Tilburg Law School

Tilburg Institute for Law, Technology and Society

Walid El Hammouti

(SNR: u1235701)

THE USE OF SOFTWARE AND HARDWARE VULNERABILITIES BY LAW ENFORCEMENT

UPHOLDING INTEGRITY IN INVESTIGATIONS

Master Thesis L.L.M. Law & Technology

Supervisors: Dr. E. Kosta & Dr. A. Martin

December 2019

Table of contents

1 Introduction

1.1 Background

Vulnerabilities are often described as flaws or weaknesses in the code of a software system or application and have been around for as long as computer systems have existed. To gain access to computer systems, attackers for decades have been taking advantage of vulnerabilities (i.e. exploiting them) leading to huge financial losses for businesses and individuals, a loss of access to vital applications and systems and also the disclosure of confidential and personal data.[1] In an ever so digitalised society, attackers have become better than ever at finding flaws in software and mobile devices up until the point where technology companies have taken a stance in protecting individuals by upping the security of their technology.[2] Devices and communications are now increasingly, as it is called, encrypted by default or from end-to-end, meaning that in practice it has become a lot more difficult to gain access to a certain device. Whilst encouraged by many, the flipside of this development is that it has also become more difficult for law enforcement to uncover information relating to criminal cases as authorities face the same level of high security. In essence, there is no simple on or off switch that allows for access by only authorities and intelligence agencies whilst still ensuring integrity and privacy of data and systems despite some interesting attempts.[3]

Over the years, law enforcement therefore has been exploring other possibilities to (re)gain access to devices or servers to fight what former FBI director Comey has dubbed as "law enforcement going dark".[4] Since then, law enforcement agencies have gradually been introducing themselves to the hacking arena as they too now are increasingly trying to gain access to devices (of criminals) through vulnerabilities. Even more so, exploiting vulnerabilities has become one of the most important methods for law enforcement to obtain information that otherwise would not have been accessible for the purpose of solving a criminal

[1] Cat Rutter Pooley, 'Cyber security efforts turn proactive after sophisticated attacks' (*Financial Times*, 15 November 2018) < https://www.ft.com/content/68a9398a-d065-11e8-9a3c-5d5eac8f1ab4> accessed 9 June 2019.
[2] CB Insights, 'How Big Tech Is Finally Tackling Cybersecurity' (*CB Insights*, 27 March 2019) < https://www.cbinsights.com/research/facebook-amazon-microsoft-google-apple-cybersecurity/> accessed 9 June 2019.
[3] Bruce Schneier, 'Ray Ozzie's Encryption Backdoor' (*Schneier on Security*, 7 May 2018) <https://www.schneier.com/blog/archives/2018/05/ray_ozzies_encr.html> accessed 9 June 2019.
[4] James B. Comey, 'Going Dark: Are Technology, Privacy, and Public Safety on a Collision Course?' (*FBI*, 16 October 2014) <https://www.fbi.gov/news/speeches/going-dark-are-technology-privacy-and-public-safety-on-a-collision-course> accessed 9 June 2019.

case.[5] One of the first larger cases of law enforcement exploiting vulnerabilities dates back to 2013 when the FBI, under "Operation Torpedo", seized a website hosting service (Freedom Hosting) that operated on the Tor Network.[6] Freedom Hosting was targeted by the FBI because it allowed for child pornography to be hosted on its servers. Numbers showed that at the time 95 percent of the hidden child pornography existed on these servers).[7] The FBI was able to expose these servers onto which this child pornography was hosted by installing an executable (instructions that cause a computer to perform a certain task) that looked up the MAC address (a unique number that is tied to a specific piece of hardware similar to a serial number) and hostnames of the computers that accessed specific – child pornography - websites hosted by Freedom Hosting.[8] This information was consequently forwarded to a server which then revealed to the FBI the real IP addresses of these users. Fundamental in this case was that the FBI was able to install the malware specifically because it was able to take advantage of a security weakness in the web browser Firefox which was not known to Mozilla (the developers behind Firefox) at that time, meaning that in that period no patch had been developed to address this vulnerability (i.e. a 0-day). As anonymous Tor browsing activities generally run over Firefox, the visitors of these child pornography websites hosted by Freedom Hosting were all using Firefox exposing them to this weakness which the FBI utilised to gather information and evidence on these suspects.[9]

Even though the vulnerability that was utilised by the FBI wasn't exploited by other attackers, the nature of vulnerabilities does allow for such a situation as a vulnerability that exists for law enforcement simultaneously exists for everyone who discovers it. Other parties upon finding or creating the same vulnerability could use it then to steal passwords, use the machine in a DDoS botnet or, as the Stuxnet cyberattack shows, even allow for access to nuclear centrifuges. Stuxnet was a so-called computer worm that exploited multiple Windows vulnerabilities in order to spread and infect other computers.[10] More specifically, it was developed to target "centrifuges which were used to produce enriched uranium that powers

---

[5] Kristin Finklea, 'Law Enforcement Using and Disclosing Technology Vulnerabilities' (2017) (Congressional Research Service R44827 <https://fas.org/sgp/crs/misc/R44827.pdf> accessed 9 June 2019.
[6] Zach Lerner, 'A Warrant to Hack: An Analysis of the Proposed Amendments to Rule 41 of the Federal Rules of Criminal Procedure' (2016) 18 Yale J.L. & Tech 1, p. 64.
[7] Kevin Poulsen, 'Feds Are Suspects In New Malware That Attacks TOR Anonymity' (*WIRED*, 8 May 2013) <https://www.wired.com/2013/08/freedom-hosting/> accessed 9 June 2019.
[8] Ibid.
[9] Zach Lerner, 'A Warrant to Hack: An Analysis of the Proposed Amendments to Rule 41 of the Federal Rules of Criminal Procedure' (2016) 18 Yale J.L. & Tech 1, p. 64.
[10] Josh Fruhlinger, 'What is Stuxnet, who created it and how does it work?' (*CSO*, 22 August 2017) <https://www.csoonline.com/article/3218104/what-is-stuxnet-who-created-it-and-how-does-it-work.html> accessed 9 June 2019.

nuclear weapons and reactors".[11] The Stuxnet attack is a clear example of how exploitation of vulnerabilities isn't just about personal computers and mobile devices that can be targeted to steal passwords or to discover IP addresses, but notably can also serve as "cyber weapons" targeting critical infrastructure.[12]

This idea unfortunately is not a theoretical one. Recently, the NSA lost an exploit ("EternalBlue") that it had developed which utilised a vulnerability in the Microsoft Server Message Block (SMB).[13] This vulnerability – dubbed BlueKeep – along with its exploit fell into the hands of the Shadow Brokers hacking collective which showcased what a mass attack on security might look like.[14] Since 2017, the exploit has been used by several state hackers in countries like North Korea, Russia and China and has caused billions of dollars in damage by "paralysing the British health care system, German railroads and some 200,000 organisations around the world".[15] The attack - famously called "WannaCry" for its disruptive impact - shut down thousands of computers worldwide and reports show that even though the vulnerability in the Microsoft software has been patched by the vendor, there is still a high number of computers that are at risk because many system administrators and individuals still haven't installed it.[16]

Today's risks and consequences of vulnerabilities in software which can be exploited, don't leave much to the imagination and if anything, questions could be raised as to the safeguards that exist for law enforcement when exploiting vulnerabilities. Mainly, there is much uncertain about what measures exist to prevent the spreading of the effects of using vulnerabilities beyond the targeted suspects. [17] In this regard, much of the current discourse revolves around questions relating to whether or not law enforcement should be allowed to exploit vulnerabilities, because of the potential risks this inherently brings. Given that over the

---

[11] Ibid.

