” reports result in the emergence of social structures. What resources go into this process, what knowledges are gained, and, closing the loop to the MQ, what kind of social structures emerge from this practice?

4. The Case Assembly: Methodology & Materials

4.1 An Introduction to the Research Field

As already stated in some instances throughout this thesis, this research is firmly located in Science and Technology Studies. This academic field prides itself with a strong focus on case studies (and rightfully so, I may add). Moving away from the "grand explanations" is somewhat inscribed into the theoretical and practical approaches. There also seems to be a long tradition in qualitative research methodology. I am guilty of both those things in the construction of the thesis at hand.

The topic of software vulnerabilities was strongly informed by personal interest. As potentially obvious from the Context section (see 1.1 Three Dimensions of Vulnerability) I recognize this topic as being at a meeting point of Computer Science, (Critical) Security Studies and STS. The inclusion of other realms of (scientific, academic) investigations is necessary. True to the ideas of "following the actor" in ANT, the research questions themselves are also aimed at investigating common understandings of technology from these perspectives. The understandings of this thesis, therefore, are definitely informed by the insights gained from STS and ANT, yet it would be intellectually dishonest to say that these would be the only influences gained to examine software vulnerabilities:

> "The hypothesis that knowledge originates in non-knowledge as it were, in nothing (*ex nihilo*), completely overlooks the societal genealogy of knowledge, such as the close, even intimate relationship between scientific and practical knowledge. The birth of a scientific discipline is no parthenogenesis. The hypothesis of the transformation of non-knowledge into knowledge favors certain knowledge in that the origin of new knowledge is simply suppressed."
> (Stehr, 2017, p. 121)

The field of research is, consequently, actually quite broad. I encountered ANT and the thoughts of scholars from the field of STS in different disciplines, approaches and discussions as well. I feel like those perspectives are definitely worth discussing, since they inform, ultimately, what software vulnerabilities are and how they act in the world. We can't talk about ontological multiplicity without

accepting the encounter of supposedly the same things in a completely different context, disregarding the part of "multiplicity". The same observation was also made by Latour himself:

> "Of course, this study is never complete. We start in the middle of things, *in medias res*, pressed by our colleagues, pushed by fellowships, starved for money, strangled by deadlines. And most of the things we have been studying, we have ignored or misunderstood. Action had already started; it will continue when we will no longer be around. What we are doing in the field - conducting interviews, passing out questionnaires, taking notes and pictures, shooting films, leafing through the documentation, clumsily loafing around - is unclear to the people with whom we have shared no more than a fleeting moment."
> (Latour, 2005, p. 123)

To understand software vulnerabilities, CVD and the emergence of social structures with a limited approach in the methodology applied would therefore constitute a perfect example of *boundary work (cf. Gieryn, 1983)*. This seems to be the case for many case studies related to digital technologies. This realm seems to avoid the grasp of any one specific realm of academic discipline, not only in the methods applied to investigate the realm *per se* but also in the terminology and concepts applied, who is considered and expert and who has the authority to talk about it in general:

> "I take terminological anxiety to be one of critical algorithm studies' defining features. But this is not because, as disciplinary outsiders, we are technically inept. Rather, it is because terminological anxieties are first and foremost anxieties about the boundaries of disciplinary jurisdiction, and critical algorithm studies is, essentially, founded in a disciplinary transgression. The boundaries of expert communities are maintained by governing the circulation and proper usage of professional argot, demarcating those who have the right to speak from those who do not […], and algorithms are no different."
> (Seaver, 2017, p. 2)

Even though Nick Seaver talks about critical algorithm studies in this quote, I feel absolutely the same about software vulnerabilities. With the emergence of ever-growing numbers of devices and practitioners in the field of ICT, however, I am simultaneously convinced that STS in general and ANT in particular have something to say about software vulnerabilities. Following the theoretical approaches and the literature discussed I was certain that this investigation is fruitful.

My first idea was to investigate bug bounty programs. I discarded that idea as I understood them to be as something more or less "stabilized" (this assertion, in hindsight, could quite easily be challenged). I had the feeling I wanted to do something more promising, adventurous, more open for negotiation. At last I found the CVD, which, as explained, seemed to fit this description quite nicely.

Necessary to say at this point, however, is the limited format a Master's thesis provides. The research into CVD quickly turned out to be even too extensive for this project. Therefore, the focus I chose was on the three research questions outlined. There is a lot to be said and investigated, still.

4.2 Methodology

Probably the first and most pressing problem I encountered during this thesis is the methodology of ANT. Not only that software vulnerabilities themselves turn out to be quite an elusive entity. The *methods* to encounter them are spread out across a lot of different realms, approaches and understandings. As will be seen in the Analysis section, the term "vulnerabilities" is somewhat contested. Also, questions of "how to investigate digital realms" itself is discussed:

> "In a nutshell, this is one meaning of 'the social life of methods', which is elaborated in the introductory essay to this special issue. But if we are to understand this in the context of the digital, then we need to attend to the lives and specificities of devices and data themselves: where and how they happen, who and what they are attached to and the relations they forge, how they get assembled, where they travel, their multiple arrangements and mobilizations, and, of course, their instabilities, durabilities and how they sometimes get disaggregated too."
> (Ruppert et al., 2013, p. 31)

It is not easily done, the search for the establishment of devices and data. I would include programs, code and programming language into this mix, as I understand them to be different things. The result is quite a methodological challenge where "[…] 'following' as a methodological principle advances from a seamless movement to a situated methodological configuration which may involve cuts, jumps and fissures" (Gerlitz & Weltevrede, 2020, p. 355). All that being said, how did I go about, then, in investigating the proposed research questions?

4.2.1 Document Analysis

With qualitative (as probably with all) research comes the "desktop research". The encounter of texts upon texts, the skimming of vast amounts, reading of lots and analysis of many core documents related to ones research. Not only being embedded in a university program necessarily leads to the consultation of many different written accounts related to your research field, also the formulation of a research question asks for specialized input. Consequently, the precise starting point of this research is probably irrecoverably lost.

Generally, a thorough literature review is necessary to identify research gaps and to get to know the work already done by other people in the field. This is seen as a social process whereas "[…] [i]deas,

research methods and knowledge all develop over time, with researchers critiquing and building on each other's work." (Jensen & Laurie, 2016, p. 30).

As can be seen in the contextualization and framing of the research, this resulted in the consultation of many documents across many domains. From policy documents, standards and online resources to monographs and academic papers a vast variety of different documents went into this thesis. The documents I consulted I deem not only as passive resources of knowledge, but as something that influences also my understandings. They were written to achieve something in the world, therefore their consultation is also something which is done actively. Reading, understanding, contextualization and handling are practices which need to be mentioned at this point. Therefore, it is worth including them here in the methodology section.

I clustered the documents thematically, so as to make their differences clearly visible. This is specifically interesting when it comes to definitions or practices. The groupings mostly were along the lines of what kind of document was consulted, from standards to "good practice" guidelines, academic papers, theoretical approaches or security related documents. Since topical overlaps are a given, in particular in the understandings of my theoretical approach, however, cross references have to be made. The contextualization shall help in understanding the relevancy to the statements made and thought processes applied.

4.2.2 Qualitative Interviews

My method of choice in doing this research are qualitative interviews. This methodological approach was chosen because it helped me to gain access to the knowledge and experiences of practitioners in the field. Their accounts, narratives and framings were of interest to me. Specifically because I considered software vulnerability research as something quite "hidden" or done in private. In my desktop research I didn't find many accounts of how to do vulnerability research and what context this practice is done in. Therefore, I had to speak with people doing it.

Qualitative interviews are a time-honored tradition in doing research in the social sciences. In contrast to quantitative approaches, they are mostly based on a smaller number of participants. There are, however, quite a lot of different approaches towards doing interviews. They range from structured to semi-structured to open interviews (Flick, 2009). Structured interviews are characterized by having a fixed canon of questions without the possibility to change the order or appearance of questions. Semi-structured interviews have a prepared set of questions to be asked, however the sequence or formulations on how they are asked may be subject to change while doing the interviews. Open interviews often take the form of conversations, only being guided by a general interest of the researcher, maybe based upon notes and previous experience, resulting in a situation which may be

adapted depending on the situation. This represents a very open and *ad hoc* format to approaching research.

In my thesis I chose to follow a semi-structured interview approach. I saw this as the most promising way since here I had the possibility to really think about my focus beforehand, prepare clearly formulated questions and having a "red thread", a guiding structure at hand. This also helps if there is a lull in conversation or having a blackout. This format helps in kick-starting the conversation, leap jump to other sections or giving me the freedom to dig deeper if some interesting or surprising aspect comes to light. For my interview partner, this opens up the possibility to also talk about their interests and experiences:

> "By minimizing restrictions on the base of scope of the conversation, **semi-structured interviews** allows your participants to answer freely based on personal reflection, knowledge and experience."
> (Jensen & Laurie, 2016, p. 173 [emphasis i.o.])

Therefore, interviews are a site where lived experiences may be categorized, ordered and taken into context. In particular this ordering aspect is relevant for my research questions as it allows me to understand the social interactions and provides the basis for my analysis.

The analysis was done using the approach of "grounded theory" as proposed by Corbin & Strauss (Corbin & Strauss, 1990). This approach is characterized by basing a theoretical approach within the data itself:

> "At the same time, a grounded theory specifies the conditions under which a phenomenon has been discovered in this particular data. A range of the situations to which it applies or has reference is thereby specified. In utilizing theory, practitioners or others may encounter somewhat different or not-quite-the-same situations, but still wish to guide their actions by it. They must discover the extent to which the theory does apply and where it has to be qualified for the new situations."
> (Corbin & Strauss, 1990, p. 15)

After doing the interviews, I transcribed them by hand using the software "easytranscript"[7]. This was done in a verbatim style, with the omission of laughter, throat clearing, background noise, pauses. This helped not only in the readability of the transcribed interviews (and therefore in coding), but I reasoned it was not necessary for this kind of research. Even though the transcription was done in a more detailed way, the quotes used in the thesis were changed to a non-verbatim writing style for better, smoother reading. The information included in the statements were, in my understanding, not changed.

7 https://e-werkzeug.eu/index.php/en/products/easytranscript

"Grounded theory" is characterized via a coding strategy, building a themes and concepts using the steps of *open coding*, *axial coding* and *selective coding*. These coding strategies were applied to the transcribed interviews using the software "Atlas.ti" [8]. The initial strategy of open coding resulted in approximately 65 codes. Switching from these 65 codes, I formed categories which resulted in approximately 25 categories. Ultimately, the categories where grouped together into yet another topical cluster, containing some 20 categories, including some smaller subcategoris. This last grouping formed the basis for the analytical chapter. In general, I started from approximately 350 relevant parts of all 6 interviews.

4.2.3 Finding Interview Partners

In my case, I focused on two groups of people. First, researchers of software vulnerabilities. These are the people that "hunt" for software vulnerabilities in bug bounties or have participated in CVDs in the past. Their experiences offer insights into how they go about in doing this research, what knowledges they may have and what their experiences in the communication with the affected parties or companies were. My approach to contact researchers was to find them online. Since there are public Hall of Fames where researchers who found relevant vulnerabilities are honored I had a list of potential candidates. However, not all of the represented researchers in those Hall of Fames gave information details. Some of them gave personal information such as their clear names, e-mail addresses or personal websites. I contacted approximately ten different researchers. Unfortunately, many didn't answer or the communication died down after an initial response. In the end, however, I managed to establish deeper connections with two of them, whereas after the interview one of my interview partners mediated a connection to a second interview partner. I am very thankful for this introduction and connection.

The second group of people I wanted to interview are the recipients of vulnerability reports. This group of people, while having a public-facing e-mail address in many cases as is the very topic of this thesis, I considered to be more difficult to approach. This has the reason that I understood their experiences to be work related. This results in two arguments:

The first being that the examination and dealing with reports is done in their work time, therefore the timing of interviews may be more difficult because it is work related and I may be not considered a worthwhile activity to deal with in their leisure time. On the other side, scheduling an interview for a Master's thesis during working hours may be seen as not work related in the companies' eyes.

The second consideration had to do with this topic (potentially) being subject to security considerations. Since I was not known to my interview partners, my intentions may be deceptive or otherwise interpreted as malicious. Especially in the context of companies and me approaching

8 https://atlasti.com/

specifically some of them may be construed as some form of "(spear-)fishing", a known tactic in cyber security to gain access to otherwise restricted information. These considerations seem to be exaggerated, yet were definitely confirmed. This resulted, for example, in the non-disclosure of specific numbers of vulnerability reports in at least one case. For the context of this thesis, these kinds of information luckily aren't fundamental or enabling, yet for research in security or privacy related contexts these aspects have to be kept in mind. Consequently I want to, again, say thank you to my interview partners for their front-loaded trust.

To find contacts for this aspect to my research I took two approaches. The first one being personal contacts to circumvent the problem of "anonymity" or at least to have some kind of accountability in place. This helped a lot in forming trusting relations. Thanks to all of the intermediaries as well at this point. The second one was writing e-mails with as much detail about me and my research as possible. Fortunately, also this strategy was successful in finding at least one interview partner.

To find suitable interview partners, I searched the realms of what I understand as "critical infrastructure". I wanted to have insights across as many domains as possible (as it if feasible to do so in a Master's thesis). I am very thankful and happy that I found quite a diverse set of companies which were open to talk to me. The business sectors I tried to find interview partners in are the technology sector, the financial sector, the telecommunications sector and the (bio-)chemical sector. Not all of those sectors are present in this research, yet I think that the information density necessary for a comprehensive analysis was reached.

4.2.4 Ethical Considerations

With this research came some caveats. The first of which is possibly exactly what makes it so enticing: Being located in a realm of the intersection between STS, ANT, security studies, potentially "dangerous" to investigate or at least something I had to be careful on how to approach.

Not only the security aspects are relevant in considering ethical issues in research, however. Often overlooked or only mentioned in passing I really want to draw the focus on the necessity of research ethics. Being a strong proponent of privacy rights, I encountered some issues in doing this research. First of all I have to mention, at last, in which period this thesis is written. Not only what is considered "late-stage capitalism" and global warming. These are other problems. What I mean with the period in which it is written is the dreaded topic of Covid-19. Specifically that this thesis is being written and the research having been done during the infamous SARS-CoV-2, also known as Covid-19, pandemic.

With that came some limitations in how I could do the aforementioned interviews. Not only that traveling were and are subject to limitations, a lot of the social interactions in general were translated and transformed to take place online. These interviews, therefore, were conducted in the very realm which is discussed here; namely using software, programs and the dependencies that come with them.

Topics such as privacy, data storage, problems with (online) services and other issues suddenly became not only a matter of concern for me because of personal interest and academic endeavor but a *necessity in life itself*. Something that cannot be easily negotiated or circumvented. Therefore, the question *how* to go about during one's research is of utmost importance.

Having thought a lot about the interactions of security, privacy and technology I found that conducting interviews online is a somewhat peculiar practice. Especially if the topic is cyber security. Since I wanted to move forward in my Master's program, however, I had to compromise. The Informed Consent Sheet as a regular research practice helped. This form is an understanding or agreement between the researcher and the research participants. In this case I formulated an Informed Consent Sheet outlining the parameters of my study. I included a description of the study interest, what the interview process will most likely look like and where they will be conducted. Being limited to online interactions at the time, I didn't want to limit my options or those of my interview partners. Therefore, I thought it wise to let the specific service/platform used open to negotiation. I was aware that some companies or people have different preferences in regards to which platform to trust online. Also, there are sometimes compliance requirements in place, limiting personnel in their choices which platform to use.

All of those considerations where put into a written format. Also, considerations regarding the use, storage and handling of data were included. I wanted to keep the interviews as guarded as possible. Having them done online is somewhat of a contradiction in and of itself. However, I made the decision to never store them online (in a cloud solution, for example), not using transcription software which is using machine learning (thereby connecting to servers outside my influence) and not sending unnecessary content in other forms. Since online services and connection are always subject to possible security incidents, hacks, data loss and other adverse events I had to put a provision dealing with such events in the Informed Consent Sheet.

In regards to the usage of quotes or information obtained in the interviews, I used anonymized (or pseudonymised) interview quotations. The difference being the ones having traceable information (names, locations, etc…) redacted through the use of numbers, letters or other codes. This practice should be done so that data can't be tracked back to a specific person or at least not in doing so without mobilizing an unreasonable amount of effort to do so. Anonymisation is (or, at least *should be*) the impossibility to connect data back to a specific person.

This is in regards to the security-sensitive nature of the topic but also to avoid attribution to specific interview partners, giving them the opportunity to speak more freely and possibly in contradiction to public facing communications.

Provisions on the storage of the transcriptions and other data material was included as well. This is in regards to the physical and digital access as well as which people in general may access it. In my case

only people directly related to the research project (e.g. my supervisor) are allowed to see the raw data. The encryption key (the lists relating quotations to specific interview partners) was (is) stored in a separate location that the data itself. Achieving complete security, however, is impossible and is definitely part of the spirit of the agreement.

All my interview partners as well as myself signed the Informed Consent Sheets. Time and possibility to change, questions, doubts or other comments was given adequately.

Having faced all these limitations and constraints, I was quite happy that, generally speaking, the interviews went well (apart from minor technical difficulties, as are to be expected in doing a project like this). That was not least thanks to the support and assistance of my interview partners, who faced the same issues as I did, at least in setting up communications.

5. Analysis

The very problem of an analysis such as expected of me in this thesis (and university program!) brings me back to the very theories encountered in the same. As Bruno Latour observed:

> "Every single interview, narrative, and commentary, no matter how trivial it may appear, will provide the analyst with a bewildering array of entities to account for the hows and whys of any course of action. Social scientists will fall asleep long before actors stop deluging them with data. The mistake we must learn to avoid is listening distractedly to these convoluted productions and to ignore the queerest, baroque, and most idiosyncratic terms offered by the actors, following only those that have currency in the rear-world of the social. Alas, this mistake is made so often that it passes for good scientific method, producing most of the artifacts of social explanations."
> (Latour, 2005, p. 47)

How to follow that assertion, then? Even though I will try to avoid the mistake Latour laid bare here, I have to start with my analysis at some point. My starting point, alas, will be the very interviews I conducted. Having in mind Latour's warning, however, I will seek not to find the explanations my interview partners so readily provided, but I very much will try to not "[…] confuse the cause and the effect, the *explanandum* with the *explanans*" (Latour, 2005, p. 63 [emphasis i.o.]). The question in practicing ANT is always "how to follow an actor?". Together with the question of when to stop creating a network this seems to be the practically most difficult thing to do in researching something. In this analysis section I will try to follow the pattern on how a CVD is laid out by the "good practice guides" mentioned in the "Context" section (see 1.4 A Complex Ecosystem). This is the standard unidirectional model, going from a researcher to the vendor or company.

5.1 Intermezzo – A Need for Clarification

5.1.1 Databases, Weaknesses and Resources

Probably one of the most explicit sites where software vulnerabilities can be observed to emerge as social structures is the establishment of databases collecting them. These repositories and resources,

where publicly known vulnerabilities are listed, may inform companies or practitioners and help them to understand potential adverse effects they may suffer and how to possibly deal with them. Three of them I want to introduce in this chapter, namely the CVE, the CWE and the NVD. Of relevancy is not only that these are based upon what are commonly considered "publicly known" vulnerabilities, but also how they relate to each other and the people who encountered them first.

The CVE is a system to organize publicly known vulnerabilities and to assign a unique identifier. This database, as well as the CWE, are handled and maintained by the MITRE organisation, a not-for-profit company from the United States[9].

The CWE, quite similar, is a system to organize the most common software weaknesses, whereas "'[w]eaknesses' are flaws, faults, bugs, or other errors in software or hardware implementation, code, design, or architecture that if left unaddressed could result in systems, networks, or hardware being vulnerable to attack" (MITRE, n.d.-c). Both of these resources are community maintained, meaning whoever follows the appropriate steps may disclose information and help to make these systems grow.

How do weaknesses then relate to vulnerabilities? Weaknesses are the underlying "problems" that can be addressed by certain mitigation strategies. Vulnerabilities refer to certain products or instances of products which are vulnerable. The National Vulnerability Database (NVD), which is yet another resource where one can find information about vulnerabilities, is maintained by the National Institute of Standards and Technology (NIST). This institution "[…] associates a given CVE vulnerability to the underlying CWE weakness" (Booth et al., 2013, p. 2).

To give a tangible example of this relationship, the vulnerability with the identifier CVE-2021-35983 is related to Adobe Reader DC. The "Current Description" of this specific vulnerability reads as follows:

> "Acrobat Reader DC versions 2021.005.20054 (and earlier), 2020.004.30005 (and earlier) and 2017.011.30197 (and earlier) are affected by an Use-after-free vulnerability. An unauthenticated attacker could leverage this vulnerability to achieve arbitrary code execution in the context of the current user. Exploitation of this issue requires user interaction in that a victim must open a malicious file."
> (NIST NVD, n.d.-a)

The website also references a "Weakness Enumeration". In this specific case it seems that the vulnerability CVE-2021-35983 is based on a weakness with the identifier CWE-416, which reads as being named "Use After Free". The description for this weakness, in turn, reads as "Referencing memory after it has been freed can cause a program to crash, use unexpected values, or execute code" (MITRE, n.d.-b).

9 www.mitre.org

We can assert that this specific vulnerability relates to a specific program based on a specific type of weakness. We know that the vulnerability is triggered when someone clicks on an infected file. This causes the memory allocation of the RAM module to be faulty. This results in the possibility for the attacker to run a program which has the same privileges (read, write, access etc.) as the user which clicked on the infected file.
Since "Use After Free" seems not to be a flaw of this *specific* program or an *instance* of this program it is a general weakness in programming. The reference to a program and its affected version numbers indicate that the vulnerability can be somehow mitigated:

> "Mitigation of the vulnerabilities […] typically involves coding changes, but could also include specification changes or even specification deprecations (e.g., removal of affected protocols or functionality in their entirety)."
> (NIST NVD, n.d.-b)

This is what is commonly called (rolling out) a "patch". In the NVD database there is also a link present to the vendor's website, the company Adobe. When I follow this link I end up the "Adobe Security Bulletin", which seems to be resource where security patches are made public. I can find yet again the description and identifiers for the vulnerabilities rolled out with the last patch, including the one discussed above. Also, I find a section "Acknowledgments", whereas is stated:

> "Adobe would like to thank the following for reporting the relevant issues and for working with Adobe to help protect our customers: Nipun Gupta, Ashfaq Ansari and Krishnakant Patil - CloudFuzz working with Trend Micro Zero Day Initiative (CVE-2021-35983)."
> (Adobe, n.d.)

Ultimately, this resource lead us back to the very people who found the vulnerability initially. These resources inform a lot of other actors and are one of the effects of vulnerability disclosure.

5.1.2 Software Security, Cybercrime and Cyberwar

If people rely on digital devices in their everyday life, it must follow that they have to trust the very people producing the software, since their intentions lay at the very core of the devices:

> "Since IoT systems will all be interlinked and sharing information, the user must - by default - *trust* everything in the chain where the data will be shared. It is not enough to be able to trust only the device that is being interacted with, all subsystems must be trusted. As few as one device or 'thing' that is not trustworthy will corrupt the entire system. This single point of failure will provide a ripple effect throughout the entire system."
> (DeHondt, 2019, p. 140 [emphasis i.o.])

With trust comes interdependencies. Since, as mentioned in the very beginning, digital devices, computer and code is ubiquitous, we have to attend to these technologies. The matters of concern have to be turned into matters of care, as stated in 2.1.3 From the Laboratory to Inscription. However, with that interdependency come additional challenges, as shown by for example the European Commission:

> "Our world relies on digital infrastructures, technologies and online systems, which allow us to create business, consume products and enjoy services. All rely on communicating and interaction. Online dependency has opened the door to a wave of **cybercrime**."
> (EC, 2020, p. 3 [emphasis i.o.])

We can, again, find the transformation of the very intention of the software itself. The "normal" functioning of the software comes hand in hand with a malicious second nature. Together, I can assert that cyber crime seems to be one of the things being tightly coupled with descriptions of vulnerabilities *because* of our reliance on the internet as a whole.
As the quote above points at, the interconnections can be broken, altered or otherwise harmed. Since access and interconnection are both a given in many areas of today's world, "ICT systems are hacked every day for robbing money and business secrets, for political aspirations or for stealing intellectual property" (Prasad & Rohokale, 2020, p. 2).

Cyber crime, that has to be said, is not a very specific term. I have to make a distinction between software vulnerabilities and other malicious activity which can be construed as "vulnerabilities" to software. My understanding relates to the technical environment, coding, programming languages, the interconnections between digital devices and so on.

- Social Engineering: This tactic "[…] refers to manipulating people to reveal sensitive information" (Klimburg-Witjes & Wentland, 2021, p. 2). Under this umbrella term fall for example the strategy of sending e-mails with a link to a false login page, urging users to enter their (correct) information and subsequently getting to know their login credentials. This area will not be part of this thesis since the information is offered "voluntarily" and therefore can't be regarded as a *software* vulnerability.

- Exploits: As Householder et al. define, "[a]n *exploit* is software that uses a vulnerability to achieve some effect. Sometimes the effect is as simple as demonstrating the existence of the

vulnerability" (Householder et al., 2017, p. 2 [emphasis i.o.]). Therefore, the exploit is already a piece of software which *uses* the vulnerability, not the vulnerability *itself*.

- Malware, as another quite similar term, can be described as "[m]alicious software designed to introduce malign actions in the intended system" (Prasad & Rohokale, 2020, p. 28). This definition can be extended with the qualifier "But not all malware involves exploits" (Householder et al., 2017, p. 2). Malware is definitely relevant in the realm of cyber security, however it goes above and beyond discussions regarding software vulnerabilities specifically.

Taken together, this chapter can be summarized by stating that vulnerabilities are ubiquitous, refer somehow to intention, can be seen as a weakness and may be expoited. Yet, if I take a closer look at all the termini we introduced in this chapter, I may end up asking the following: Cyber crime? Threat source? Security policy? Protocols? Exploited? Triggered? Controls? All of that very much sounds like a terminology of conflict, war and even terrorism. True. This connection, however, is deeply rooted in the very real fears as is for example put forward by, again, the European Commission:

> "The ever-increasing ways in which digital technologies benefit our lives has also made the **cybersecurity** of technologies an issue of strategic importance. Homes, banks, financial services and enterprises (notably small and medium enterprises) are heavily affected by cyber-attacks. The potential damage is multiplied still further by the interdependence of physical and digital systems: any physical impact is bound to affect digital systems, while cyber-attacks on information systems and digital infrastructures can bring essential services to a halt."
> (EC, 2020, p. 3 [emphasis i.o.])

Since there is a reference to "essential services" and "infrastructure", I will follow up with this quote:

> "In cyber conflict, the terrestrial distance between adversaries can be irrelevant because everyone is a next-door neighbor in cyberspace. Hardware, software, and bandwidth form the landscape, not mountains, valleys, or waterways. […] Basically, tactical victories amount to a successful reshuffling of the bits – the ones and zeros – inside a computer."
> (Geers & NATO Cooperative Cyber Defence Centre of Excellence, 2011, p. 10)

Ultimately, the discussions surrounding (software) vulnerabilities also reach deep into discussions of (cyber) warfare, terrorism and the like. That all sounds very alarmist, dangerous and exaggerated. I agree, to some extent. Yet I very much want to point to the fact that software and its applications aren't something innocent *per se* or can't be utilized in a way which very much is talked about in the contexts of loss of life and security. Software vulnerabilities as the carriers of capabilities lie at the

core of these debates, as can be seen, for example, in the "Vulnerability Equities Process" set up by the government of the United States of America (USA):

> "The Vulnerabilities Equities Process (VEP) balances whether to disseminate vulnerability information to the vendor/supplier in the expectation that it will be patched, or to temporarily restrict the knowledge of the vulnerability to the USG, and potentially other partners, *so that it can be used for national security and law enforcement purposes, such as intelligence collection, military operations, and/or counterintelligence*."
> (White House, 2017, p. 1 [emphasis M.C.])

This perspective stands in stark contrast to the goal of, for example, the US-CERT in reducing cyber security risks. The same observation was made by Mimansa Ambastha in 2019 when they wrote:

> "Herein lies the dilemma: our state agencies are tasked with protecting the nation, a task that involves *both* securing the nation's systems *and* gathering valuable intelligence against actual and potential adversaries. The former would require the agency to disclose any vulnerability to the vendor so that it may be patched, whereas the latter would require restricting disclosure and exploiting the vulnerability to target potential adversaries at the cost of general cybersecurity."
> (Ambastha, 2019 [emphasis i.o.])

Obviously, we therefore have to distinguish between nation-states as being an entity of their own in the discussion of software vulnerabilities and their respective parts. There may be organizational units and social structures which follow their own imperatives, logics and patterns. Which only further informs the ideas of ANT that the most minuscule interactions result in the formation and understandings of bigger structures. In this case the different enactment of software vulnerabilities results in the formation of, one the one hand, a security apparatus and on the other hand the formation of an institution tasks with caring for and remediation of the same vulnerabilities.

As this chapter has shown and to reiterate, the discussions surrounding software vulnerabilities are deeply connected to debates of cybercrime, terrorism and war. But, rest assured, this chapter shall be the most hyperbolic I will be. I primarily want to make sure that my readers keep in mind that these discussions are also rooted in quite catastrophic and conflict-laden terminology. As seen here, they warrant our attention because they are discussed in quite alarming terms and places. Which, ultimately, brings us back to the very strategies on *how* to handle these vulnerabilities:

> "Each of those [the previous mentioned estimated fourteen vulnerabilities of the average programme, M.C.] weaknesses could permit an attacker to compromise the integrity of the product and exploit it for personal gain. Therefore, software vulnerabilities and their timely patching pose a serious concern for everyone. What can we do to protect ourselves?"

(Schaake et al., 2018, p. 1)

The CVD seems to be one of those sites where we can protect ourselves. But how does this process look like? In the next section I will “follow the actor” “software vulnerability” in the context of CVD..

5.2 Before Bugs – Starting Locations

5.2.1 Expectations

In researching vulnerability disclosure, the first thing I came across was a certain notion of expectation. Quite possibly related to the aforementioned ubiquity of such bugs, there always seemed to be an idea of “just doing it” and you will eventually encounter some bugs. This resulting in sentences such as:

> “So... so, it’s just mostly like I take something that I find interesting and then I start learning about it and then the bugs come naturally, I’d say.”

> “Oh, yeah, definitely, like, nobody is perfect coding and everyone makes mistakes. So, it’s just... it’s just finding them.”

I think that’s quite curios, the expectation that there are just are bugs in whatever you take a look at and then they will reveal themselves. A viewpoint I want to draw attention to is the active stance towards bugs, they being the ones coming to the researcher, not the other way around. As we I have discussed in the first section of this thesis, however, I think that the expectation by researchers or practitioners is a fair assumption. This can also be seen in the next part regarding a feeling when searching for them.

5.2.2 Feelings

Interestingly, however, most of the answers I got introduced a very diffuse notion of finding them. Some answers I got mention that there is a moment where vulnerabilities can be felt *before* they actually encounter them, *before* they know it exists or what it does:

> “So, I always jokingly say, in response to these kind of questions, that I have a gut feeling now. And, luckily, over the years now, I’ve gained experience to where I sort of know and recognize when something is impactful.”

> “It’s like, yeah, experience accounts for it, most of the time. But, like, more generally, how a site would look vulnerable, feel vulnerable.”

> "Yeah, well, usually, you encounter a vulnerability when you, like, can exploit something. But, like, even before that, you can somehow sense the vulnerability coming. As, like, perhaps like, you see a feature and you're like 'Surely they did not account for that.'"

The observation I want to make that there is a recurring theme of "feeling" to vulnerabilities. This notion carries a very subjective notion to it, one which seems to stand in stark contrast to the possibly expected mundane handicraft of searching for them in a technical environment. With the introduction of the categories "shouldn't do that", "potential" and "experience", however, I can extrapolate that there seems to be some kind of *intention*, *function* and *learning* related to vulnerabilities. Also *impact*, *feature* and *account* seem to play a role in searching for them.