[12] Nicole Perlroth & Scott Shane, 'In Baltimore and Beyond, a Stolen N.S.A. Tool Wreaks Havoc (*NY Times*, 25 May 2019) <https://www.nytimes.com/2019/05/25/us/nsa-hacking-tool-baltimore.html> accessed 9 June 2019.

[13] CERT-EU, 'WannaCry Ransomware Campaign Exploiting SMB Vulnerability' (2017) CERT-EU Security Advisory 2017-012 <https://cert.europa.eu/static/SecurityAdvisories/2017/CERT-EU-SA2017-012.pdf> accessed 20 November 2019.

[14] Dawn Kawamoto, "'WannaCry' Rapidly Moving Ransomware Attack Spreads to 74 Countries" (*Dark Reading*, 5 December 2017) <https://www.darkreading.com/attacks-breaches/wannacry-rapidly-moving-ransomware-attack-spreads-to-74-countries/d/d-id/1328874> accessed 20 November 2019.

[15] Nicole Perlroth & Scott Shane, 'In Baltimore and Beyond, a Stolen N.S.A. Tool Wreaks Havoc (*NY Times*, 25 May 2019) <https://www.nytimes.com/2019/05/25/us/nsa-hacking-tool-baltimore.html> accessed 9 June 2019.

[16] Zack Whittaker, 'Two years after WannaCry, a million computers remain at risk' (*TechCrunch*, 12 May 2019) < https://techcrunch.com/2019/05/12/wannacry-two-years-on/> accessed 9 June 2019.

[17] Thomas P. Bossert, "It's official: North-Korea Is Behind WannaCry" (*WSJ*, 18 December 2017) <https://www.wsj.com/articles/its-official-north-korea-is-behind-wannacry-1513642537> accessed 20 November 2019.

last years, globally, these risks have manifested in some severe real life effects with damaged computer systems and data breaches stemming from attacks like WannaCry and the Baltimore ransomware attack, not much doubt remains to exist about what can happen when law enforcement is not careful in utilising flaws in computer systems. However, there is one other aspect that is interesting in the course of this subject which is the question what can be said about the data or evidence that is gathered when there are no sufficient safeguards or measures to preserve the integrity of such data.

Inherent to the exploitation of vulnerabilities as a means to gather evidence is that integrity is not always preserved, especially in light of other more conventional investigative techniques. From the perspective of a defendant, gaining knowledge about how police entered a certain device and comprehending what type of vulnerability was utilised, might prove to be fundamental when it comes to the question whether other illegitimate attackers had access to the same device and whether the evidence that was collected has been tampered with. Not disclosing all the details about a certain exploit that is used has already led to the withdrawal of certain cases in the United States. Although the Dutch court system does not require all information about investigative techniques to be disclosed in each and every case, this does touch upon a noteworthy clash of interests: on the one hand guaranteeing integrity of evidence gathered through computer exploitation and on the other hand secrecy of investigative techniques.

This research does not intend to fully explore the issue of disclosing vulnerabilities or whether Dutch police should be allowed to exploit vulnerabilities, but its purpose is rather to examine the importance of (technical) measures and safeguards for preserving integrity of investigations and the role these play in that process.

1.2 Research- and sub-questions

As government agencies increasingly resort to exploiting vulnerabilities to gain access to systems of suspects in order to obtain evidence and the grave risks that come along with this power, there are several questions that can be raised relating to the current framework and how these measures stack up to provide an adequate baseline that establishes that any data and evidence that are collected remain uncompromised. To examine this, the research question for this thesis will be:

Why are (technical) safeguards and measures important for law enforcement' exploitation of vulnerabilities to preserve the integrity of gathered evidence?

To answer this research question, the following sub-questions will be examined:

- In what way do hardware and software allow for vulnerabilities to come into existence?
- Under which circumstances and how is law enforcement allowed to exploit vulnerabilities?
- What controls exist to prevent proliferation and preserve the integrity of investigations?

1.3 Overview of chapters

Chapter 2 will describe the technical mechanisms behind vulnerabilities and their exploitation. How is it that these flaws always exist in systems and software and how do they allow for access? To come to a good understanding of how vulnerabilities over the years have been exploited by bad actors it is also fundamental to understand the difference between vulnerabilities that are publicly known (at least to the vendors) and on the other hand so-called zero-day vulnerabilities (hereinafter 0-day). These 0-days are vulnerabilities not known to vendors and the public, and therefore can cause harm as in these cases there is no patch available to resolve these flaws. Also of great importance to the issue at hand, is to assess which ways exist to otherwise obtain a vulnerability or exploit if no flaw can be found by an attacker itself. Finally, to illustrate the importance and effects of vulnerability exploitation means to properly examine how some of the most common vulnerabilities are utilised in practice.

Chapter 3 will describe how law enforcement agencies can obtain a warrant to exploit a vulnerability that is discovered or obtained and more importantly whether there are cases in which perhaps authorisation is not required. How then do these agencies operate in practice, what mechanisms and tools do they have at hand? Furthermore, this chapter dives into all the cases in which it is allowed to exploit a vulnerability and explores how the safeguards instated by law actually manifest in practice. These safeguards will furthermore be examined in light of risks that come with vulnerability exploitation like misuse and proliferation.

In chapter 4, several controls will be discussed that contribute to maintaining the integrity of digital evidence. The law often sets outs different procedural measures (e.g. a warrant is generally only allowed for a certain period of time), however, as exploiting vulnerabilities in this regard causes significant difficulties because of its inherent nature, it could be desirable to

discuss some of the more technical controls that can be instated to guarantee a fair investigation as these can lead to a more strict enforcement of safeguards. This chapter therefore touches upon some of the measures that can be implemented in this regard and how legally enforcing them hugely benefits law enforcement agencies in their investigations when vulnerabilities have been used to gather evidence. Finally, it will illustrate what the consequences of the omission of such measures can be in relation to the integrity of evidence.

1.4 Significance and aim

When law enforcement obtains a vulnerability there are basically two paths it can take. Firstly, it can disclose the vulnerability to the vendor allowing it to patch up the weakness thereby preventing anyone, including police agencies, who know of the vulnerability and want to exploit it, the ability to do so. Or, as opposed to disclosing it, law enforcement can keep a vulnerability to themselves and use it as an offensive weapon to "gather intelligence, help execute search warrants or deliver malware".[18] The purpose of this research is to shine a light on the legal process behind making use of vulnerabilities or exploits and to examine what controls exist and should be implemented when it comes to utilising vulnerabilities. Exploiting vulnerabilities does not only impact personal devices and data, but also (critical) infrastructure making it massively important that the right checks and balances are implemented to prevent any proliferation.