5.2.3 Intentions

Intention seems to be a clear cut feature to technological devices. There is some idea behind a technology, otherwise it probably wouldn't exist, as we have seen in the section regarding inscriptions. Yet the intention of a vulnerability as is described by the definitions given is in and of itself nothing that exists by itself: The intention as well as the vulnerability is inscribed into the code by the programmer. Both exists in the code at the same time. The intention is the thing which should be there, which is the inscribed function of the code itself. However, there is a "dark side" to this code, an unintended consequence of the same piece of software. This unintended consequence results in a second de-scription, a second set of rules and possibilities:

> "Well, I would say, it's some behaviour from software, that is not intended and can cause harm. Or, maybe even it's intended by the developer, could be, of course, some bugs, there is a works-as-designed, but in a security regard you would say that's not okay."

> "A vulnerability is an error in the code or an... yeah, backdoor of some sort. Put in deliberately or on accident, which can be exploited by malicious intent to cause harm in IT systems. Again, could be a human mistake, could be error, could be some compilation mistake, but could be also put in deliberately to provide a backdoor for some kind of malicious actor."

> "I'd say it comes with experience. Like, if I see something, and I see a potential danger to it, then I might report it. […] Yeah, I'd say that, just, it comes with practice. Like, you try some things, then you see that there is something out of place that shouldn't do that, and then, yeah, you see all the potential that it has and then you just report it."

> "From a technical perspective that would anything that would lead to compromise. Like, in general, that make you-well, yeah, vulnerabilities would make your system less secure, but that's, like, redundant. Well, vulnerabilities are like, something, that was not accounted for that could lead to unpleasant things."

All of these quotes point toward that: An unintended, second form of possible usage. Mostly connected to some form of malicious usage or exploitation. This possibility for exploitation somewhat explains the discussions of vulnerabilities in a "security" realm. This I find particularly curious since this assumption of exploitation is an utterance which happens before the vulnerabilities themselves even have been discovered. The mere expectation of a possible vulnerability is connected to malicious behavior, intended backdoors or may just be human error. Every single one of those possibilities, however, splits up the meaning of software vulnerabilities, mind you:

> "You want it to work as intended, and therefore anything that doesn't work as intended I consider, first of all, a bug, and anything that's security conscious and is then therefore a security vulnerability."

This brings me to yet another viewpoint, that is the intention of the researcher themselves. If they are looking for vulnerabilities in the technology with intention (and not just stumbling upon them by accident), they have to have a certain mindset in place. Where does that come from?
‘

5.3 Finding Vulnerabilities

5.3.1 Mindsets and Corporeal Engagements

What mindset does one need to find vulnerabilities? The two most striking answers I got in this regard were probably the following two sentences:

> "Learn to build it, then break it."
>
> "Break stuff, hack users."

Within those two quotes there are probably summarized all descriptions I got. You have to have a good understanding of what you're doing. Building the technology, understanding how it functions, what else it's connected to, what privileges it has, what its parameters are. And then you may understand how to break it. How to find different pathways to do something else, which reminds me of "attack vector" as a geometrical (and epidemiological) expression as well as "reverse-engineering", which also has a strong directional component. "Hacking" users, as previously discussed, is in and of itself a contested term, yet I would argue that it mostly includes some kind of breaking, entering, extraction or other topologically readable understanding. Generally speaking, however, it related to some form of knowledge you have to have in order to come to the point where you may be able to find a different

understanding or potential use of the technology. What does it need to understand vulnerabilities? The most precise answers I got where the following:

> "Well, I feel like it depends on where you're looking for vulnerabilities. Like, if you're hacking websites, for instance, you need to know how websites work. And, usually, you can, like, look how-at how people found vulnerabilities and just like, try to copy them. That's usually the way how beginners do it first. But, anyway, at the end, you do need knowledge of how websites work and how to design your own sites. If you're doing web application security. Yeah, the other stuff, it's the same stuff. You need to know the roots to be able to exploit stuff."

> "Like, for instance, let's say you found an issue, whereby just changing a number, you can get other users information. Then you don't need to know about programming languages to try that. […] And then I googled and I found Python and I was like 'Okay, let's learn how to program.' And then, from there, I sort of spent a few years developing little stuff, like, for myself websites, you know, things that you learn, like, random stuff. And then got into security. And then you start learning more bits and pieces. And then it sort of falls back into piece [sic!], where you just pull everything together and you've got this knowledge about bits and pieces of everything that makes sense."

> "So, I'd say that learning about-learning a programming language, it helps, but there are some great bug hunters out there who don't know any programming languages and they seem to be doing okay. So, I think it mostly depends on how you do stuff."

What was possibly the most surprising insight here where the complete negligence of formal education as well as the what I thought were "classical" approaches towards programming and IT devices. That programming or coding wasn't necessary to find bugs and actually be an "successful" bug hunter was really somewhat of an unexpected finding. One little detail in the quote above I want to point out: The last quote mentions "how you *do* stuff", not necessarily the *knowledge* is relevant but also how you approach things and how you encounter those things. This, I would say, is quite similar to an *enactedment* or *performativity*, since it is not quite only to know about things but points toward a material encounter, be it typing, tinkering or some other form of corporeal engagement.

5.3.2 Practice and Enactment

This corporeal engagement can take many forms, I suppose. While doing the research, one of my interviews offered quite an interesting analogy what it means to do vulnerability research:

> "Imagine you're writing an essay, for your teacher, okay? And you make lots of grammatical mistakes and so on. Those are flaws in your essay. And you've got the teacher pointing them out and putting little red circles around them and saying 'Hey, you got to go back and you got to go fix that. Because otherwise people who read your essay may not understand it, it might not produce what you're actually intending to

> produce. It can have some-sometimes damaging effects, you might say something you didn't mean to say and people misunderstand that."

This interpretation of vulnerability disclosure is quite charming. But also it does hint at some points already made. There is, yet again, the expectation that there are just flaws in everything one may encounter. Without moral judgment, that is. Then there is an instance where someone points out the flaws, particularly *marks them*, having some kind of *pressure* behind it, a reference to *intention* as well as the *security* aspect and, last but not least, *misunderstanding*. The intention and the security I already mentioned what I mean by them. The *marking*, in this case, I understand as a reference to the practice of reporting, which we will come to in a short while. The *pressure*, the prompt to go back and fix them, invokes urgency but also some kind of authority, which is somewhat curious. This aspect will be discussed as well in the context of morality. *Misunderstanding* I will discuss in regards to trust and the public image of companies.

For the moment, let's concentrate on the analogy. Where does the exchange of this information happen? And how does it do so? The understanding of my interview partners what a CVD is was mostly framed like this:

> "Vulnerability Disclosure Programs, on the other hand, they are basically bug bounties without the pay. They are just a policy for you to-if you've something, if you came across something, tell us about it."

> "Yes, sometimes you do get paid on these random CVDs in the Wild West. You're still not bound by the policy. You submitted the report, that is done. With a bug bounty program, if you've got a policy and there's a payment, they're saying there is an agreement here."

What I found of interest here is that they aren't considered to be a strict contract as sometimes discussed in the literature regarding for example "safe habour" terminology. It is a policy and *possibly* an agreement, yet there seems to be some discrepancies here. If you just submit your report, why do have people trouble with that (as will be discussed *en detail* later regarding legal aspects to CVD)? And if they are not an agreement or a policy, what then? Mostly, these aspects were mentioned primarily in regards to bug bounty programs and the monetary reward. There seems to be a clearer understanding that if there's money involved, there has to be a clearer form of framework in place than just the reporting of vulnerabilities. However, to report vulnerabilities, one has to have a contact. How does that work?

5.4 Contact

5.4.1 Finding a Point of Contact

As a researcher or someone who has encountered a vulnerability and understands them as such, you may want to contact someone. In the best scenario, you find an e-mail address, a “security.txt”-file as mentioned previously or some other way of contacting the party you see as being responsible for fixing the bug. That would be the best approach as one interview partner explained:

> “So, if they don’t have a bug bounty program, then you look for a vulnerability disclosure policy. If they don’t have that, then you, like, search for a security.txt file. Which is the standard. And, if that doesn’t exist, you look for some like security email. Well, if that doesn’t exist, you go to Twitter.”

It is quite interesting that one of the most approachable ways to contact a company seems to be via their social media channels nowadays. Customer hotlines seem to have fallen from grace. When asked about the reporting strategies, this was not the only mention. In trying to reach a company or party to report a vulnerability and in the case there is no obvious or dedicated way to do so, social media seems to be a viable strategy:

> “I don’t know, maybe on social media or something. Where I know that there is going to be different people. Because it is not the same people who reply to email, that is the ones [sic!] who reply to other stuff.”

> “But I try the best I can to... Escalation. Get their attention. If that does not work and there’s, like, [a] threat to people’s safety then, yeah, probably, disclosure is the way to go.”

Here, we again have the security aspect to it. Combined with the urgency of doing *something* to prevent harm. Interesting in the first quote is the mentioning of different people which I understand as relating to different departments within the company. Between the people monitoring “security@<company>” or customer service addresses and social media, there may be possibly other priorities or understandings of what may be important to give notice. This hints at companies being not only one coherent block or entity in this discussion but having possibly divergent viewpoints. At least the people engaging with outsiders seem to follow their own attitudes to some degree, which only highlights ANT’s understanding of “*ex nihilo nihil*”- or, more precisely, everything stands in relation to something.

5.4.2 Resources in Reporting

When reporting a vulnerability, there has to be time and resources mobilized. Since they do not report themselves, some work has to be done to frame vulnerabilities in the right terms, to produce a document or what is called a "Proof of Concept" (PoC). A PoC is a piece of code or example of practice that the claimed vulnerability exists in some way. The reporting process may look differently, depending on the researcher:

> "So, if you're a bug bounty hunter and you want to speed things up, you want to template as much as possible, because you want to safe the amount of time you write. […] You sort of have to write a lengthy explanation of what you've got in front of you. So what I do, is, while having a template, I simply answer the 'W-questions'. The what, the why, in fact, the how and so on, covering the questions that the company would want. […] They don't want a Wikipedia definition of the vulnerability. They want to know 'Why does my company care about this right here. Please explain it to me.' So, I get straight to the point."

> "Like, if I find it and it's... like, it doesn't take me more than ten minutes to write a quick e-mail and then send it. So, I try to. Like... but then, if there's, like, some sort of danger, that I can see, then I may think about it twice. Because, in the end, if there's something and then there is no program or there is nothing, then you're risking them not liking it."

> "I try to, like, explain stuff more broadly. Because, like, I'm aware that many teams don't have security awareness or knowledge. So, I try to explain everything from a technical standpoint, assuming, like, they don't have enough knowledge in security. And also, like, I try to introduce myself. Like, 'Hey, I'm an ethical hacker.'"

What we can extrapolate from these accounts are, again, quite a few aspects to CVD. There seems to be efforts to spend as little time as possible in writing those reports, possibly hinting towards that being a annoying or unpleasant task. Secondly, you have to think about what your counterpart wants to hear. Third, you have to think about what your counterpart may understand or what their perspective is. Again, the interpretation of what you write them seems to be as important as how you approach them. A fourth part is the explanation of what the counterpart has in front of them. The possibility that they not know that strongly hints towards the question of knowledge, experience and understanding. The fifth part relates to the legal perspectives already hinted at which will be discussed later. Let's suppose the report is written, sent and has gone through (let's assume an e-mail, for the time being). How does a company react to such a report? What do they do with it?

5.4.3 Judgment Calls

As one interview partner put it quite eloquently, it is important to include a PoC into your report because otherwise you risk it not being taken seriously:

> “So, there is a famous philosophy in the bug bounty community called ‘PoC or GTFO’”

This has the reason, as is potentially clear by now, that there is an uncertainty attached to the reporting process. There are reports being written with quite a low effort, resulting in it being not taken seriously:

> “So, if I get these reports... they’re treated in the responsible disclosure process, but they’re usually not treated with the same priority as when I get a serious report with ‘Hey, look, we found this vulnerability, it is bad because of this, you can fix it with this.’”

> “We just look at it, if it contains certain key indicators, like... [we] mention the potentially affected products, they have tested this, ideally with the version information, they try to give their best in a vulnerability description and come along with a PoC or at least with a description, a verbal description, how to detect or demonstrate this vulnerability. If these parameters are in some form available, then we take it already seriously.”

While sometimes reports being treated with a different urgency than others because of the effort that went into it, sometimes it happens that vulnerabilities aren’t acknowledged because they actually are seen as something different:

> “If this is really an incident or if this is... as strange at it sounds, if it is a feature. Sometimes something is seen as a vulnerability, but in the end it is a business request, for example.”

> “It gets a bit harder, occasionally, with researchers who claim to have a vulnerability which we don’t see as a vulnerability. This also... can be often the case. But, in most cases, we can argue, based on even what the CVSS score is doing and explaining.”

This is quite curious indeed. While the first quote discusses potential vulnerabilities in terms of the aforementioned second de-scription, another way to use a technology, the second one speaks of something else entirely. The second quote invokes the possibility to discuss what vulnerabilities are, even though they are being discussed as something stabilized. There seems to be suddenly two parties, claiming two different things and arguing about the parameters, invoking such things as the CVSS score. Incidentally, however, I mentioned that even what information that goes into this score may be relevant is somewhat argued (see 1.2 Bug Bounty Programs & Coordinated Vulnerability Disclosure – A Specification). Here it seems to play the role of an impartial, disinterested party. This, in turn, strongly reminds me of the enrolment of actors as was discussed. Who invokes what arguments or

institutions? And who, ultimately, has the authority to makes the judgment calls on what is a vulnerability?

5.5 The Triage Process

5.5.1 Mediating Receivers

The triage process represents all of the activities a person employs to reproduce, examine and configure a reported vulnerability. That doesn't necessarily mean that this has to be a lenghty process:

> "So, in that regard, usually, I try to see if the report is good enough to send it-to just forward it. And, if it is, then I just forward it to the service responsible person. And, if not, then I will try to enrich with information, so that the people can understand them."

As already established, sometimes the reports speak for themselves and can just be transferred without being changed. Also in this case ANT concepts can clearly be seen. Either the receiving person serves as an mediator, someone who transfers entities without changing or manipulating it, or they interact with the entity and become and actor themselves. Specifically, the already introduced notion of "enactment" by Annemarie Mol (Mol, 2002, p. 32 see also chapter 4.2) seems *very* applicable.

5.5.2 Ontology, CVSS and Actionability

Whatever triage activities are performed, the CVSS score as an evaluation standard seems to come into play at some level. The CVSS score, as previously mentioned (see 1.2 Bug Bounty Programs & Coordinated Vulnerability Disclosure – A Specification), brings together some parameters to form a singular metric, "[…] communicating the characteristics and severity of software vulnerabilities." (FIRST, n.d., p. 1) While this seems to be invoked as an "objective" resource in handling conflicts over what is and isn't a vulnerability, there seems to be more examination methods, viewpoints and aspects related to this question:

> "Well, for the responsible disclosure I'm not sure if I would say that I have this matrix. I would say it's a gut feeling, but it's not, like, okay I get this and I [fully] trust my gut, because I know from the CVSS ranking I already can consider how critical will this be for us. We don't have a definitive matrix, now, in this form, or something like that."

> "But we usually see these standards as the baseline of what needs to be implement. And from this baseline on you can start to enhance and to customize it to your needs. […] So, yes, definitely, it is a factor, because in the end, a standard helps you to implement something which is comparable to others. But, it's not like the bible that you have to pray to, it's an indication and you have to see what you can do as well."

There seems to be additional contextual information needed to *really* understand the potential impacts and the relevancy of assumed vulnerabilities. CVSS is described as being one factor and a standard to make it *comparable*, yet it has to be put in your specific context:

> "Because it can be the case that it looks like a high critical vulnerability from the outside, but then if we have the context data of how does the protection work, how does the detection work, what kind of other influencing factors are there, there could be a low to none criticality."

> "And, of course, then, CVSS has its disadvantages in some cases. So, we also could rate one case as more urgent, even if it has a lower CVSS score, because of an anticipated possibility of publication or of technical risk that goes beyond CVSS or... yeah. Depending on various other factors, yeah."

Taking a standard that is in some contexts described as being some form of "neutral" agent and in other contexts being quite a fluid thing indeed relates, again, to the actors of ANT. Them being invoked, enrolled and made to be allies in certain debated and being alienated in others is quite interesting. "Enhancement" and "customization" specifically speak to this transformation of interests.
If there is a result, namely a vulnerability report will be taken seriously, the conflict is settled. We can say that the report is acknowledged, the vulnerability is "stabilized" and we've reached a state of "closure" of the debate. There is a common understanding that the vulnerability is "critical" enough, being pronounce worthy to be acted upon. The vulnerability is made *actionable*.

5.5.3 Stabilization through Communication

In practice, for the researcher this is an achievement in its own right. Having made a vulnerability actionable doesn't just mean that there is a "stabilized" item, but also that it can inform further discussions, documentations and communications:

> "But that you can somehow demonstrate that the vulnerability is there to the extent that the company can go away, you can give them some indications, some bullet points and so on, for them to go and research this further."

> "So, we acknowledge this and create a case internally, for that particular report and then analyze, first of all, handing over back to the researcher an unique case ID, so, which we're using in reference to all further communication with that researcher."

> "So, it's not-I will document every little vulnerability, usually, so I'm not really strict with 'Okay, that's not really an issue' or 'It's so small, we will not take care of it or will not document it'. So, then, yeah. Usually, I accept the vulnerabilities that are getting reported and then they are being documented in our [internal documentation software, anonymisation M.C.] form, where we do a rating, how big of an impact is this

> vulnerability, which services or which service is impacted by the vulnerability and... Yeah. Then, and in the next step, I will contact the service responsible, for the service that is being affected by the vulnerability. And get into discussion with him on fixing this vulnerability."

Specifically the case number here is a good example of how vulnerabilities then become actionable. While the report and the vulnerability in the moment of reporting being one and the same (is that report being acted upon? Does it seem legit?), only after the triage process are the two things separated again. In the communication with the researcher, however, they become, at least in the practices of one of my interview partners, *identifiable* by a unique number. This is similar to the CVE ID. The CVE is a system to organize publicly known vulnerabilities and to assign a unique identifier, whereas "[e]ach identifier references a specific vulnerability. A CVE ID enables automation and multiple parties to discuss, share, and correlate information about a specific vulnerability, knowing they are referring to the same thing" (MITRE, n.d.-a). This not only helps in the clarification which vulnerability report is talked about but also in establishing a (respectful) relation with the researcher:

> "Yes, we have. So, we have a formal answer template, so to say, for the first response and then also for the follow-up. This helps to give immediate feedback, so to say, while still having some time to analyse it. […] Well, if you report a vulnerability and you don't receive feedback for, like, two weeks, then this could be taken as a bad sign. And therefore it is really necessary to give immediate feedback, 'We will reproduce it in the next couple of days, we will do our rating, and we'll get back to you once we have our... so to say, scope, of the report.' But to not leave the one who is reporting waiting."

In the CVD, the report and the vulnerability at some point become one and the same thing, they serve, in ANT terminology, as "obligatory passage points" (Callon, 1984, p. 205). Only as long as the report is considered valid and the vulnerability as described within holds the "trial of strength", the triage process with the CVSS examination, the vulnerability exists. If the vulnerability or the report are rejected, the communication ceases to exist, the vulnerability vanishes. At the end of a successful enrolment of all the actors, if the report was convincing, the vulnerability held the trial of strength in the triage process and the receiving party acknowledged the report, a vulnerability becomes stable enough to be a token of exchange.

5.6 Social Tokens

5.6.1 Supply Chains, Code Bases and Trust

"Stabilized" vulnerabilities are quite important to inform other entities in the upstream or downstream of product development, since there are many different actors included in the construction of a

finished device. Distributed code bases and vulnerabilities among different products result in the necessity point out the origin of specific code (segments). Relevant here is the distinction between *vertical* and *horizontal* supply chains:

> "In a **vertical supply chain,** multiple products all share dependency on a vulnerable library or component. When the patch is developed for a given component, it can be used for all products. In a **horizontal supply chain,** multiple products implement the same vulnerability (from underspecified protocols or design flaws). Therefore, each vendor must develop patches for their own implementation of the vulnerability."
> (Schaake et al., 2018, p. 11 [emphasis i.o.])

Similar to the supply chains there has to be made "[…] a clear distinction between a *product* being vulnerable, and an *instance of a product* being vulnerable" (Householder et al., 2017, p. 6 [emphasis i.o.]). It makes a difference if, for example, an operating system currently running a browser is vulnerable or if the browser *itself* is susceptible to an attack.
In practice, this means that there has to be a lot of communication about what exactly has to be done about a vulnerability:

> "Sometimes it's a technical person, sometimes it's someone not with a technical background, then you have to explain kind of what is going, and what they should do. Sometimes we get software from a third party, so they will have to talk to a third party or establish a communication channel with me and the third party, so that we can discuss this."

> "And, also, the [country#1, anonymisation M.C.] CERT and the [country#2, anonymisation M.C.] CERT are among them. With whom we, for example, also exchange our vulnerability disclosures a few days ahead of the public disclosure. So they can also prepare the information, create their own documents, inform their audience and so on. This is an ongoing collaboration with them. And, with regard to other vendors' PSIRTs, we also collaborate with them on an as-need base, often we have the same vulnerabilities in their products-or, and our products, so we coordinate. Or, we are depended on them to... so, we also communicate vulnerabilities to the software upstream, the software we're needing and so on."

> "Yeah. I think, as software is getting more complex and software-as-a-service or infrastructure-as-a-service and cloud computing is more and more in trend right now and more and more being used, this vulnerability disclosure is getting really, really important. Not only that you are informed directly by a researcher, but that you're also informed by your vendor that something has happened."

Because vulnerabilities are hidden in the distributed code bases among and across companies, the need to share this information results in communications between them. This is a result of the vulnerabilities themselves, their potential impacts and *agency*, not because the companies necessarily want to engage with one another. The need for communication stems from the emergence and

introduction of vulnerabilities in the lives of companies, not because companies need to engage with one another:

> "I think the most important part is to get it fixed, no matter who is now responsible for it. What is definitely is a factor that you have to have a lessons learned session afterwards. […] So, this lessons learned session is definitely important, but without any kind of blame, but just to get to the root of it and to understand how could this happen and to prevent it from happening again. […] It could also be the case that some incident like that leads to a stronger SLA, a stronger Service Level Agreement to the vendor, to ask for specific additional security controls."

> "And we also keep relationships with, of course, the CERTs, so, the Computer Emergency Response Teams, in [country#3, anonymisation M.C.], because it is also important to communicate vulnerabilities that are not specific to one's environment, but that are in general in the public to distribute that as fast as possible to be able to also see if we're affected and if we have to respond to that."

> "We're also part of a what's called a [name of informal meeting space, anonymisation M.C.], where we can talk openly with other [business sector, anonymisation M.C.], without any kind of protocol. To discuss certain vulnerabilities, certain threat scenarios that are currently going and also concrete incidents that happened. In an open way, without any kind of protocol, without any kind of finger pointing, just to say 'Hey, look, that happened to us, let's check that it doesn't happen to you guys as well'."

The result of such communication spaces is a dense network of relations. This network is not only organized within the "official" structures of the CERTs and other institutions such as the FIRST, but also in "informal" settings within the same business sector.

The trust between affected parties is not only restricted to the companies. There is also the trust between the researchers and the receivers of reports that has to be taken into account. This relationship is somewhat more fragile, since it isn't governed by contractual obligations such as a SLAs, (inter-)dependencies or voluntary cooperation because of similar market position.

5.6.2 Trust and Legal Issues

In the communications between the researchers and the companies there often are, as already hinted at often times, tensions. Specifically tensions in regards to the legal aspects of vulnerability disclosure. Where does this tension come from? I would argue that this tension is a result of the material basis of the companies' network or code base. In my interviews, there were quite often references toward the *scope* of the CVD, the declared software or hardware parts that are allowed (or encouraged) to take a look at as a researcher:

> "And that's why we published the responsible disclosure program where we specify what is part of the scope of our responsible disclosure program and what kind of things can you do and what parts are definitely out of scope of the program."
>
> "And this should provide a framework for how we feel about responsible disclosure and how we expect people to interact with us and also what we offer in-in return for that. So, for example, that, if you use-exploit this issue and it's in the-within the rules that we've specified, then we will not take any legal actions against you."
>
> "You must not attack life systems, in order to exploit something, just to give to [company name, anonymisation M.C.] back a proof that there is some vulnerability in a certain [company name, anonymisation M.C.] product."

So, the companies are the ones setting the rules for what is allowed and what is out of scope. This, however, sometimes prevents researchers to declare issues they have encountered:

> "Because, I mean, everything you put online is available for everyone. So, I don't think it's normal, like, to say 'It's illegal if you poke at that.' Because, in the end, it's the web admin who has put that online, available for you, if you get what I mean? So, it's as if I have an apple and I put it, like, in front of your face. And then I tell you, like, 'Don't look at it!' But then, where ever you look, it's there. So, it doesn't make much sense. Like, they put stuff on the web and it's available for everyone, so, you cannot pursue legal charges, because I am just using your web. Then it's your job to secure it, if you don't want me to use it maliciously."

This problem potentially lies at the core of what is commonly called the "safe harbour" debates. The question of how to govern the relationship between the researchers and the recipients, whereas CVD is, as discussed in conjunction with the bug bounty programs, just one format among many. However, the question of how to construct the legal framework incidentally is rooted in the very material basis. The connection of the legal aspect (as part of our social world) and the governing structures (law) as well as the material basis becomes quite clear:

> "But, I think there needs to be a legal framework to incentive companies to have, like, vulnerabilities disclosure policies and bug bounty programs. Because, there has to be a way to report vulnerabilities. I'd say it's ridiculous that you cannot, like, report stuff. It's pretty bad."
>
> "So, this is definitely an increasingly important factor, and having these legal frameworks in place, like a responsible disclosure program, helps you to streamline this reporting process. Because if you have to really come up with a dedicated contract with this reporter and you have to find out all the terms and conditions every time once again, this just delays your patching process. And if you have already the framework and say 'Look, accept these terms, then we're fully ready to go' and you can then immediately start with the remediation."

In legal terms, the specific declaration what is in scope and what is out of scope therefore results in a situation where the researchers have permission to investigate whatever they would like. Yet even with these provisions in place there are concerns:

> "Okay, me, at the moment, I just hunt in bug bounty programs. Because I know I've got the specific permission to do it. […] But I can't do that, because it is illegal... In the end, even if you do it with good faith, like, they may not take that it that will. And they may pursue legal charges against you."
>
> "Except when I, like, try to report it outside of bug bounty programs. There is like that worry in the back of your mind that 'Oh, they [can send] their lawyers your way.'"

Every person I spoke to agreed that there should be some kind of rules, yet where to draw the line(s) seems to be controversial. The two extremes are possibly formulated best in those two statements:

> "Whatever they report to us, we look at it and try to find a common solution. It might come into legal discussions, let's say, if they find something which is really critical and they don't want to collaborate with us."
>
> "If a company has a product and some sort of liability there, especially with customers and so on, at the very least set up a communication channel. At the very least. I don't say you have to have a bug bounty program, I don't expect a reward, but at the very least a communication channel."
>
> "Beyond that, legal […] does not play a very high importance [sic!], because we want researchers to report vulnerabilities to us. So, it does not make sense to restrict-or try to restrict them to do anything. The more and transparent the communication runs, the better both benefit. Because we're-at the end these researchers are not our enemies, that want-try to do our harm, but we collaborate together in order to achieve the common goal to secure the environment for our customers."

Although there seems to be a common understanding that both parties want to "do good" in a way, or at least have the common goal to remedy the vulnerability, there is a constellation of conflict, an opposition. Sometimes, that is being seen as stemming from the expertise or background that people have, not as something inherently difficult to formalize because of how (material) networks of devices are organized:

> "We need more technical people to get involved there and to redefine what the boundaries are. Yes, there is a fine line. There is a fine line between what's malicious and what's legitimate, but I do think it needs to be there, formally, yes. To sort of encourage people that want to do good to do good."
>
> "Like, they think you're doing something malicious. Or sometimes they're like... they're scared or upset that you came across something."

> “Yeah, I-I-it shouldn’t be considered as a threat, but, I think you need some rules.”

The discussions surrounding the legal framework is often framed with formal definitions or other specific formal language. The safe harbour provisions are a tell-tale sign of this, and I have also encountered these arguments in my research:

> “Standardizing formal language, just, I don’t see what’s wrong with that, if you see what I mean. Does it hurt anyone? No. Does it potentially help someone? Yes. So why not?”

Taken together and seeing the problems arising from the formulation of a scope in the respective networks where researchers may encounter vulnerabilities, this results in what Amit Elazari Bar On means with their title “Private Ordering, Shaping Cybersecurity” (Elazari Bar On, 2019). The responsibility to formulate specific language and to give researchers the freedom to search for vulnerabilities without the fear of retribution of companies is negotiated between the researcher and the company themselves, in a *private* realm. Legal frameworks would shift this negotiation into a *public* sphere (incidentally creating both of them in the process and pointing towards the boundary between them). For the researcher, that means they are subject to the goodwill of companies:

> "Wo kein Kläger, da kein Richter."
> [“Where there’s no plaintiff, there’s no judge.”, translation M.C.; common saying in German]

Vulnerability researchers have found some nifty ways to get around this problem. Not only the strategies discussed in taking the reporting attempt to social media as previously mentioned, but also through the utilization of their personal social networks and connection:

> “Because that way I know, the relationships there, I don’t really need to think about the legal aspects of it. It’s established, it’s well established. They know me, I know them. When I go beyond that, that’s when it gets a bit difficult. And that’s when I have to rely on a middle man. Such, there’s a platform, where I know, if something goes south, they know me, they’ll be on my side.”

> “Because that will be a much safer approach. It’s not some random hacker, some stranger, contacting us, it’s so-and-so’s friend or so-and-so’s acquaintance. If that’s not possible, there is always the option of going via a CERT or something like that, a big organisation that would represent you.”

Which is, in the context of this thesis, yet another node in the building of what constitutes vulnerability disclosure. The “obligatory passage point” of the CVD involves not only the

technological side of things but also builds networks of people, invokes friendships, acquaintances or organizations and builds their relations in the process as well.

5.6.3 Politics

What was quite an astonishing outcome of the interviews was the complete absence of politics. At least not in the very obvious way I have presented it in this thesis up until this point. My understanding is that politics plays a big part in how to deal with vulnerabilities, however, my interview partners somehow managed to circumvent this topic and, somewhat contrary, saw politics to be absent of this debate:

> "But, normally, in the-on a daily basis, like, I don't think politics mixes up too much with cyber security."
>
> "People around me know, that I'm not particular political or anything like. I don't have policies and so on. Or ways and philosophies and so on."
>
> "But, to be completely honest, I'm not really active in these-in these political communities."

My understanding of politics however is slightly different. I would argue there is no such thing as politics being absent from something, especially if there is some form of conflict or debate going on. This is clear in how I write, I suppose. However, in researching this topic I wanted also to trace the political elements to this debate. I think they are always there in the arguments my interview partners made. In regards to the vulnerability brokers, for example, that is, the companies running bug bounty programs, I found some statements that could be construed as having at least a political stance to it:

> "So, a big, big, big factor is, what the brokers do. If the brokers decide 'We go in this direction', the industry goes in that direction, unfortunately, it's how it is. They have a massive influence."
>
> "And, thankfully, the big players, BugCrowd and HackerOne, eventually saw the significance of this, saw the push back from hackers and so on, and that had enough influence for them to start to make a, sort of, an industry-wide adopted policy, of sorts. So, they have the little safe harbor verbiage now in pretty much every policy in a bug bounty program."
>
> "I don't like this idea of security through, like, insecurity. Like, keeping everything insecure will not make us more secure. […] It's for the benefit of the people, very much. Like, in-it's kind of, like, speaking truth to power, at the end of it. […] Like, you can try to maintain stuff by making everything insecure but that is not truthfully secure."