The aim is to be descriptive in illuminating this development. Not only to lay out how law enforcement handling of vulnerability exploits can actually come about, but also to examine whether proper measures and efforts are established in the current legal framework that prevent the spread of exploits to any third (unlawful) party. Important in this regard is to not only look at previous law enforcement exploits and accompanying effects, but also to take into consideration future implications and what vulnerability exploitation in this context will do for solving criminal cases (hence the emphasis on integrity of evidence). Furthermore, the dichotomy 'privacy vs. security' in this perspective seems to have shifted to 'security vs. security' meaning that technical aspects and developments are also critical for understanding the issue at hand.

This thesis contributes to existing literature by focussing on the controls that exist and by examining how the current Dutch legal framework works out in practice in light of issues

[18] Bruce Schneier, 'WannaCry and Vulnerabilities' (*Schneier on Security*, 2 June 2016) <https://www.schneier.com/blog/archives/2017/06/wannacry_and_vu.html> accessed 9 June 2019.

like misuse and proliferation. Little is written about law enforcement guarantees- and accountability in this context. By examining control and oversight, this thesis will primarily address transparency and clarify whether law enforcement's use of vulnerabilities comes with enough consideration. Research into judicial and procedural practices in this context is necessary and welcoming as many of the discussions revolve around an analysis of the technical tools and the broader notion of hacking, and less so on several evidentiary questions that are raised along with exploitation.

Finally, by examining some cybersecurity attacks that were the result of government agencies losing possession over their exploits, this research intends to explore whether these losses can be characterised as the result of a slippery slope, as law enforcement increasingly being a target of hacking or as the result of an incentive that is created by law enforcement hunting for vulnerabilities.

1.5 Methodology

This thesis will be based on a traditional literature research. It will primarily focus on primary sources including legislation, regulations, (lawful) hacking handbooks and reports that have been written up by non-profit organisations. Examining the legal framework in this research is one of the most important methods to come to the answers to the research questions. This will be accompanied by researching policy and legal research reports on government hacking in Europe and the United States to develop an adequate frame for understanding the issue at hand and might deliver some insights on how to address certain issues this research identifies.

As law enforcement in many instances did and does not report on the exploits it has developed, discovered or used, case law and secondary sources like online news media and security expert/researcher blogs will also be studied. A descriptive analysis of the latest developments in the context of recent cyberattacks in which law enforcement exploitation played a key role will be fundamental towards explaining law enforcement accountability and will offer a springboard to answer the research question.

Next, this thesis will require a multi-disciplinary approach as its subject stands at the intersection of several disciplines and not only considers the technical mechanisms behind exploiting vulnerabilities, but also requires legal doctrine to critically assess the warrant-mechanics that accompany the technical use of exploits when utilising vulnerabilities and evidence handling coming out of these practices. This doctrinal analysis is fundamental in order to be able to explain in detail the risks that come with exploiting vulnerabilities by authorities

and even more so offers insights to what other alternatives exist offering a balanced evaluation of the current situation and can suggest how other controls might benefit law enforcement.

2 Computer system structures and vulnerabilities

2.1 Computer layers

A technical vulnerability is often described as a weakness or flaw in a computer system or software which can be manipulated by a malicious entity allowing it to gain access and to cause harm.[19] Such a flaw can be the result of not adequately implemented, and perhaps even absent, security controls and procedures such as not changing a default password (at its simplest), but can also stem out of errors or mistakes in the code that make up a software application or system. To understand how vulnerabilities can come into existence and can be utilised, it is fundamental to look at the relationship between computer systems and vulnerabilities and how these are structured. Accordingly, mobile systems nowadays are designed in a similar fashion and thus require no separate explanation in this regard.[20]

As computer systems are often described in terms of "layers" that communicate with each other, this chapter will first discuss the two layers that play the most significant role in vulnerability exploitation, namely hardware and software. [21] Secondly, the process of finding vulnerabilities will be explored to give an overall impression of how attackers and researchers can go about finding vulnerabilities that allow for exploitation. Finally, as the notion of vulnerabilities is fairly broad, the final part of this chapter will give an overview of some of the most common vulnerabilities allowing for a better understanding of what a weakness in a computer system actually comes down to in practice.

2.1.1 Hardware

At the lowest layer of a computer system sits the hardware layer.[22] This layer exists of the tangible components and architecture that physically make up a computer system including essential parts like the central processing unit (CPU), hard drives and network interface cards that are contained in and on a system. These different hardware components are driven and controlled by smaller software programs (i.e. microprograms) and as they do, vulnerabilities

[19] Arthur Conklin, Gregory White, *Principles of Computer Security: CompTIA Security+ and Beyond* (McGraw-Hill Education 2018), p. 686.
[20] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 26.
[21] Ibid.
[22] Ibid.

can come into existence through mistakes in the code that make up these programs.[23] Hardware vulnerabilities mostly exist because hardware design is difficult which allows more opportunities for mistakes to be made.[24] The technical specifications of a certain hardware component like the memory (i.e. RAM) of a computer might match perfectly with the overall design of a certain system on paper, however, in practice this might turn out to be the opposite.[25] This also has to do with the fact that many manufacturers that produce chips for instance, don't necessarily design them from scratch but use different elements and processes that come from other third parties.[26] This then can lead to a vulnerability as the components that come from these different parties can interact with each other in a way unforeseen at the time of building the system.[27]

Consequently, the aforementioned issues often result in it being fairly expensive to develop a patch for vulnerabilities that are found in one or more of the hardware components.[28] To complicate things further, a lack of expertise in terms of knowledge and the ability to repair hardware vulnerabilities characterises the many difficulties related to developing an adequate fix.[29] Most troubling in this regard, however, is that in the cases where there are enough resources and experts to iron out a flaw, many companies run into the issue of how fixing a certain hardware vulnerability might lead to interoperability issues with the other hardware contained in a certain system.[30]