In all of those statements, the very basic notion of politics, the distribution of power among entities and/or persons, lies at the core. This understanding of power of course can be broadened and deepened to include many other aspects as well, but in this context I thought it was at least curious that practitioners in this field don't consider themselves particularly involved in politics. However, I feel like there is an understanding of the *effects* of power, not only in regards to the issuing of legal guidelines such as a formal language or legal frameworks on what is considered hacking, but also in terms of a community effect of software vulnerability research in particular and how it changes things:

> "I think that, like, knowledge about vulnerabilities should be free and open to everybody. Because that's like how the whole-that's how the whole industry grows. By sharing information. And if we're not going to share information, then nobody's going to learn about these, you know, attacks and secure themselves. So, I think that, by disclosing vulnerabilities, not only the researchers learn, but also companies learn. And the developers at companies learn. [Mhm.] And if you're not going to disclosure vulnerabilities, that just only benefits like two people in the transaction instead of, like, having to benefit the entire company-community."

> "You know, everyone uses computers, nowadays. Like, and yet, computer science courses are optional, at schools, which doesn't make sense. Because, you are making people learn about, I don't know, literature or... in, second languages, but then you are not teaching them how to use computers, which they are using on a daily basis?"

Probably one of the most explicit statements in regards to global politics I got was in the context of hacking. Discussing vulnerability research and the moral implications of doing so, what to do about the vulnerability and how to avoid legal retribution ended up being seen as somewhat devoid in the digital realm:

> "I mean, it's all funny, of course, you can do a lot in the law and also in our rules, for our responsible disclosure program, but then, in reality, there's still a good share of people, they don't care about the rules. You know, in the internet, when you come from a Russian IP or a Chinese IP, no one cares about your rules, about your country."

Which, of course, is not meant to be a reason to abolish all efforts to establish rules or laws governing this space, but it points towards the many difficulties dealing *with* this space and, possibly, how many actors and their diverse interests are being negotiated in this realm and, ultimately, shape it.

5.7 Morality, Incentives and Markets

5.7.1 Help and Incentives

Morality itself is often located in philosophy, being a realm of its own and not necessarily attached to technology, let along software and code. However, in this research, we encountered already quite a lot of instances of morality as well as references to something being "good" or "bad". For example the wordings of "white hat hacker" in contrast to their malicious counterparts "black hat hacker" speaks to this. Also in doing my interviews I stumbled upon this notion of morality being a philosophical stance:

> "It's a philosoph-in my opinion, we are starting to go into philosophy. There is no, necessarily, a clear cut answer that I [can] give you. Personally, I can only talk from my process. I want to disclose it if I can. I want to get it to the respective parties, I want to help, if I can."

There were quite some references towards the aspect of "help" being provided to companies without having any expectations, just "doing the right thing", even though that sometimes is seen as not having to do something with morality, being a separate category altogether:

> "So, no, there is no morality. I don't think morally I'm obligated to do that. […] Because I don't really hack with the incentive of making things better and so on. It's very much a selfish thing. […] I want people around me to be happy and I do that by doing what I love doing. And that's learning, building, breaking. And so, if I find something I can break, I want to communicate with the team, I want to have a relationship with the team, I want to help them, have a good day, get on."

The notion of helping, regardless of being seen as a moral stance or not, clearly speaks to the notion of "matters of care". Yet another instance were "help" was quite often invoked was in regards to improving services, products and software in general:

> "The first ones research something or find out something and they want to report it to be fixed. Those are the white hat hackers that usually report to us. […] So, they want to help us."
>
> "And then you have the other researchers that they just want to-yeah, see that things are getting better."
>
> "Because, in the end, you're doing, like, a good deed. Like, you're just telling them something you've found. And they're just glad that you're telling them."
>
> "Because, like, if it's sever enough, then they need to know."

If you decide to help, then there was given in many instances some kind of reward. This was given sometimes in the form of goodies, also called "swag" (that is, small objects with the company logo, for example) or sometimes monetary rewards:

> "Then, we had some Goodie bags that we sent. With some merchandise from the company. Or, at one time, someone reported an issue to us, and, then, he was also a customer for one of our plans, and then we said 'Okay, you get three months for free.' […] And... we do that. But that's a limited scope and it's not for everyone. And it's being decided on a case-by-case basis."

In the CVD, as discussed, this is not something you should expect. The CVD itself is seen as just the policy in place and the point of contact. There seem to be, however, always some people how try to get something out of reporting a vulnerability. Which is quite curious since that means there is an instance of making a vulnerability a tradeable good, transforming it from a mere technological effect of software to a commodity or, at least, something of worth. If you contact parties that have a CVD policy in place but not a bug bounty program, however, this is commonly seen as a "bad" move, and even has its own name, being called "Beg Bounty":

> "They expect you to pay a bug bounty. And... so, they're reporting everything, that is also the smallest thing that could be considered a vulnerability."

> "But there are no rewards. And then, sometimes, communication turns more in a bad way, because then they expect a bug bounty. Or they are really starting to ask you for a bug bounty and that is a little bit uncomfortable."

> "We also experience some people who are not happy with our approach, that we don't have a bug bounty program, for example. We only have a hall of thanks. So, we credit them in public. That we acknowledge their work, but there is no money associated with it. And there are researchers who are frustrated by that and then go to others and sell the found vulnerabilities"

Most often than not the first and only reward a person reporting a vulnerability gets in a CVD is a position in an Hall of Fame. That is, a list of people having reported vulnerabilities, sometimes ranked following specific systems, sometimes just being an informal acknowledgment. Whatever the form, the idea is that the report is transformed into bragging rights or reputation. These acknowledgments also help in bolstering the Curriculum Vitae or employability of the respective researcher:

> "And for individuals I think it's mostly the fame and the reference."

> "But you can, kind of, you can give a thumbs-up to the researcher or you can write something on their page. And that's things that we do. If we get a report. […] And then-that's the least we can do and that we do in that regard."

> "At the beginning I'd say it was more because of the fame. Like, I search for (bugs) in BBC, Apple, Intel, big corporations, so that I could get into their hall of fame. Because that was cool."

The prospect to end up being mentioned in the Hall of Fame of big company seems to be one of the first stepping stones or incentives for researchers to enter vulnerability research. This Hall of Fame possibly stems also from the "Capture the Flag" events in "hacker communities", whereas the reward is more often than not also reputation. As we will see later, however, this incentive comes with some caveats as well, namely the outsourcing of work.

5.7.2 Public disclosure/Non-disclosure

As shortly mentioned in the introduction to this thesis, there are two extremes to vulnerability disclosure. The first being public disclosure, putting the vulnerability for everyone to see in a forum post or otherwise making it publicly accessible:

> "I think that's-that's completely okay and normal. A good thing even, I think, yeah. When the vulnerability is fixed, I think there is no real reason not to go public."

> "I just does [sic!] once, like, it has been patched and fixed and I've gotten their permission. Like, I don't want to disclose a vulnerability that still exists."

As this quote shows, however, there are sometimes negotiations about the timing of making vulnerabilities to a broader audience. This stems from the possibility that at any given moment a vulnerability may be discovered by someone else (Johnson et al., 2016). These timelines become of importance when thinking about reporting a vulnerability:

> "Like, you don't want other people, malicious people to exploit it. So you gonna report it."

> "I mean-[we're trying to explain helps], like, we're trying to tell people 'Here, we have these skills, and you want to patch your systems before, like, somebody else comes and finds these vulnerabilities.'"

Generally speaking, the knowledge about a vulnerability puts person in a moral field, if they want to acknowledge that or not. To report or not report becomes the question. There were, however, instances were also the other extreme, non-disclosure of the vulnerability, keeping it a secret, was mentioned:

> "Because I don't owe them, like, I'm not working for them, they are not paying me... Like, it's not my job to tell them. So, I can just keep it for myself."

Secret, in that regard, may only be the finder themselves, without reporting the vulnerability to whoever is deemed responsible for rolling out a patch. These can be sometimes singular people, sometimes companies specialized in vulnerability research, sometimes nation-states or state-sponsored "hacking groups". Those "hacking groups" are also often called "Advanced Persistent Threats" (APT) (Greenberg, 2019, p. 301). They all follow their own motivations and goals, so there is not just one answer to have to why people may keep vulnerabilities secret:

> "I believe that any vulnerability, which is discovered by person X and is not disclosed, but used by that person or that organization, for legitimate purposes, that could totally be legitimate, but the probability, that the same vulnerability is used in a malicious way, through other actors, is, from my perspective too high that it would justify the use."
>
> "And, it's unethical to just keep vulnerabilities to yourself."
>
> "As soon as there could be multiple companies, multiple institutions affected, and you keep it secret, then it's pretty much a zero day. And just waiting to be leaked and waiting to be put in the wrong hands. So, you have to differentiate between, here, where only [company name, anonymisation M.C.] is using this software, then totally fine, keep it for yourself, but, if there is a vulnerability as well included in a certain type of open source library, in some kind of technology that you can buy, then there has to be some kind of sharing."
>
> "And I would not encourage to hold vulnerabilities undisclosed and exploit it by entities."

The other possibility is to involve the company, report the vulnerability, but afterwards keep the information about the vulnerability a secret. This possibly has to do with the vulnerabilities being seen as liability for the company in terms of reputation.

5.7.3 Reputation

In the case of not making vulnerabilities public after the report, the information cannot be used further to inform other researchers or companies to gain new knowledge. This may help the reputation of a company in the short term. But since "[a]nnouncing successful breaches would harm a firm's reputation and negatively affect its market value." ((Pala & Zhuang, 2019, p. 181), this may be considered harmful to the reputation of a company. With the publication of vulnerabilities there is also connected an acknowledge of a security incident, which in the eyes of the customers (but also by other opinion-forming devices such as in journalism) this could result in a potential harm:

> "So, this heavily depends on the industry, for example, in the [business sector, anonymisation M.C.] we rely on the trust of our customers. If something like that is being publicized and it's been wrongly understood by a certain news outlet or a certain group of people, then this could definitely harm the trust in institutions."

This is probably the most extreme case and is definitely not true across all industries or even true to all companies within one industries. There are doubts and problems associated with this, but generally there was more an understanding of the positive aspects to vulnerability disclosure:

> "Because, of course you have different opinions within the company and some say 'Any vulnerability you disclose is a sign of weakness of our company and we don't do errors. Don't publish it.' and so on. But the majority now and the process has this also included, so these discussions, although they pop up, occasionally, for a particular case, but, normally, this is no longer an issue. Because the majority of people very well know that it is contributing to a good reputation. Because every software may have vulnerabilities, every software has vulnerabilities, it's only a matter when and by whom it is detected."

So, even though the public disclosure of a vulnerability may be harmful to a companies reputation, there is an understanding that the same publication may result in possible harm to unintended targets:

> "And, you can do harm to the reputation of the company, if you want to that and if you want to, for example, the world to see how fucked up something is with them. […] You endanger all the other customers that might be affected by the vulnerability. And I think that is something you should not want to want. […] If you have, like, good motivation, then you would not want to want for other customers or people that interact with a service to also suffer from the same vulnerability that could be exploited by someone. So, I would say, full disclosure from this perspective… I can not do anything against it anyway, but I wouldn't say that it's a good thing because of that. […] I mean, fair enough, if you do it, if we cannot fix it."

> "So, yeah, I think for some it's like the motive to just see the world or see a company, a service, getting better. Or to complain, also, even, about an issue, because, sometimes, you know, they're right, it just shouldn't be there."

There is yet another aspect to disclosure that has to be discussed and is hinted at here. For researchers to make vulnerabilities to pressure companies to do something about their faults brings me to the aspect of compliance.

5.7.4 Compliance

Companies are seen to have a responsibility to their customer base. This can be, yet again, be understood very much in the context of care. Companies have to mobilize resources to deal with complaints, failures or other issues they are made responsible for. Similar to the very discussion of

establishing a CVD for having a point of contact, there is a recognition for the need to remedy vulnerabilities:

> "It's also our-we have to fix issues. And we have to fix issues that people do not report us. So we have to make sure that the quality of products is good enough to withstand attacks."

> "Because, in some way, if we offer a service then-and we offer the service in a bad way or with vulnerabilities, we are also in some way the bad guy, because we are having vulnerabilities, we are not offering the full quality. I think there is also a lot of responsibility on ourselves as well. Or, on the people that produce software or offer services to make sure that these are secure."

> "I think, yeah... I think you have the responsibility to keep your system secure and to not be... so, that your service is not being abused or your systems."

> "But, we also had reports about vulnerabilities that took much longer than ninety days. And, when this happens, we try to be as transparent as we can about the issue, but sometimes it just takes longer and then we ask people to please not go public with the information until we fix the issue."

> "No, I think it should be, like, if you build something, and you know it is critical, then you need to make sure that it is also secure. Like, it is no one else's job to do that. Because, I mean, you're building something... Same as you make it look nice, you need to make it be secure."

So, there is not only morality attached to vulnerabilities and their remediation process, but also a long chain of responsibilities. Going from the researcher finding those vulnerabilities and the decisions they have to make (depending on their world view) to how companies decide to make them public or how they position themselves to their customer base to the very production processes of products, services or software.

5.7.5 Markets

There has to be mentioned another aspect in regards to morality, namely vulnerability markets. I mentioned them in passing in the (see 1.3 Parties, Protagonists and Organizations). There are markets for vulnerabilities. Some are considered legitimate or "white", where security companies or governments may shop for vulnerabilities, some are seen as illegal or at least shady, called "black", similar to the distinctions made in regards to "hackers" (Libicki et al., 2015):

> "And then there are the ones (reports, addition M.C.) who are investigating and researching and then want to sell the vulnerability, for example."

> "So, you cannot prevent people finding exploits or there will-there is always the possibility that people find exploits, they don't tell you about them and then they sell it."
>
> "I mean, morally it's questionable, but of course, I can see, if you can earn-for example, if you report a bug to us and you get nothing, maybe a goodie bag, versus you, for example, sell the bug to some site and you get fifty thousand dollars... Yeah, you know what makes more money, but from a moral standpoint I'm not sure."
>
> "And, the third part, then, is definitely the ones that are actively searching for vulnerabilities with a malicious intent. So, either receiving some ransom payment or selling it on the darknet for some money."

Vulnerabilities may be seen as a commodity or at least have the potential to be one, they can be bought and sold, sometimes scoring quite high prices as can be seen here. This move, however, is quite a transformation that has to be done. Similar to intellectual property, these commodities have some relevant properties: Software vulnerabilities, or at least the knowledge about them, are reproducible, for one. Anyone can know about them without them loosing their base item. This stands in contrast to, say, an apple or any other tangible good. A tangible good disappears at some point if you divide it too often. This has the strange effect that selling vulnerabilities results in diminishing returns.
Consequently, there has to be a limited supply, meaning you have to exercise a certain amount of control over it. The worth of a vulnerability primarily stems from how many people know about it. If you make a vulnerability public, therefore, it looses a lot of its potential value.
Another aspect on where vulnerabilities derive their worth from is how much you can do with it. This is indicated, for example, with the already mentioned CVSS score. These parameters define for example which software is affected, how many people use that or how ubiquitous it is as well as how easily it is patchable or detectable.
Ultimately, there are many considerations that can go into vulnerability pricing (for some see Laszka et al., 2016; Libicki et al., 2015). Generally speaking, however, it is enough to say at this point that a vulnerability has the *property* or *potential* to be a commodity, this aspect is part of its *ontology multiplicity*.

5.8 Maturity and Management

5.8.1 Commitment

The implementation of a point of contact is not something to be done without a second thought. Generally, setting up a point of contact is considered a good and helpful idea across all my interview partners:

> "If it says 'You have to have a contact address for security topics' I think that probably wouldn't hurt and would be a good idea."

Yet, with that being said, there always was follow up. After the ideal was acknowledged to be probably useful to have, there were some limitations brought into the mix. This was mainly connected and seen to be an issue to set up since it is also a management decision:

> "There is a whole management process there, you've got to actually triage the vulnerabilities and so on, but at the very least it should be considered."

The problem, however, comes with setting up this point of contact. Once you've made this decision, you can't easily track back. You have to *commit* to actually do something about them. This is also in regards to the aforementioned company compliance. This can go two ways, either one "public" sees that you have security issues and you loose their trust. The other possibility is that a possible different "public" understands that there just are vulnerabilities and it judges a company in regards to their *handling* of the vulnerabilities reported to them. Those two public spheres may exist at the same time, consequently it is maybe better to speak of multiple "publics". Going back to the point of contact, the viewpoint on how companies handle vulnerabilities result in a commitment issue for them:

> "Or, what I learned in the last year was, kind of, okay, we-we have this responsible disclosure or the CVD, that you mentioned. Then we have bug bounty and then we have maybe managed bug bounty. I think you have to clarify first, within your company, what are your expectations and how-what do you want to do? Before you do any of these, you have to commit yourself to fix issues that get reported."

> "Because if-of course, you can say you have a single point of contact, but then you-I think you really have to commit to reply to things that get reported there. Even if there are sometimes wrong reports, false reports, things that have nothing to do with the security of the website, maybe some complaints. So, I think it only makes sense if you're also committed to take care of the things that get reported there."

Following up with the sentece:

> "Of course, you could put them in the trash immediately, but then it doesn't make sense at all."

There has to be a commitment, therefore. But with a commitment come larger decisions to be made, it is not just establishing this contact address. You will need to invest further resources to do so, having people actively monitoring this point of contact or address (which cost money), you have to have a

triage process in place etc. These decisions are not only of a managerial type but also are deeply connected to the (material) networks of the companies themselves:

> "So, you would need to start with a single or with a narrow, isolated part of it to clarify and define the real rules, what can you expect? But then, yeah, all of a sudden, you open a can of worms, because, then the people are starting 'Yeah, why this product line and not that product line' and so on. So... I think... that's a conflict to solve first."

The mobilization of resources is a critical aspect to these discussion. If you decide to get reports, the amount of money you spend is an indicator of how many reports you will receive. Similar to the properties of a commodity, a market mechanism declares that if you put up a lot of monetary reward you will have, a lot of people searching for them since the potential reward is so big. This follows the logic of "given enough eyeballs, all bugs are shallow" (Elazari Bar On, 2019, p. 231). But, not only the monetary resources are a decision to be made and accounted for, the resources of people handling them come into play as well:

> "And then, the second one is, if you want to spend money on it. And how much money you want to spend on it. […] And... so, we said, on the one hand, we want to get reports, but we don't want to get overwhelmed with reports. So, that's why we don't have monetary rewards. […] You have to know for your yourself how many reports do you want to get, what kind of reports do you want to get and how do you want to handle them? Or, how many can you handle them? So... I think that's something important to decide on if you want to jump into this."

> "I think I would even go with a managed bug bounty, because there you can say 'We only want high and critical vulnerabilities' and you can also say, okay, not every hacker on the world can now try to attack you but this managed bug bounty service will also choose which kind of people can interact with your bug bounty program. And, I would do this for critical and high vulnerabilities first, because then I think then it makes sense to pay."

Taken together, there has to be a commitment in actually dealing with reports if you decide to implement such a process, you have to think about what scope you want people to look at your assets and you have to think about what *form* this reporting process will take. There are, as mentioned, a lot of viable mitigation pathways.

5.8.2 Hierarchy of Maturity

The basis of these vulnerability handling formats is often discussed in the terminology of "maturity" (e.g. Householder et al., 2017, p. 23; Votipka et al., 2018, p. 386; Woszczynski et al., 2020, p. 13). With that maturity comes also a hierarchy. Some companies are seen to not have reached a level to deal with their vulnerability reports in a way. Some are even called out to start *somewhere*:

> "So, it's definitely being discussed in the circles that I'm in, knowingly that some are still not ready for that. And they should do their homework first, establish these internal processes, mitigation processes, rating processes, before you get bombardment with information from the outside."
>
> "Or with the more basic topics they don't [get] right."

So, at the very basis, you have secure coding practices and other internal processes. First, you are expected to sort out your strategies, without relying on outsiders. However, the view to have this outside perspective is also always present:

> "Because, alone you will not be able to handle and to find everything and to take care of everything. And, so you will need the community."

A short excursion from this quote is the reference to the community. There were many accounts of the researchers being one somewhat stable "community", the hacker community or public reporters and so. I disregarded this aspect for the thesis at hand, yet it is definitely a topic worth looking into, how these communities are defined, made or work. What I want to mention at this point is another shift between "public" and "private": The costs of "doing security" or the distribution of responsibilities. It is quite a entangled topic with the commitment to fix vulnerabilities, their potentials in being harmful and so on. But I think one very important aspect to keep also in mind in this relationship between researchers and companies is the distribution of "doing work":

> "In that point of view, that there are so many things getting reported, and, if you have, like, a bug bounty, then you have some kind of crowd-sourcing your security, right?"
>
> "So, we started with a responsible disclosure program, which is publicly available, the email address, and we will also include it better in our websites as well. But, we started with that, to give somebody who passively found something, who, by chance, or actively searched on our website, or in our web applications, and found a vulnerability, to give them a coordinated way to report it to them. So, this is working out and this is established. Now, from a maturity perspective, the next step would be not to rely on somebody finding something by accident but really to actively encourage the security community to look for certain bugs in our systems. So, this is from a maturity perspective the next step and we are currently in the process to find a suitable partner, who fulfills all of our requirements and who fulfills the necessary also legal requirement to work together with us."

Even though my interview partner talked about bug bounties, where there is a promised monetary reward, the same applies to CVD as well (as can be seen in the wording of "actively searched", for example). The work being done is by the researcher, without the promise of pay. So, on the one side

we can see a commodification of vulnerabilities, on the other hand we can see an “exploitation” of the people doing the work since there is more often than not no pay associated with the work done. I am not sure what to make of this thought, nevertheless I wanted to point out this opposition.
What we can see, however, is that bug bounty programs and CVD are quite often taken together in these discussions. As established before, the managerial decisions to be made at this level introduce some overlaps between those structures. Interestingly, CVD as well as bug bounty programs are often seen as being quite at the top of the hierarchical structure of company security maturity:

> “I think responsible disclosure or bug bounty is, like, it’s not a foundation. If you look at the pyramid, it’s more...rather on the top. So... I wouldn’t say that you have to have something like that. To make it obligatory, because there are many other things in the permit. On the foundations, that are more important to have right then to have the bug bounty.”

> “So, I think there is some kind of journey, that you’re taking, when it comes to security maturity. The first part is getting to know yourself, so, what are your internal process, what are your internal devices, IT assets, marking them, classifying them, rating them, getting the data flows between them. Then, scanning them for vulnerabilities on your own. We internally, in the network, doing internal penetration tests, doing internal check-ups. Then, of course, the outside views. Starting with external vulnerability scans, starting with external penetration tests. Then, I think the next logical step is a responsible disclosure program. So to say to give a legal way to report such a thing. And then, the-near the pinnacle, so to say, you have then the bug bounty programs.”

The hierachisation of security practices in regards to vulnerability research is quite manifold problem. Do you start off with “just” having a point of contact so that you’re aware of a potential security risk or issue? Do you start off by trying to fix your own problems first? How does this relate to your resources? It definitely makes a difference if you are a singular person running your private website on a home server or if you are a big corporation with an IT department with hundreds of people. Or a flower shop which runs a webshop and has responsibilities towards their customers but doesn’t necessarily have the resources to deal with their IT infrastructure. Or possibly even a nation-state with departments tasked with securing critical infrastructure. So, the CVD process is quite a chimerical thing. Depending on in which context you talk about CVD and vulnerability mitigation strategies, this can get complicated fast.

5.8.3 Precedence and Guidelines

The creation of a CVD process and introduction of a point of contact is not only a managerial decision and results in an ever-growing decision making process, but may also result in *structural* changes to the company:

> "And, from our experience, multiple times, already, experienced that, if there is one-if there is a first serious vulnerability to handle, then the companies… yeah, got the incentive to create such a team."

The potentials and capabilities of a technological entity, following the path laid out here, therefore result in the implementation of new department, hierarchies and institutional changes. The ontological state of a potential harmfulness of vulnerabilities results in the implementation of handling process, which ultimately manifest in the creation of dedicated groups of people dealing with them. But also the structural frameworks guiding the groups of people, the processes are formalized. Not only in the form of guidelines and "best practices", but also legal definitions follow with "maturity". This may explain why only in recent months and years there was movement in this direction. Before that, the mitigation processes follow a principle of "try and fail". Yet, at the same time, the introduction of more "stabilized" items such as guidelines, frameworks, teams and strategies may result in ever-expanding circles of engagement in this direction:

> "Because, obviously, it-not only sets a precedence for their agencies and their-the federal agencies and so on, but, equally, it sets a precedence for others."

> "They can do it, you can probably do it. So it's not necessarily just from their perspective that it would help them, they're very much doing it also in the interest of setting an example and encouraging other agencies and organizations to do so."

This adaptation across industries or companies is also discussed in the literature consulted. Not only in regards to the specific practice of vulnerability research or mitigation, but generally as a ways to "make things" work:

> "The Internet, and IoT work, only because groups of people develop 'standards, best practices, and guidelines' that others' follow. Why do others follow? This is because it is in their best economic interest to do so."
> (Kovac, 2019, p. 53)

Therefore, I felt the need to ask my interview partners a last question regarding their views of the future of the CVD as a practice of vulnerability research and handling.

5.8.4 Future of Vulnerability Research

The future of vulnerability research is seen in quite some diverse ways. One aspect I encountered during my interviews was that the CVD process as a point of contact which doesn't offer any reward just will stay-as-is, being an institution which will exist in parallel to specialized programs with a payout structure:

> "It will coexist, yeah. I think bug bounty programs often attract researchers, who only want to get money and apply a certain standard tooling on finding some bugs. To earn their money. That's a legitimate way of earning money. On disclosing things. The other section still remains in parallel. I don't think that one will supersede the other or vice versa. But... both have their legitimate use cases and will continue to coexist."

Some aspects in regards to the future of vulnerability disclosure therefore concern the internal development of structures, streamlining aspects to product or service development:

> "Hm, I would say, unfortunately, we are not at this point as to really have to say 'Okay, we get a lot of cross-site scriptings, so we should get a feedback to the developers.'; 'Look, we get-eighty percent of our disclosures are cross-site scripting vulnerabilities, so, we say, we focus on this.' We-we don't have this feedback loop yet. […] But, of course we still have secure coding topics. […] It's just not connected yet with the responsible disclosure program."

This aspect further deepens my argument that technological entities shape and form their surroundings, as they are implemented also in already existing organizational structures and processes. In this case, the feedback between vulnerability reports and their handling further informs downstream or forward processes, things that lie in the future through having cycles of engagement. We already have seen another instance of this with the previously mentioned "lessons learned" session in the dealing with vulnerabilities in supply chains (see 5.6.1 Supply Chains, Code Bases and Trust).
Some of my interview partners brought up problems in relations to the bug bounty programs. The Hall of Fame as is now is problematized in the form of clogging the system, thereby having "negative externalities" or unintended consequences for vulnerability research and reporting as a whole:

> "So, that's where my point comes about, like, platforms advertising it as a cash grab. Like, when they market bug bounty programs as a thing, it's just 'make quick money', it ends up getting a pool of researchers that just want to get quick money. Which incentives lower severity reports. And makes the whole thing worse for everybody."

> "A joke response is because they're a marketing team. Basically, their interest is, if they can, obviously, from just a purely theoretical point of view, they want to have everyone on their platform. Right? If they could have every single company on the world paying them, they will do that. […] They-whatever decision they make very much shapes the vulnerability disclosure process for the company. They will either like what they see with the platform or hate it. And as such, it might shift their view on vulnerability disclosure as a whole. […] What they then associate with CVD will suddenly become HackerOne. It will become synonymous as such."

This point of view also relates to the managerial decisions to be made within a company, how to handle the reports. How many resources does one want to mobilize? Who are the people who are

getting paid or do the work in searching for vulnerabilities? The question of the mobilization of resources ultimately leads to the question of how much does one *want* to care about ones own technology:

> "Affirming that care is necessary to maintain technologies, even technologies that are not necessarily desirable or even harmful, so that they continue to work well opens to further ethico-political interrogations, such as: What worlds are being maintained and at the expenses of which others?"
> (Puig de la Bellacasa, 2017, p. 44)

What world do we want to maintain in expense of which others? How do we want to care for technology? What practices do we employ in this care? And, ultimately, is CVD a practice of care in this regard?

6. Conclusion

The thesis at hand had two goals in mind: Understanding the emergence of software vulnerabilities as a entity in its own right and the formation of what is considered "social structures" through the Coordinated Vulnerability Disclosure process.

Science-Technology-Studies, the academic field within which this thesis was written, provided the broader influences on how to go about investigating these goals. Actor-Network Theory provided the theoretical groundwork and concepts in finding an appropriate language to explore this field, as it doesn't discriminate between humans or non-humans. Everything affects *something*, therefore having a "neutral" (Michael, 2016, p. 26) language in place helps a lot.

Why is this relevant? In STS and other academic disciplines an turn towards the influences of material objects and their emergence, commonly known "material turn", can be observed (e.g. Mutlu, 2012, p. 173ff.). With that change of perspective, new challenges in the way we think about technologies, the interactions between technologies and a "social" realm as well as the *situatedness* of "the humans" in this realm become of interest. At the same time, the material turn introduces new and exciting intersections of different research fields such as "Critical Algorithm Studies", "Political Ecology" (cf. Bennett, 2010), or "Critical Security Studies".

In reference to the topic of this thesis, CVD, this theoretical approach offers a fascinating site to interview the digital sphere. As I have discussed before, most of the offered literature and documents discuss the CVD process in the context of "best practices" (ENISA, 2019b; ENISA & RAND Europe, 2015; FIRST, 2020; Householder et al., 2017; Schaake et al., 2018) or possibly as standards (ETSI, 2021; Foudil & Shafranovich, 2021; ISO/IEC, 2018). The move towards legal frameworks is only recently taken by *some* countries for *some* of their institutions (MIITNS, 2021; U.S. Department of Homeland Security, 2020). The discussion surrounding the need for safe harbour provisions (Elazari Bar On, 2019; Kilovaty, 2017; Weulen Kranenbarg et al., 2018) introduces the need for a more precise language for vulnerability reporting processes.

With all that being said, however, this thesis was meant to come closer in understanding the very diverse factors which build the groundwork of policy documents, standards and scientific studies. The

main research question guiding this endeavor, "*How do software vulnerabilities inform the formation of social structures?*", speaks to that.

6.1 Software Vulnerabilities in CVD

The first observation made is the "boringness" of vulnerabilities. In my research, the documents encountered and people interviewed talked about vulnerabilities and their potentials in quite unexcited terminology. Which I would argue speaks, together with being apparently ubiquitous, to them being quite a "mundane" thing, nothing to be alarmed about. With that comes an expectation for researchers to encounter them at some point. This process of discovery is described as being deeply connected with "social" and corporeal things such as feelings. With the encounter of potential vulnerabilities comes an exchange and negotiation of intentions between the software and the researcher. *Should* the program be able to do that? This also speaks to the knowledges applied, needed and mobilized in finding vulnerabilities, the skill set needed to identify potential vulnerabilities. These processes, "[h]ow to do things with words and then turn words into things is now clear to any programmer" (Latour, 1992, p. 255). There is not only one specific meaning inscribed in a device (or also a program), there could be many different ones. To de-script a program in more than one way intended could be seen as a form of "hacking", merely a being "mis-use": "No artifact is idiot-proof because any artifact is only a portion of a program of action and of the fight necessary to win against many antiprograms" (Latour, 1992, p. 254).

Only after all these steps is it possible to transform *potential* vulnerabilities in something that may be described and made *actionable*. Software vulnerabilities change from a "matter of fact", something that is just there, to a "matter of concern", something that carries potential and may be harmful or disruptive in other ways. The finding here is that vulnerabilities are an assembly of diverse practices, notions, ideas and technical devices (computers, programs,…) representing the material realm.

6.2 CVD in ANT

Similar to the software vulnerabilities themselves, the CVD process has to be examined carefully. CVD is a site of mitigation of the potentially harmful effect of software vulnerabilities. This genealogy, therefore, is deeply rooted within the material realm itself. However, in my thesis it is described as (primarily) being the site where the complex exchanges of knowledge, ideals, ideas, opinions, standards and tests *about* software vulnerabilities take place. The finding in this thesis, again spoken in ANT terms, is that the very process of CVD introduces many instances of simplification,

transformation and reconfiguration. It presents the “Obligatory Passage Point”, the chokepoint every vulnerability to be acknowledged as such has to traverse. At least in the context of this thesis, this obligatory passage point may shift if we take other mitigation strategies or practices in to account.