---

[23] Paulo Garcia, 'Don't trust your hardware: Why security vulnerabilities affect us al' (*The Conversation*, 1 November 2018) <https://theconversation.com/dont-trust-your-hardware-why-security-vulnerabilities-affect-us-all-105773> accessed 25 June 2019.
[24] Jordi Mongay Batalla, George Mastorakis, Constandinos X. Mavromoustakis, Evangelos Pallis, *Beyond the Internet of Things: Everything Connected* (Springer 2017), p. 72.
[25] Paulo Garcia, 'Don't trust your hardware: Why security vulnerabilities affect us al' (*The Conversation*, 1 November 2018) <https://theconversation.com/dont-trust-your-hardware-why-security-vulnerabilities-affect-us-all-105773> accessed 25 June 2019.
[26] Gedare Bloom, Eugen Leontie, Bhagirath Narahari and Rahul Simha, *Handbook on Securing Cyber-Physical Critical Infrastructure* (Morgan Kaufmann 2012), p. 306.
[27] Paulo Garcia, 'Don't trust your hardware: Why security vulnerabilities affect us al' (*The Conversation*, 1 November 2018) <https://theconversation.com/dont-trust-your-hardware-why-security-vulnerabilities-affect-us-all-105773> accessed 25 June 2019.
[28] Bill Horne, 'Hardware Security: Why Fixing Meltdown & Spectre Is So Tough' (*Dark Reading* 26 January 2018) <https://www.darkreading.com/risk/hardware-security-why-fixing-meltdown-and-spectre-is-so-tough/a/d-id/1330908> accessed 25 June 2019.
[29] Jordi Mongay Batalla, George Mastorakis, Constandinos X. Mavromoustakis, Evangelos Pallis, *Beyond the Internet of Things: Everything Connected* (Springer 2017), p. 72.
[30] Robert Donovan, 'Are Some Security Vulnerabilities Too Complex to Fix?' (*InfoSecurity Magazine*, 28 May 2019) <https://www.infosecurity-magazine.com/infosec/security-vulnerabilities-1-1-1-1/> accessed 25 June 2019.
[30] Joseph M. Kizza, *Guide to Computer Network Security* (Springer 2015), p. 161.

2.1.1.1 Spectre, Meltdown and Checkm8

This all makes that hardware vulnerabilities often have long lasting, more severe effects which can be illustrated by examining three noteworthy vulnerabilities. The first two are often regarded as one vulnerability as they were discovered at the same time and in the same type of hardware, namely the processor.[31] Spectre and Meltdown – as the flaws are often referred to - shook up the industry back in 2017 when they were exposed by security researchers as these did not only exist in machines manufactured around that time, but affected every computer chip that had been manufactured over the last 20 years.[32] Even though there have been no reports of any exploitation of these vulnerabilities in practice and thus did not have any real world consequences, they did potentially give rise to an attacker gaining access to data and have usernames and passwords revealed.[33] More characteristic, however, was that as these two vulnerabilities were hardware related, it wasn't possible for a patch to directly be developed (i.e. due to the complexity that comes with hardware vulnerabilities). The eventual solution had to come from vendors like Microsoft, Google and Apple in the form of software patches that worked around the problem that allowed for these vulnerabilities to exist.[34] Despite this solution, it is striking that some researchers have argued that it is likely that these vulnerabilities will never be able to be fixed completely.[35]

Where the Spectre and Meltdown weaknesses have been patched and it could be said that the effects these two vulnerabilities could cause are mitigated, the current state of technology and expertise still does not prevent vulnerabilities to be found that will reside in hardware for an infinite amount of time without it being possible to develop a fix. In this regard, the recently discovered "checkm8" vulnerability which was found by an iOS security researcher in the Apple iPhone bootrom, showcases that in some cases a "workaround patch" won't be able to repair the weaknesses found, meaning that the vulnerability will reside in the

---

[31] Andy Greenberg, 'Triple Meltdown: How So Many Researchers Found a 20-Year-Old Chip Flaw at the Same Time (*Wired,* 1 July 2018) <https://www.wired.com/story/meltdown-spectre-bug-collision-intel-chip-flaw-discovery/> accessed 24 November 2019.

[32] Josh Fruhlinger, 'Spectre and Meltdown explained: What they are, how they work, what's at risk' (*CSO* 15 January 2018) <https://www.csoonline.com/article/3247868/spectre-and-meltdown-explained-what-they-are-how-they-work-whats-at-risk.html> accessed 25 June 2019.

[33] Andy Greenberg, 'Meltdown Redux: Intel Flaw Lets Hackers Siphon Secrets from Millions of PCs' (*WIRED*, 14 May 2019) <https://www.wired.com/story/intel-mds-attack-speculative-execution-buffer/> accessed 25 June 2019.

[34] Josh Fruhlinger, 'Spectre and Meltdown explained: What they are, how they work, what's at risk' (*CSO* 15 January 2018) <https://www.csoonline.com/article/3247868/spectre-and-meltdown-explained-what-they-are-how-they-work-whats-at-risk.html> accessed 25 June 2019.

[35] The New York Times Editorial Staff, *Hacking and Data Privacy: How Exposed Are We?* (The Rosen Publishing Group 2018), p. 196.

piece hardware permanently and can only be mitigated by replacing the actual component.[36] Characteristic to the long-lasting impact again is that even though these flaws were discovered in the fall of 2019, every iPhone model that is manufactured from 2013 until 2017 was affected by this vulnerability.[37]

Nevertheless, as impactful as hardware vulnerabilities might be, they are generally also significantly difficult to exploit.[38] Often detailed knowledge of for instance CPU architecture is required to be able to discover, let alone exploit, a certain weakness in such a physical component.[39] As not many people have this specific knowledge, hardware vulnerabilities aren't encountered as often. Generally, attackers therefore try to acquire knowledge about software vulnerabilities which have more immediate implications.[40]

2.1.2 Software

2.1.2.1 System software

Software is typically classified in two different categories that allow for vulnerabilities in separate ways, namely system- and application software.[41] What is understood under system software is the software that allows a computer system to function. Put differently, it is the software that makes it possible to operate the hardware in and on a computer system and acts as an infrastructure onto which software applications can be developed and used.[42] It generally comprises of the operating system, utility software (software applications that perform standard tasks like editing files and assist in the maintenance of a system) and the language processor (allows for the translation of machine language into a so-called high-level language which is understandable by humans).[43] Particularly important in the context of vulnerabilities is the

---

[36] Thomas Reed, 'New iOS exploit checkm8 allows permanent compromise of iPhones' (*Malwarebytes*, 27 September 2019) <https://blog.malwarebytes.com/mac/2019/09/new-ios-exploit-checkm8-allows-permanent-compromise-of-iphones/> accessed 24 November 2019.
[37] Axi0mX, 'ipwndfu' (*Github*, 1 October 2019) <https://github.com/axi0mX/ipwndfu> accessed 24 November 2019.
[38] Apostolos P. Fournaris, Lidia P. Fraile, Odysseas. Koufopavlou, 'Exploiting Hardware Vulnerabilities to Attack Embedded System Devices: a Survey of Potent Microarchitectural Attacks' (2017) 6 (52) Electronics <https://www.mdpi.com/2079-9292/6/3/52/pdf> accessed 23 November 2019.
[39] Inside Battelle, 'Hardware vs. Software Vulnerabilities' (*Inside Battelle*, 18 January 2018) <https://inside.battelle.org/blog-details/hardware-vs.-software-vulnerabilities> accessed 25 June 2019.
[40] Ibid.
[41] Joseph M. Kizza, *Guide to Computer Network Security* (Springer 2015), p. 162.
[42] Ciprian Rusen, *IC3: Internet and Computing Core Certification Computing Fundamentals Study* (Sybex 2015), p. 4.
[43] Nancy Sehgal, *Let's Log In: A Textbook for Introductory Information Technology* (Dorling Kindersly (India) Pvt. Ltd. 2006), p. 32.

operating system which can be described as a layer of software that manages all communications between the hardware components, the software applications used and the user.[44] Any interaction with a certain system ranging from managing the file system on a machine, taking input from a keyboard and mouse or actually allowing a user to use the hardware components for their purpose is possible due to the operating system.[45]

At the core of the operating system sits the kernel which, first and foremost, performs some common tasks that applications require such as power management and memory allocation (i.e. assigning memory to software applications that are being used).[46] More specifically, the kernel is the element in a computer system that actually allows the machine to communicate with external hardware such as the network and thus can be seen as a central piece that ties the software and hardware components together.[47] When a certain software application like a word editor wants to save information on a hard disk, it is the kernel that in essence makes this possible. In order to do so, the kernel enforces so-called "file permissions" meaning that it specifies which files are owned by which users, who can read and write certain files and finally which users can execute these files.[48] Consequently, when an application wants to copy, edit or load in a certain file, the kernel checks whether the application (i.e. user) has the right permissions for these actions and upon confirmation will perform, or allow, for the action to take place. Arguably, this makes the kernel one of the most important targets for attackers. If one would be able to change these file permissions so as to become the owner of a certain file (i.e. read, write, execute files etc.), this basically amounts to a compromise of the whole system as he or she would be able to do everything with and on the system.[49] As it is such a fundamental element, the kernel therefore protects itself by what is called separation from any of the other running software applications (often called "separation at the software level").[50] What this means in practice is that the code that the kernel runs to perform a certain action is run with "full privileges" or in other words, it basically can perform any action

---

[44] Ciprian Rusen, *IC3: Internet and Computing Core Certification Computing Fundamentals Study* (Sybex 2015), p. 4.
[45] Ibid.
[46] Gordon Haff, 'Why the operating system matters even more in 2017' (*opensource*, 7 December 2016) <https://opensource.com/16/12/yearbook-why-operating-system-matters> accessed 27 June 2019.
[47] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 26.
[48] Bruce Duyshart, *The Digital Document: A Reference for Architects, Engineers and Design Professionals* (Routledge 2013), p. 62.
[49] Zhen Yan, Refik Molva, Wojciech Mazurczyk, Raimo Kantola, *Network and System Security: 11th International Conference, NSS 2017, Helsinki, Finland, August 21–23, 2017, Proceedings (Lecture Notes in Computer Science)* (Springer 2017), p. 233.
[50] Enrico Perla, Massimiliano Oldani, *A Guide to Kernel Exploitation: Attacking the Core* (Syngress 2010), p. 4.

possible (i.e. has access to all components in a system). All the other code that is run outside the operating system, also referred to as code run in user-land, is subject to several limitations and in essence is not able to access all components and carry out every desirable action.[51] Fundamentally, vulnerability exploitation when it comes to the operating system mainly is about gaining access to various hardware components by owning a certain process allowing one to maliciously impact other, more vital, processes.[52]

2.1.2.2 Application software

As discussed in the previous section, the kernel protects itself by separation at the kernel level which makes that it is well protected and much more complex to exploit directly. Especially when considering that in order to be able to utilise a vulnerability at this level, data will need to be sent and received which then has to be captured.[53] The kernel in this regard carries out almost no processing of data packets thus making capturing data from and to the kernel almost impossible.[54] Therefore, arguably, most penetrations in the case of software do not take place in system software, or at least do not start there, but in software programs.[55] Application software concerns the software that most users are familiar with and use on a day-to-day basis. Applications like web browsers, e-mail clients, word editors, graphic editing applications and so on all exist on top of the operating system and generally allow a user to carry out a certain operation for a specific purpose (as opposed to operating system tasks which manage the computer system).[56] The ways in which attackers gain access through vulnerabilities in software programs is often through malware on webpages (e.g. through XSS vulnerabilities as discussed in section 2.3.4), users downloading malicious applications that contain different vulnerabilities which can then be utilised by an attacker and "poor implementation of network protocols".[57] Regardless, the outcome is the same as the main purpose of these different mechanisms is to have a user run a certain program that contains weaknesses with the user's file permissions, allowing for access to a certain part of a system and arguably a compromise

---

[51] Ibid.
[52] Trent Jaeger, *Operating System Security* (Morgan and Claypool Publishers 2008), p. 1.
[53] Jidong Xiao, Hai Huang, Haining Wang, *Security and Privacy in Communication Networks: 11th EAI International Conference, SecureComm 2015, Dallas, TX, USA, October 26-29, 2015, Proceedings* (Springer International Publishing 2015), p. 135-136.
[54] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 27.
[55] Ibid.
[56] ITL Education Solutions Limited, *Introduction to Computer Science* (Pearson Education India 2004), p. 296.
[57] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 27.

of the whole computer (directly or indirectly, in the case of the latter a second exploit is needed that allows for so-called "privilege escalation").[58] It is beyond this research's scope to explore the subject of privilege escalation, however, essential to understand for this thesis is that in order to be able to exploit a certain vulnerability on a software level, an attacker can attack the kernel directly (which is most complex) or a software application will need to be penetrated after which "more rights are appropriated" allowing an attacker to do everything with the targeted system.[59] In sum, exploiting software application vulnerabilities is all about trying to gain system privileges as this allows an attacker to modify applications and drivers.

2.2 Vulnerability lifecycle

2.2.1 Technical reconnaissance

Before vulnerabilities can actually be exploited, attackers start with what is called a reconnaissance phase. This period of studying and discovering the target system is necessary in order to be able to install or create an exploit and more so because these exploits must be precisely tailored to the target system in terms of the type of operating system (Windows, macOS or Linux), the exact version, patch levels and so on.[60] To do so, one of the first steps that attackers take is to look at information that is publicly available, so information that can be found by anyone on the internet. A so-called "DNS" and "Whois" lookup are two resources that offer information as to the internet domain and IP address that are used by a certain target.[61] The use of OSINT (Open Source Intelligence) tools allow an attacker to scour search engines and social media platforms for information about the "identity" of the victim to get a more general overview of what devices and services the victim uses.[62]

2.2.2 Mapping and enumeration

Once this information is gathered an attacker will try to obtain more specific information which brings an attacker to the next phases of "enumeration" and "mapping" of the target system or network. In other words, after first gathering more general information about the target, information will be gathered concerning actual usernames, specific versions of software

[58] Ibid.
[59] Ibid, p. 28.
[60] Ibid, p. 38.
[61] Garth O. Bruen, *WHOIS Running the Internet: Protocol, Policy, and Privacy* (Wiley 2015), p. 113.
[62] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 39.

applications and operating systems, open ports, vulnerable services that the victim runs and so on in order to perform a more intrusive and active check on the system or network.[63] This information is then used for vulnerability mapping meaning that the specific components and use of the target system by the victim will be crosschecked for any known vulnerabilities.[64] For instance, upon discovering that the victim machine uses port 23 (over which the Telnet service runs), an attacker can search through certain vulnerability databases (e.g. NIST Vulnerability Database) and check whether a vulnerability exists for this specific service.[65] In the case of this example, the attacker would have found that there is indeed a vulnerability known in Telnet and can now write an exploit to gain access to the system remotely (i.e. in the case no prior exploit has been developed).[66]