6.3 Software Vulnerabilities in CVD

Generally speaking, the main finding of this thesis is that the CVD is ultimately a “stabilization” model where software vulnerabilities become a “stabilized” item, an actor in its own right. Through many different minuscule steps, vulnerabilities are *made* (actionable) *in* and *during* the CVD and present a thing which is “ontologically multiple”:

- carrying with them the possibilities of being a mere “fault” in programs to deal with
- being “knowledge” or
- being “non-knowledge” (if some know about them, but others do not, they are “known unknowns”)
- being a “weapon” (as a spying tool, as a tool for disrupting supply chains or technical devices)
- being a “commodity” (enabling and rendering vulnerability markets)
- being a “point of pride/fame” (for researchers in Hall of Fames or generally bragging rights)
- being a “token of exchange” (in social configurations: [in]-formal [industry/government] meetings)
- being a “token of professionalism” (CV of researchers, companies handling them “responsible”)
- being a “liability” (for companies in regards to trust of their customer base).
- being the reason for companies to establish new teams, department to deal with themselves
- being the reason to mobilize resources (monetary, time, energy,...)

The main finding in the theoretical aspect, ANT, therefore presents itself as the CVD making visible and traceable the ontological multiplicity of software vulnerabilities. Only after the construction of this entity does it become possible to have a “closed” item to deal with.

6.4 CVD as “Matter of Care”

The CVD, through being a site of mitigation of harmful technologies and being an “established” practice, becomes a maintenance process. It is not only a “one-time” event taking place but is an implemented process in social institutions. Therefore, we can say that it is not only a “matter of concern”, something to be dissected and examined, but also is a “matter of care”, something which asks us how we deal with them. The CVD is a “political” encounter with the world in the sense that

vulnerabilities themselves and vulnerability disclosure is political. The potential to “weaponize” them as one ontological reading merits a positioning, even “non-positioning” in the spectrum of disclosure vs. non-disclosure becomes a political decision.

In this regard, the CVD opens up the question of how to maintain technologies and what worlds to disregard, as the (managerial) decision made always have a cost of opportunity. Either they end up in the (constant) need for mobilization of resources or the decision is made to do anything else than fixing potential (or at least as so perceived by the researcher/reporter) security issues.

Taken together, the CVD is a move towards the stabilization of objects. Because of the potentials of software vulnerabilities, one should “care” about them.

Software vulnerabilities are something to make visible and relevant, they pose a “threat” against something. Therefore, they present something to secure, to act upon. Ending up being an aspect of risk management processes, CVD is something which is institutionalized, formalized, needs continuous engagement and resources.

The CVD is a “matter of care” and a translation of material aspects (potential of software vulnerabilities) to a social thing (being made actionable), resulting in the mobilization of resources (money, time, people, computer, measurement devices, ideas, papers, reports, hierarchies,…) which form social structures.

6.5 Further Research

The scope of a Master’s thesis is necessarily limited, just doesn’t allow for a thorough investigation of a topic. I wished to include a lot more aspects to this topic and potentially speak to more people about it. However, this thesis should be a mere glimpse of how social worlds are constructed and made, how structures are being formed, stabilized and maintained. In doing this research, I encountered a lot of other potential pathways to further investigate. Some of which I will shortly lay out here:

One fascinating aspect I wished I could discuss more is the materiality of vulnerabilities, the very interaction between the physical sphere and the inscription of intentions through the practice of programming. A site to investigate this would be to take a closer look at the very basis of computer chips, the Random Access Memory (RAM). Here, the functioning of an electronic/physical/engineering level converges with (human) intention and mathematics. From an ANT perspective, this would be the site were the translation of (programming) language to material (bits and bytes) happens.

In this thesis I talked a lot about security, risk and harm. This stems from an interest in (digital) security. I chose this topic because of this disciplinary overlap. Specifically the overlap between security, (non-)knowledge and technologies I found to be fascinating:

> "Critical analyses of security have focused on the production of knowledge, techniques, and devices that tame unknowns and render social problems actionable." (Aradau, 2017, p. 327 [in Abstract])

As can be gathered from this quote, also the idea of making something *actionable* stems from this quote. The question I would like to introduce here to possibly investigate further would be how "(digital) materiality" informs aspects of security. Security for *whom*? What does it mean to talk about (cyber/IT) computer security? Doesn't that leave out the human security? The practices of securing computer networks, where CVD is just a fraction of, ultimately also forms our social world. I think there is still a lot to be investigated and learned. I hope to see a lot of such (case) studies in the future.

Thank you for caring.

Bibliography

Adobe. (n.d.). *Adobe Security Bulletin APSB21-51*. Retrieved September 1, 2021, from https://helpx.adobe.com/security/products/acrobat/apsb21-51.html

Akrich, M. (1992). The De-Scription of Technical Objects. In W. E. Bijker & J. Law (Eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (pp. 205–224). MIT Press.

Allodi, L., Banescu, S., Femmer, H., & Beckers, K. (2018). Identifying Relevant Information Cues for Vulnerability Assessment Using CVSS. *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, 119–126. https://doi.org/10.1145/3176258.3176340

Ambastha, M. (2019). *Taking a Hard Look at the Vulnerabilities Equities Process and its National Security Implications—Berkeley Technology Law Journal*. https://btlj.org/2019/04/taking-a-hard-look-at-the-vulnerable-equities-process-in-national-security/

Amoore, L. (2013). *The Politics of Possibility: Risk and Security beyond Probability*. Duke University Press.

Aradau, C. (2017). Assembling (Non)Knowledge: Security, Law, and Surveillance in a Digital World. *International Political Sociology*, *11*(4), 327–342.

Bastian, M. (Ed.). (2017). *Participatory research in more-than-human worlds*. Routledge, Taylor & Francis Group.

Bennett, J. (2010). *Vibrant matter: A political ecology of things*. Duke University Press.

Bijker, W. E., & Law, J. (Eds.). (1992). *Shaping technology/building Society: Studies in Sociotechnical Change*. MIT Press.

Blanchette, J.-F. (2011). A material history of bits. *Journal of the American Society for Information Science and Technology*, *62*(6), 1042–1057. https://doi.org/10.1002/asi.21542

Bloor, D. (1991). *Knowledge and Social Imagery* (2nd ed). University of Chicago Press.

Booth, H., Rike, D., & Witte, G. (2013). *ITL BULLETIN FOR DECEMBER 2013 THE NATIONAL VULNERABILITY DATABASE (NVD): OVERVIEW*. 3.

Bugcrowd. (n.d.). *Defensive Vulnerability Pricing Model. How to budget for your crowdsourced security program.* https://www.bugcrowd.com/resources/guides/bugcrowds-defensive-vulnerability-pricing-model/

Callon, M. (1984). Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of St Brieuc Bay. *The Sociological Review*, *32*(1_suppl), 196–233. https://doi.org/10.1111/j.1467-954X.1984.tb00113.x

Callon, M., & Latour, B. (1981). Unscrewing the big Leviathan: How actors macro-structure reality and how sociologists help them to do so. In K. Knorr-Cetina & A. V. Cicourel (Eds.), *Advances in social theory and methodology: Toward an integration of micro- and macro-sociologies*. http://site.ebrary.com/id/10913328

Caravelli, J., & Jones, N. (2019). *Cyber security: Threats and responses for government and business*. Praeger, an Imprint of ABC-CLIO, LLC.

Computer Fraud Abuse Act, 1001 18 USC (1986).

Corbin, J., & Strauss, A. (1990). Grounded Theory Research: Procedures, Canons, and Evaluative Criteria. *Qualitative Sociology*, *13*(1), 3–21.

Couture, S. (2019). The Ambiguous Boundaries of Computer Source Code and Some of Its Political Consequences. In J. Vertesi & D. Ribes (Eds.), *DigitalSTS: a field guide for science & technology studies* (pp. 136–156). Princeton University Press.

Daras, N. J. (Ed.). (2019). *Cyber-security and Information Warfare*. Nova Science Publishers, Inc.

DeHondt, G. (2019). Ethics and Policy of IoT. In R. Hammons & R. Kovac (Eds.), *Fundamentals of Internet of Things for Non-engineers.* (pp. 135–144). CRC Press, Taylor & Francis Group.

Dickow, M., Hansel, M., & Mutschler, M. M. (2015). Präventive Rüstungskontrolle – Möglichkeiten und Grenzen mit Blick auf die Digitalisierung und Automatisierung des Krieges. *Sicherheit & Frieden*, *33*(2), 67–73. https://doi.org/10.5771/0175-274x-2015-2-67

Dickson, E. (2020). *High-Tech Sex Toy Sales Rising During COVID-19 Pandemic*. Rolling Stone. https://www.rollingstone.com/culture/culture-news/teledildonics-remote-sex-toy-sales-covid19-coronavirus-pandemic-975140/

Digital Millennium Copyright Act, 18 (1998).

DoJ. (2017). *A Framework for a Vulnerability Disclosure Program for Online Systems*. https://www.justice.gov/criminal-ccips/page/file/983996/download

EC. (2017). *Resilience, Deterrence and Defence Building strong cybersecurity for the EU*. https://www.consilium.europa.eu/media/21479/resilience_deterrence_defence_cyber-security_ec.pdf

EC. (2020). *EU Security Union Strategy*. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52020DC0605&from=EN

Elazari Bar On, A. (2019). Private Ordering Shaping Cybersecurity Policy. In V. Mohan & R. Ellis (Eds.), *Rewired. Cybersecurity Governance* (First edition, pp. 231–264). Wiley.

ENISA. (2018). *Economics of vulnerability disclosure.* Publications Office. https://data.europa.eu/doi/10.2824/49807

ENISA. (2019a). *Advancing software security in the EU: The role of the EU cybersecurity certification framework.* Publications Office. https://data.europa.eu/doi/10.2824/81950

ENISA. (2019b). *Good practices for Security of IoT: Secure Software Development Lifecycle*. Publications Office. https://data.europa.eu/doi/10.2824/742784

ENISA. (2020). *Proactive Detection—Survey Results*. https://www.enisa.europa.eu/publications/proactive-detection-survey-results/at_download/fullReport

ENISA & RAND Europe. (2015). *Good practice guide on vulnerability disclosure: From challenges to recommendations*. Publications Office. https://data.europa.eu/doi/10.2824/610384

ETSI. (2021). *CYBER; Guide to Coordinated Vulnerability Disclosure*. https://docbox.etsi.org/cyber/CYBER/Open/Latest_Drafts/cyber-0062v005_Guide%20to%20Coordinated%20Vulnerability%20Disclosure.pdf

Finifter, M., Akhawe, D., & Wagner, D. (2002). *An Empirical Study of Vulnerability Rewards Programs* (USENIX Association, Ed.; pp. 273–288). USENIX Association.

FIRST. (n.d.). *Common Vulnerability Scoring System version 3.1 Specificiation Document Rev1*.

FIRST. (2020). *Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure*. https://www.first.org/global/sigs/vulnerability-coordination/multiparty/FIRST-Multiparty-Vulnerability-Coordination-latest.pdf?20180424.pdf

Fleck, L., Trenn, T. J., Merton, R. K., Bradley, F., & Fleck, L. (2008). *Genesis and Development of a Scientific Fact* (Repr. 11. Aufl). Univ. of Chicago Press.

Flick, U. (2009). *An Introduction to Qualitative Research* (4th ed). Sage Publications.

Foudil, E., & Shafranovich, Y. (2021). *A File Format to Aid in Security Vulnerability Disclosure draft-foudil-securitytxt-12*. https://datatracker.ietf.org/doc/html/draft-foudil-securitytxt-01

Galov, N. (2021, August 24). *How Many IoT Devices Are There in 2021? [All You Need To Know]*. https://techjury.net/blog/how-many-iot-devices-are-there/

Geer, D. (2014). *Cybersecurity as Realpolitik*. http://geer.tinho.net/geer.blackhat.6viii14.txt

Geers, K. & NATO Cooperative Cyber Defence Centre of Excellence. (2011). *Strategic cyber security*. NATO Cooperative Cyber Defence Centre of Excellence.

Gerlitz, C., & Weltevrede, E. (2020). What happens to ANT, and its emphasis on the socio-material grounding of the social, in digital sociology? In I. Farias, C. Roberts, & A. Blok (Eds.), *The Routledge Companion to Actor-Network Theory* (pp. 345–356). Routledge, Taylor & Francis Group.

Gieryn, T. F. (1983). Boundary-Work and the Demarcation of Science from Non-Science: Strains and Interests in Professional Ideologies of Scientists. *American Sociological Review*, *48*(6), 781–795. https://doi.org/10.2307/2095325

Greenberg, A. (2019). *Sandworm: A new era of cyberwar and the hunt for the Kremlin's most dangerous hackers* (First edition). Doubleday.

HackerOne. (2020). *The 4th Hacker Powered Security Report*. https://www.hackerone.com/resources/reporting/the-2020-hacker-report

HackerOne. (2021). *The 2021 Hacker Report*. https://www.hackerone.com/resources/reporting/the-2021-hacker-report?utm_source=website&utm_medium=homepage&utm_campaign=2021_hacker_report_spota-2021-03-02_06-30&utm_content=spota-2021-03-02_06-30

Haraway, D. J. (1985). *A Cyborg Manifesto*. University of Minnesota Press. https://doi.org/10.5749/minnesota/9780816650477.001.0001

High Level Group of Scientific Advisors. (2017). *Cybersecurity in the European Digital Single Market*.

House, N. (2016). *The Complete Cyber Security Course. Volume I. Hackers Exposed.* StationX.

Householder, A. D. (2015, June 7). *Like Nailing Jelly to the Wall Difficulties in Defining "Zero-Day Exploit."* https://insights.sei.cmu.edu/blog/like-nailing-jelly-to-the-wall-difficulties-in-defining-zero-day-exploit/

Householder, A. D., Wassermann, G., Manion, A., & King, C. (2017). *The CERT Guide to Coordinated Vulnerability Disclosure*.

ISC2. (2020). *Global Information Security Workforce Study*. https://www.isc2.org/Research/-/media/0AAF29023217474EB5D0D76170A75ABB.ashx

ISO/IEC. (2018). *ISO/IEC 29147 (2018) Information technology—Security techniques—Vulnerability disclosure*.

Jensen, E. A., & Laurie, A. C. (2016). *Doing Real Research: A Practical Guide to Social Research*. SAGE.

Johnson. (1988). Mixing Humans and Nonhumans Together: The Sociology of a Door-Closer. *Social Problems*, *35*(3), 298–310.

Johnson, P., Gorton, D., Lagerström, R., & Ekstedt, M. (2016). Time between vulnerability disclosures: A measure of software product vulnerability. *Computers & Security*, *62*, 278–295. https://doi.org/10.1016/j.cose.2016.08.004

Joint Task Force Interagency Working Group. (2020). *Security and Privacy Controls for Information Systems and Organizations* (Revision 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5

Jones, N. (2019). Innovation as a Driver of Cyber Security. In J. Caravelli & N. Jones (Eds.), *Cyber security: Threats and responses for government and business* (pp. 175–199). Praeger, an Imprint of ABC-CLIO, LLC.

Kilovaty, I. (2017). Freedom to Hack. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3035518

Klimburg-Witjes, N., & Wentland, A. (2021). Hacking Humans? Social Engineering and the Construction of the "Deficient User" in Cybersecurity Discourses. *Science, Technology, & Human Values*, 016224392199284. https://doi.org/10.1177/0162243921992844

Knorr-Cetina, K. (1999). *Epistemic cultures: How the sciences make knowledge*. Harvard University Press.

Kovac, R. J. (2019). IoT Positioning in the Verticals. In R. L. Hammons & R. J. Kovac (Eds.), *Fundamentals of Internet of Things for Non-Engineers* (1st ed., pp. 35–61). Auerbach Publications. https://doi.org/10.1201/9780429274992-1

Kovacs, E. (2018, January 26). *Maersk Reinstalled 50,000 Computers After NotPetya Attack*. https://www.securityweek.com/maersk-reinstalled-50000-computers-after-notpetya-attack

Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*. The University of Chicago Press.

Laszka, A., Zhao, M., & Grossklags, J. (2016). Banishing Misaligned Incentives for Validating Reports in Bug-Bounty Platforms. In I. Askoxylakis, S. Ioannidis, S. Katsikas, & C. Meadows (Eds.), *Computer Security – ESORICS 2016* (Vol. 9879, pp. 161–178). Springer International Publishing. https://doi.org/10.1007/978-3-319-45741-3_9

Latour, B. (1992). Where are the Missing Masses? The Sociology of a Few Mundane Artifacts. In W. E. Bijker & J. Law (Eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (pp. 225–258). MIT Press.