2.2.3 Finding and exploiting vulnerabilities

This process of crosschecking the found information with any known vulnerabilities can thus be done manually, but an attacker can also scan a target system using certain services which automate the processes described in the previous section.[67] For this purpose, many attackers turn to so-called automated vulnerability scanners (e.g. Nessus) or other frameworks which are used not only by malicious entities, but also by legitimate users in order to find out how vulnerable they are to attacks.[68] These vulnerability scanners operate by sending (i.e. pinging) certain data to a computer system forcing it to return certain information about it which it then crosschecks in vulnerability databases allowing it to output what weaknesses exist in a machine.[69] However, many vulnerabilities and the way in which these can be exploited are also readily available in so-called pre-packaged scripts.[70] The "Metasploit" framework in this regard is arguably the most popular resource for attackers as it hosts the largest database of these scripted publicly available exploits (also called modules).[71] Put differently, Metasploit

---

[63] WebIMX, 'Penetration Testing' (*WebIMX*) <http://www.webinfomatrix.com/penetration-testing.html> accessed 5 July 2019.
[64] Gerard Johansen, Lee Allen, Tedi Heriyanto, Shakeel Ali, *Kali Linux 2 – Assuring Security by Penetration Testing* (Packt Publishing 2016), p. 64.
[65] Daniel Miessler, 'Public Vulnerability Database Resources' (*Daniel Miessler*, 5 December 2018) <https://danielmiessler.com/study/vulnerability-database-resources/> accessed 3 July 2019.
[66] Microsoft, 'MS09-042: Vulnerability in Telnet could allow remote code execution' (*Microsoft*, 17 April 2018) <https://support.microsoft.com/en-us/help/960859/ms09-042-vulnerability-in-telnet-could-allow-remote-code-execution> accessed 24 November 2019.
[67] Sean Oriyano, *Penetration Testing Essentials* (Sybex 2016), p. 122.
[68] Ibid.
[69] Harold F. Tipton, Micki K. Nozaki, *Information Security Management Handbook* (CRC Press 2007), p. 2831.
[70] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 40.
[71] Ibid.

offers not only information about known vulnerabilities, but simultaneously through the same framework allows for direct exploitation of the vulnerabilities it has identified.[72]

2.2.4 Zero-day vulnerabilities

Aside from these vulnerabilities that are available to the public, there are also vulnerabilities which are kept secret and aren't known to anyone (or at least, have not been disclosed). These vulnerabilities are called zero-days (hereinafter: 0-days). More specifically, 0-days are vulnerabilities that have not been disclosed to a software vendor (i.e. the vendor is not aware of these vulnerabilities) and thus no patch for them has yet been developed.[73] This makes these vulnerabilities especially interesting (and pricey) to attackers as this lack of knowledge about their existence makes everyone using the software application vulnerable to exploitation. There has been increasing policy debate about the use of 0-days by law enforcement and intelligence services in particular, regarding the question of whether law enforcement should or should not disclose these vulnerabilities to the appropriate software vendors so that a patch can be developed.[74] Knowledge about 0-days is not exclusive to law enforcement meaning that other attackers that discover the same weakness and likewise develop an exploit to utilise that weakness, will be able to execute malware or some other attack to collect private information like usernames and passwords of individual users.[75] Thus, there are basically two competing interests at stake in this debate, namely on the one hand law enforcement needs to gather intelligence to solve crime for which purpose the private retention of 0-days is essential and arguably necessary.[76] On the other hand, these same weaknesses expose not only innocent individuals but also governments and other institutions to attacks from criminals and increasingly other intelligence agencies and hacking groups (under the notion of advanced persistent threats) therefore suggesting that law enforcement should disclose these vulnerabilities to the vendors so that these can be patched as soon as possible.[77] Still, there is not much research on the true extent of 0-days and many experts in the field have argued that most vulnerabilities that are exploited involve not these undisclosed vulnerability, but actually

---

[72] Sagar Rahalkar, *Metasploit for Beginners: Create a threat-free environment with the best-in-class tool* (Packt Publishing 2017), p. 43.
[73] Lillian Ablon, Martin C. Libicki, Andrea A. Golay, *Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar* (Santa Monica, CA: RAND Corporation, 2014), p. 143-152.
[74] Lilly Ablon, Andy Bogart, *Zero Days, Thousands of Nights, The Life and Times of Zero-Day Vulnerabilities and Their Exploits* (RAND Corporation 2017), preface.
[75] Ibid.
[76] Matt Blaze, 'When Should the Government Disclose "Stockpiled" Vulnerabilities?' (*Matt Blaze*, 2017) <https://www.mattblaze.org/blog/between_immediately_and_never/> accessed 15 July 2019.
[77] Ibid.

vulnerabilities that have already been disclosed and for which even a patch already can exist.[78] The NSA reported that over the years, 91% of the vulnerabilities that it has discovered have been disclosed or known to the appropriate vendors.[79] Still, disclosed or not, 0-days remain the most impactful and worthwhile vulnerabilities for many attackers as their impact generally is more severe (demonstrated by the Heartbleed and Stuxnet attacks that exploited 0-days).[80]

2.2.5 Creating vulnerabilities

In the case a certain system that is targeted exhibits neither an already publicly known vulnerability nor a 0-day, an attacker will need to set out to find new vulnerabilities. This process involves creating crashes in the code that makes up a software application which then are researched.[81] This process of examining code for some error to be triggered is called "code auditing" and can be done on two different levels.[82] Firstly, one can run an automated fuzzer on a software application. Fuzz testing, or simply fuzzing, is a technique whereby "random, invalid or unexpected data is sent as inputs into a software application with the purpose of uncovering unexpected and undesired behaviour".[83] So, at its simplest, a fuzz test will send completely random input strings (i.e. text) to the interface of a computer program after which one just awaits what the output will be, desirably an error.[84] Finally, in the event an application crashes, attackers and researchers go about reworking the crash to determine the root cause in the program and this information is then used to determine whether the bug is actually a vulnerability (for instance because it results it read, write and execution rights) and consequently what type of vulnerability.[85] Important to stipulate for the purpose of this research is that this method of "creating vulnerabilities" generally is not carried out by law enforcement or other attackers, but more often so by security researchers at universities to study software application and hardware behaviour.

---

[78] Michael Sulmeyer, Kate Miller, 'Indicting Hackers and Known Vulnerabilities' (*Lawfare*, 27 May 2016) <https://www.lawfareblog.com/indicting-hackers-and-known-vulnerabilities> accessed 15 July 2019.
[79] NSA, 'Discovering IT Problems, Developing Solutions, Sharing Expertise' (*NSA*, 30 October 2015) <https://www.nsa.gov/News-Features/News-Stories/Article-View/Article/1625787/infographic-discovering-it-problems-developing-solutions-sharing-expertise/> accessed 20 July 2019.
[80] Lilly Ablon, Andy Bogart, *Zero Days, Thousands of Nights, The Life and Times of Zero-Day Vulnerabilities and Their Exploits* (RAND Corporation 2017), p. 3.
[81] Ibid, p. 66.
[82] Ibid.
[83] Arthur Conklin, Daniel Shoemaker, *CSSLP Certification All-in-One Exam Guide* (McGraw-Hill Education 2013), p. 170.
[84] Ibid.
[85] Lilly Ablon, Andy Bogart, *Zero Days, Thousands of Nights, The Life and Times of Zero-Day Vulnerabilities and Their Exploits* (RAND Corporation 2017), p. 67.

2.2.6 Purchasing vulnerabilities on the market

Perhaps the easiest way to obtain a vulnerability is to acquire one on the market. Though this might sound as simple as purchasing software or hardware at a vendor, it often isn't about convenience or simplicity but more so about necessity as most systems nowadays are better secured.[86] Vulnerabilities and exploits have become harder to find and utilise due to improved security, but also because software vendors have become quicker at releasing security patches to fix reported and known weaknesses, especially in the case of critical vulnerabilities.[87] Accordingly, in some cases an attacker needs to operate under certain time pressure, for instance when law enforcement needs immediate access to the system of a criminal suspect. When in time-sensitive cases there is no vulnerability and exploit "on the shelf" time will not allow police to go through all the different phases described in the previous sections to discover a new vulnerability and write a new exploit, but what it will do in these cases is check whether a certain vulnerability can be purchased on the open market.[88]

When it comes to the vulnerabilities market, just as with goods, there is an overt and a black market for those who are in the business or in need of specific vulnerabilities. The overt market generally exists in the form of bug bounty programs that are offered by different hardware and software vendors and simply come down to a monetary reward (in most cases) being offered to those (researchers) who find weaknesses in a software product.[89] These programs or incentives to report original vulnerabilities aren't reserved for the private sector as governments also increasingly pay for vulnerability information particularly in the case of 0-days. In the infamous San Bernardino case, the FBI reportedly paid for a vulnerability which finally granted it access to an iPhone after months of trying to force access through the vendor.[90] There are also open exploit markets such as Zero Day Initiative (ZDI) and iDefense that operate under a different model whereby a vulnerability is disclosed to a vendor for free serving as a marketing tool for their security services.[91]

---

[86] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 40.
[87] Bernardo Lustosa, "Apple's iOS update frequency has increased 51% under Cook's management" (*VentureBeat*, 28 February 2018) <https://venturebeat.com/2018/02/28/apples-ios-update-frequency-has-increased-51-under-cooks-management/> accessed 24 November 2019.
[88] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 40.
[89] Ibid.
[90] Jack Nicas, Apple to Close iPhone Security Hole That Law Enforcement Uses to Crack Devices (*NY Times*, 13 June 2018) <https://www.nytimes.com/2018/06/13/technology/apple-iphone-police.html> accessed 20 July 2019.
[91] Alana Maurushat, *Disclosure of Security Vulnerabilities: Legal and Ethical Issues* (Springer; 2013), p. 12-13.

Whereas vulnerabilities that are placed on the overt market are primarily disclosed to allow a vendor to patch it up and improve the security of its product or service, the black market for vulnerabilities exists for one purpose only which is financial profit. Someone might discover a new vulnerability, contact the vendor or owner of the software product and demand a monetary reward in exchange for them to not publish or reveal the vulnerability and exploit, all to gain a financial benefit.[92] Deemed by many as unethical as this mostly resembles extortion, such a request or move in fact is not illegal.[93] The black market indiscriminately allows any group or organisation (ranging from cyber criminals to governments) to acquire vulnerabilities.[94] The price paid for vulnerabilities on the black market is said to be five to ten times the amount of vulnerabilities sold on the open market.[95]

Needless to say, the markets for vulnerabilities have expanded over the years and many security companies are now in the prime business of finding and developing vulnerabilities and exploits as a business model. Though not all security vendors disclose the figures behind their business, research has shown that prices can range from 20-250,000 dollars per vulnerability.[96] Some vendors even sell subscriptions for which hundreds of thousands of dollars will need to be paid.[97] Exclusive access to 0-days can generally be considered as the most expensive vulnerabilities out on both the overt and black market. Finally, as news reports over the years have suggested, national governments (i.e. intelligence and military agencies) have also become major buyers in this market.[98]

2.3 Vulnerability classes

This chapter up until this point has discussed how attackers can obtain vulnerabilities. After a vulnerability has been discovered or acquired attackers generally will try to gain access to systems. This section will explore how some of the most common vulnerabilities allow for exploitation to create an understanding of this process and to answer the question what it actually means to gain access to computer systems through a flaw. Based on the type of

---

[92] Ibid.
[93] Ibid.
[94] Abdullah M. Algarni, Yashwant K. Malaiya, 'Software Vulnerability Markets: Discoverers and Buyers' (2014) 8 International Journal of Computer, Information Science and Engineering 3, p. 75.
[95] Ibid.
[96] Andy Greenberg, 'Meet the Hackers Who Sell Spies The Tools To Crack Your PC (And Get Paid Six-Figure Fees)' (*Forbes*, 21 May 2012) <https://www.forbes.com/sites/andygreenberg/2012/03/21/meet-the-hackers-who-sell-spies-the-tools-to-crack-your-pc-and-get-paid-six-figure-fees/> accessed 20 July 2019.
[97] Nicole Perlroth & Scott Shane, 'In Baltimore and Beyond, a Stolen N.S.A. Tool Wreaks Havoc (*NY Times*, 25 May 2019) <https://www.nytimes.com/2019/05/25/us/nsa-hacking-tool-baltimore.html> accessed 20 June 2019.
[98] Ibid.

vulnerability, vulnerabilities are often categorised in classes. It should be noted that a vulnerability does not have to be directly exploitable for it to be considered a vulnerability.[99] Some flaws for instance weaken the security of a certain program but do not directly lead to access. They do however act as a springboard as they make other attacks possible (as discussed in section 1.1.2.2. which briefly touched upon the notion of privilege escalation). An example of this is a so-called information disclosure vulnerability that undesirably discloses certain information about an application that could be used to further exploit a device.[100] In the next sections, the most well-known classes or types of vulnerabilities (directly exploitable or not) will be described to give an overall impression of what a vulnerability actually allows for.

2.3.1 Buffer overflows

Buffer overflows (or memory corruption vulnerabilities) are the most common type of vulnerabilities and are often the starting point for attackers when trying to intrude a network.[101] In the case of a software application that requires or allows for certain data to be entered (e.g. a username and password), a programmer will need to specify the amount of data that is expected to be entered by a user. To make this possible, memory storage will be set aside (i.e. a buffer will be created to accommodate this data). For a chat application, developers might for instance create a 50-byte buffer meaning that they expect a user to not enter a username that contains more than 50 bytes. However, in the case a user enters a username that is 90 bytes which consequently is not checked by the application, the additional storage of 40 bytes that exceeds the 50-bytes buffer that was created by the program, may be written over different areas in the memory that is used by other applications.[102] Put simply, when a programmer fails to limit the amount of data that can be entered (i.e. written) in a "predefined buffer" this can lead to an overflow of this buffer and in the end create a memory corruption which could allow an attacker to tamper with other applications or cause the operating system to execute certain commands.[103]

---

[99] James Forshaw, *Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation* (No Starch Press 2017), p. 346.
[100] Ibid, p. 348.
[101] Ibid.
[102] Ibid.
[103] Jason Andress, *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice* (Syngress 2011), p. 192.