Latour, B. (2004). Why Has Critique Run out of Steam? From Matters of Fact to Matters of Concern. *Critical Inquiry*, *30*(2), 225–248. https://doi.org/10.1086/421123

Latour, B. (2005). *Reassembling the social: An introduction to actor-network-theory*. Oxford University Press.

Latour, B., & Woolgar, S. (1986). *Laboratory Life: The Construction of Scientific Facts*. Princeton University Press.

Law, J. (1999). After Ant: Complexity, Naming and Topology. *The Sociological Review*, *47*(1_suppl), 1–14. https://doi.org/10.1111/j.1467-954X.1999.tb03479.x

Lewis, P. S. (2017). *The Global Vulnerability Discovery and Disclosure System: A Thematic System Dynamics Approach*. Cranfield University.

Libicki, M. C., Ablon, L., & Webb, T. (2015). *The Defender's Dilemma: Charting a Course toward Cybersecurity*. RAND.

Matwyshyn, A. M., Ang Cui, Keromytis, A. D., & Stolfo, S. J. (2010). Ethics in security vulnerability research. *IEEE Security & Privacy Magazine*, *8*(2), 67–72. https://doi.org/10.1109/MSP.2010.67

Michael, M. (2016). *Actor network theory: Trials, trails and translations* (1st edition). SAGE Ltd.

MIITNS. (2021). *Regulations on the Management of Network Product Security Vulnerability*. [website in Chinese, translation done by using https://translate.yandex.com/translator/Chinese- English]. http://www.cac.gov.cn/2021-07/13/c_1627761607640342.htm

MITRE. (n.d.-a). *CVE - Terminology*. Retrieved September 1, 2021, from https://cve.mitre.org/about/terminology.html#cve_id

MITRE. (n.d.-b). *CWE-416 Use After Free (4.5)*. Retrieved September 1, 2021, from https://cwe.mitre.org/data/definitions/416.html

MITRE. (n.d.-c). *CWE-About-CWE Overview*. Retrieved September 1, 2021, from https://cwe.mitre.org/about/index.html

Mol, A. (2002). *The Body Multiple: Ontology in Medical Practice*. Duke University Press.

Moore, T., Friedman, A., & Procaccia, A. D. (2010). Would a "cyber warrior" protect us: Exploring trade-offs between attack and defense of information systems. *Proceedings of the 2010 Workshop on New Security Paradigms - NSPW '10*, 85. https://doi.org/10.1145/1900546.1900559

Mutlu, C. E. (2012). The material turn. Introduction. In M. B. Salter & C. E. Mutlu (Eds.), *Research Methods in Critical Security Studies: An Introduction* (pp. 173–180). Routledge.

NIST NVD. (n.d.-a). *CVE-2021-35983 Detail*. Retrieved September 1, 2021, from https://nvd.nist.gov/vuln/detail/CVE-2021-35983

NIST NVD. (n.d.-b). *Vulnerabilities*. Retrieved August 24, 2021, from https://nvd.nist.gov/vuln

Offensive Security. (n.d.). *PWK and OSCP Frequently Asked Questions*. https://www.offensive-security.com/offsec/pwk-oscp-faq/#what

Ozment, A., & Schechter, S. E. (2006). Milk or Wine: Does Software Security Improve with Age? ∗. *15th USENIX Security Symposium (USENIX Security '06)*, 12. https://www.usenix.org/legacy/event/sec06/tech/full_papers/ozment/ozment.pdf

Pala, A., & Zhuang, J. (2019). Information Sharing in Cybersecurity: A Review. *Decision Analysis*, *16*(3), 172–196. https://doi.org/10.1287/deca.2018.0387

Pickering, A. (1995). *The Mangle of Practice: Time, Agency, and Science*. Univ. of Chicago Press.

Pinch, T. J., & Bijker, W. E. (1984). The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology might Benefit Each Other. *Social Studies of Science*, *14*(3), 399–441. https://doi.org/10.1177/030631284014003004

Prasad, R., & Rohokale, V. (2020). *Cyber Security: The Lifeline of Information and Communication Technology*. Springer International Publishing. https://doi.org/10.1007/978-3-030-31703-4

Puig de la Bellacasa, M. (2017). *Matters of Care: Speculative Ethics in More Than Human Worlds*. University of Minnesota Press.

Ribes, D. (2019). Materiality Methodology, and Some Tricks of the Trade in the Study of Data and Specimens. In J. Vertesi & D. Ribes (Eds.), *DigitalSTS: a field guide for science & technology studies* (pp. 43–60). Princeton University Press.

Ruefle, R. (2007). *Defining Computer Security Incident Response Teams*. https://us-cert.cisa.gov/bsi/articles/best-practices/incident-management/defining-computer-security-incident-response-teams

Ruppert, E., Law, J., & Savage, M. (2013). Reassembling Social Science Methods: The Challenge of Digital Devices. *Theory, Culture & Society*, *30*(4), 22–46. https://doi.org/10.1177/0263276413484941

Schaake, M., Pupillo, L. M., Ferreira, A. H. B., Varisco, G., & Centre for European Policy Studies. (2018). *Software vulnerability disclosure in Europe: Technology, policies and legal challenges : report of a CEPS Task Force*. https://www.ceps.eu/publications/software-vulnerability-disclosure-europe-technology-policies-and-legal-challenges

Scheier, B. (2014, May 19). *Should U.S. Hackers Fix Cybersecurity Holes or Exploit Them*. https://www.theatlantic.com/technology/archive/2014/05/should-hackers-fix-cybersecurity-holes-or-exploit-them/371197/

Schulze, M., & Reinhold, T. (2018). Wannacry About the Tragedy of the Commons? Game-Theory and the Failure of Global Vulnerability Disclosure. *Proceedings of the 17th European Conference on Cyber Warfare and Security. University of Oslo. Norway. 28-29 June 2018*, 454–463.

Seaver, N. (2017). Algorithms as culture: Some tactics for the ethnography of algorithmic systems. *Big Data & Society*, *4*(2), 205395171773810. https://doi.org/10.1177/2053951717738104

Shah, N. (2012). The Internet as evocative infrastructure. In M. B. Salter & C. E. Mutlu (Eds.), *Research Methods in Critical Security Studies. An Introduction.* (pp. 186–190). Routledge.

Stamatia, S. (2019). Ethics in Cyberspace. In N. J. Daras (Ed.), *Cyber-security and Information Warfare* (pp. 325–336). Nova Science Publishers, Inc.

Stehr, N. (2017). Knowing and Not Knowing. In P. Meusburger, B. Werlen, & L. Suarsana (Eds.), *Knowledge and Action* (Vol. 9, pp. 113–125). Springer International Publishing. https://doi.org/10.1007/978-3-319-44588-5_7

Storm, B., Applebaum, A., Miller, D., Nickels, K., Pennington, A., & Thomas, C. (2020). *MITRE ATT&CK: Design and Philosophy*. https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf

The Economist. (2017a, April 8). *The Myth of Cyber-Security*. https://businessmirror.com.ph/2017/04/17/the-myth-of-cyber-security/

The Economist. (2017b, April 8). *Why everything is hackable. Computer security is broken from top to bottom*. https://www.economist.com/science-and-technology/2017/04/08/computer-security-is-broken-from-top-to-bottom

US CERT. (n.d.). *US-CERT: United States Computer Emergency Readiness Team*. https://us-cert.cisa.gov/sites/default/files/publications/infosheet_US-CERT_v2.pdf

U.S. Department of Homeland Security. (2020). *Binding Operational Directive 20-01*. https://cyber.dhs.gov/assets/report/bod-20-01.pdf

Votipka, D., Stevens, R., Redmiles, E., Hu, J., & Mazurek, M. (2018). Hackers vs. Testers: A Comparison of Software Vulnerability Discovery Processes. *2018 IEEE Symposium on Security and Privacy (SP)*, 374–391. https://doi.org/10.1109/SP.2018.00003

Weulen Kranenbarg, M., Holt, T. J., & van der Ham, J. (2018). Don't shoot the messenger! A criminological and computer science perspective on coordinated vulnerability disclosure. *Crime Science*, *7*(16), 1–9. https://doi.org/10.1186/s40163-018-0090-8

White House. (2017). *Vulnerabilities Equities Policy and Process for the USG*. https://www.hsdl.org/?view&did=805726

Winner, L. (1980). Do Artifacts Have Politics? *Daedalus*, *109*(1), 121–136.

World Economic Forum. (2018, January 24). *Securing a Common Future in Cyberspace*. https://www.youtube.com/watch?v=Tqe3K3D7TnI&t=322s

Woszczynski, A., Green, A., Dodson, K., & Easton, P. (2020). Zombies, Sirens, and Lady Gaga – Oh My! Developing a Framework for Coordinated Vulnerability Disclosure for U.S. Emergency Alert Systems. *Government Information Quarterly*, *37*(1), 1–15. https://doi.org/10.1016/j.giq.2019.101418

Zhao, M., Grossklags, J., & Liu, P. (2015). An Empirical Study of Web Vulnerability Discovery Ecosystems. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1105–1117. https://doi.org/10.1145/2810103.2813704

Appendix

Abstract

(english)

Software and devices that use or rely on software are ubiquitous. With this software come vulnerabilities. These vulnerabilities are the basis for far reaching consequences, being responsible for a lot of phenomena from mundane computer crashes to malicious activity. This activity can be anything from cybercrime to state-sponsored hacking or the disruption of (global) supply chains. Therefore, vulnerabilities are also deemed relevant to notions of security. To limit possible harmful usage of software vulnerabilities, there are a lot of diverse mitigation strategies. One of them is the Coordinated Vulnerability Disclosure process. In this process, vulnerabilities are reported without the prospect of a reward to the party deemed responsible to fix this vulnerability, that is, to roll out a patch.

Before they can be reported, however, they first have to be made actionable. Actor-Network Theory serves as the theoretical lens applied in this thesis. This theoretical framework is located in the realm of Science-Technology-Studies, an academic field concerned with the interactions between technology and society. The primary problem is the understanding of vulnerabilities as a tangible object, an entity of its own. Since vulnerabilities are located not in the physical realm but in a digital sphere, they do not have certain properties or materials which provide the fundamental basis to understand them in the same sense as ANT's standard analytical model introduces. Therefore, the focus is on the parts that make up a vulnerability, to dissect it in a way it can be understood as an object which then insights from ANT can be applied.

ANT investigates different modes of ordering in the world around us. Which boundaries are drawn, what concepts are applied and what makes up the things we take for granted. In this regard, the journey of a vulnerability starts with the people searching for them. The methods applied are document analysis as well as semi-structured interviews. The interviews are conducted with people searching for vulnerabilities "in the wild", in software already deployed, as well as with people holding the position of Chief Information Security Officer.

This thesis investigates what knowledge goes into software vulnerability research, how the reporting process is organized and how the examination process of reported vulnerabilities looks like. In doing so, the goal is to gain a deeper understanding of how the social world is constructed through minuscule interactions as well as gaining insights into what negotiations are performed in the CVD.

The results of this thesis suggest that the boundaries of what is considered a vulnerability are quite diffuse and that they are formed as well as uphold by doing (social) work. Also, the "ontological status" of what vulnerabilities are, is found to be formed only in relation to outer influences and spheres, they are therefore contestable. It is understood as an assemblage. Lastly, the notion of "matters of care" is invoked, understanding the concept of "security" as a continuous effort and the constant mobilization of resources to deal with potentially adverse technologies such as software vulnerabilities.

Abstract

(deutsch)

Software und Produkte, die Software enthalten, sind überall zu finden. In dieser Software sind Schwachstellen, auch genannt Sicherheitslücken. Diese Sicherheitslücken sind die Ursache vieler weitreichender Konsequenzen. Sie sind verantwortlich für eine Vielzahl an Phänomenen, von banalen Dingen wie Computerabstürzen bis hin zu böswilligen Handlungen. Das Handlungsspektrum hier erstreckt sich von Internetkriminalität über „Hackerangriffe“ im Namen von Staaten bis hin zur Unterbrechung oder Störung von (globalen) Lieferketten.
In diesem Sinne stellen Softwareschwachstellen auch ein Sicherheitsproblem dar. Die Eindämmung von böswilligen Effekten oder Handlungen verlangt daher nach einer Vielzahl von Überlegungen. Eine dieser Strategien ist die Offenlegung der Schwachstellen, in der Fachsprache „Coordinated Vulnerability Disclosure“ (auch: Responsible Disclosure) genannt. Dieser Prozess ist gekennzeichnet durch die Mitteilung von Sicherheitslücken an die jeweils als Verursacher identifizierten Parteien mit dem Ziel einer Behebung derselben.
Bevor die Sicherheitslücken allerdings behoben werden können, müssen diese praktikabel oder aufbereitet werden. Akteur-Netzwerk Theorie ist die theoretische Basis dieser Masterarbeit. Diese Theorie ist Bestandteil der Science-Technology-Studies, was wiederum der Wissenschaftssoziologie anhänglich ist. Dieses akademische Feld erforscht das Spannungsfeld zwischen Technologie(n) und sozialen Strukturen, Institutionen und Verhältnissen.
Die primäre Problemstellung ist das Verständnis von Sicherheitslücken als greifbare Objekte, ein „Ding an sich“. Diese Sicherheitslücken sind in diesem Verständnis nicht eindeutig dem „Physischen“ zuordenbar, da sie sich in einem digitalem Raum bewegen. Durch die bisherig vorrangig Behandlung physischer Objekte verschließt sich die Akteur-Netzwerk Theorie dem Untersuchen von digitalen Objekten. Deshalb müssen als erster Schritt die Begrifflichkeiten der Akteur-Netzwerk Theorie dem digitalen Objekt „Sicherheitslücken“ angepasst werden, um die theoretischen Annahmen dann auf diese „Sache“ anzuwenden.
Das Grundverfahren von Akteur-Netzwerk Theorie ist verschiedene Formen von Ordnungsmäßigkeiten oder -prinzipien zu verfolgen. Welche Arten von Grenzen oder Barrieren werden von wem, wie und wo gezogen und aufgebaut, wie werden diese argumentiert und wie formen diese Tätigkeiten unsere Verständnisse der Welt?
In dieser Hinsicht beginnt die Reise einer Sicherheitslücke bei den Personen, die danach suchen oder diese entdecken. Die Methoden dieser Arbeit umfassen Dokumentenauswertung und die Durchführung von teilstandardisierten Interviews. Diese Interviews wurden mit Leuten durchgeführt, die Sicherheitslücken „in freier Wildbahn“, d.h. in Software oder -produkten die bereits in Anwendung sind, suchen. Eine zweite Gruppe umfasst Personen mit der Berufsbezeichnung „Chief Information Security Officer“, einer Managementposition die sich um die Informationssicherheit in Betrieben kümmert.
Diese Arbeit untersucht, welche Arten von Wissen in die Suche nach Sicherheitslücken fließt, wie das Meldeverfahren sowie die Verifizierung und Einstufung derselben organisiert sind. Das Ziel ist, ein besseres Verständnis über die Erschaffung von „sozialen Welten“ durch als unbedeutend wahrgenommene Praktiken zu bekommen. Auch die Aushandlung was als Sicherheitslücke gilt ist Teil dieser Arbeit.
Die Resultate dieser Arbeit zeigen, dass die Grenzen von was als „Sicherheitslücke“ gilt fließend sind und dass das Verständnis derselben (sozialer) Arbeit bedarf. Der „ontologische Status“ was das „Ding“ Sicherheitslücke ist, ist einzig erklärbar durch die Verknüpfung mit äußerlichen Einflüssen oder Sphären, das Verständnis ist daher verhandelbar. Sie sind ein Knotenpunkt. Als letzter Punkt werden Sorgfalt und Achtsamkeit als Kategorien eingeführt. Diese sollen dabei helfen, „Sicherheit“ als eine fortlaufende Anstrengung durch die Mobilisierung von Ressourcen mit dem Ziel potenziell schadhafter oder schädlicher Technologien, wie zum Beispiel Sicherheitslücken, zu begreifen.