2.3.2 Race conditions

Race conditions are often described as programming conundrums. These vulnerabilities occur when two processes compete, i.e. race, to access the same resource before the other.[104] The infamous example of a bank withdrawal clearly illustrates what a race condition in practice looks like: when person A wants to withdraw an amount of €100 from an ATM, the processes that take place are checking what the account balance is, consequently withdraw this amount and then update the account balance with €100. However, if person B would withdraw an amount of €50 at the same time (i.e. start the same process), this could result in the situation where the process behind this withdrawal finishes about a second earlier leading to an incorrect account balance.[105] When it comes to race conditions, as the same resource is shared, the correct handling of that resource thus depends on the proper ordering or timing of these processes, in other words, the results of the actions that these processes are attempting to carry out will be different depending on the order in which they occur. To prevent the situation from the earlier example, in practice, banks will have implemented several measures to prevent this from happening, but this example illustrates how two processes, or users, "race" to access the same resource which can lead to a security vulnerability (e.g. privilege escalation or redirect information) if not correctly handled. [106]

2.3.3 Command (SQL) injections

Command injection attacks can occur when a programmer, or application, doesn't properly validate the input a user enters into a database.[107] An example is when a certain program prompts a user to enter a new name for a folder or directory that will need to be created and instead of entering a regular name like "Folder 1", the input that is given is "newdirectory&cmd".[108] This would result in the creation of a new directory and simultaneously open a command terminal which can then be used by an attacker to execute a different command as the input thus isn't checked properly (an application should not allow for an additional command to be accepted).[109] SQL injections then are a type of command

[104] Ibid.
[105] Ibid.
[106] James Graham, Ryan Olson, Richard Howard, *Cyber Security Essentials* (Routledge 2010), p. 148.
[107] Michael E. Whittman, Herbert J. Mattord, David Mackey, Andrew Green, *Guide to Network Security* (Cengage Learning 2012), p. 351.
[108] Ibid.
[109] Ibid.

injections which are among the most reported vulnerabilities.[110] Almost all applications need to store and retrieve data and one of the most common ways this is done is by using what is called a relational database. Relational databases offer the main advantage of issuing queries (i.e. search commands) and SQL (Structured Query Language) - the main programming language for managing data in relational databases - defines what data is to be read and how to filter that data to get the results the application, i.e. user, wants.[111] What happens however when an SQL injection vulnerability is exploited is that an attacker injects undesired data into an SQL query which then is executed on the database. This action results in the ability to retrieve sensitive user data or the ability to bypass security mechanisms.[112] SQL injection attacks are regarded as one of the top cyber-attacks on the internet as almost all websites that contain a databases use SQL as a programming language and it is this standardisation that makes hacking multiple databases much easier.[113] Noticeably, the largest security breach that exposed a huge amount of sensitive user information because of an SQL weakness is the Sony hack that took place in 2011.[114]

2.3.4 Cross-site scripting (XSS) vulnerabilities

Again related to the issue of not properly checking the input a user enters, are so-called cross-site scripting (XSS) vulnerabilities. These vulnerabilities can come into existence by carrying out an attack in which maliciously designed scripts are injected into trusted websites.[115] Put differently, when a website requires certain input from a user and this input isn't validated properly, an attacker can place malicious code into that website (for instance in the form of an outdated Adobe Flash advertisement).[116] When a user then visits and views this website, he or she – unintentionally - executes the code automatically thereby carrying out the attack. XSS attacks often target websites from banks or retailers in the form of a comment that is left by an attacker with a malicious script that is then executed when a user reads the comment thereby

---

[110] Giorgio Franceschetti, Marina Grossi, *Homeland Security Technology Challenges: From Sensing and Encrypting to Mining and Modeling* (Artech House Publishers 2008), p. 88.
[111] James Forshaw, *Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation* (No Starch Press 2017), p. 381.
[112] Lech J. Janczewski, Andrew M. Colarik, *Cyber Warfare and Cyber Terrorism* (Information Science Reference 2019), p. 161.
[113] Ibid.
[114] Rupert Goodwins, 'Sony hacked again in Lulzsec breach' (*ZDNet*, 3 June 2011) <https://www.zdnet.com/article/sony-hacked-again-in-lulzsec-breach/> accessed 25 July 2019.
[115] Michael E. Whittman, Herbert J. Mattord, David Mackey, Andrew Green, *Guide to Network Security* (Cengage Learning 2012), p. 351.
[116] Jason Andress, *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice* (Syngress 2011), p. 195.

executing it.[117] In short, XSS weaknesses are exploited when a user trusts a website that it visits and consequently a web browser not picking up malicious code as it trusts the webserver behind that website and thus expects it to deliver "trustworthy content".[118] In the case of a successful XSS exploitation an attacker will have full access to cookies and another sensitive data, but also potentially the webcam or microphone of a user can be activated if these are accessible to the website that is being visited.[119]

2.4 Chapter conclusions

This chapter described the most important aspects of exploiting vulnerabilities. It explains how attackers can obtain vulnerabilities and illustrates that exploiting flaws in software and hardware does not come down to a simple application that is run, but more importantly is about a specific tailor-made attack enabling attackers to gain access to the target system. Furthermore, the severity of vulnerabilities has been explored by touching upon the Spectre, Meltdown and checkm8 vulnerabilities which allows for a proper comprehension of the potential consequences when law enforcement goes about exploiting weaknesses. More importantly, however, is that the focus of this chapter was to provide sufficient understanding from a more technical point of view of how police (i.e. an attacker) accordingly will operate in practice when it utilises flaws in computer systems and what it can achieve with these techniques, in order to apprehend whether the safeguards that are put in place by the law, and are discussed in the next chapter, indeed are sufficient to prevent such severe effects. In sum, the general technical framework of exploiting vulnerabilities has been set out in this chapter; the next chapter will focus on the legal framework.

[117] Ibid.
[118] Michael E. Whittman, Herbert J. Mattord, David Mackey, Andrew Green, *Guide to Network Security* (Cengage Learning 2012), p. 351.
[119] Ibid.

3 Under which circumstances and how is law enforcement authorised to exploit vulnerabilities?

Over the years, law enforcement has been resorting to exploiting vulnerabilities as one of the main methods to obtain access to devices of suspects. In the literature, this use of vulnerabilities by police is often subsumed under the notion of 'lawful hacking' or 'police hacking'.[120] Though this term includes a wide range of other methods and techniques that accordingly allow police to gain access to a system and extract evidence, legislators have labelled "exploiting existing vulnerabilities in software in order to gain control of devices or networks to remotely extract material or monitor the user of the device" as perhaps the most complex operation police can carry out under this umbrella term of lawful hacking techniques.[121] Continuing, Dutch law does not explicitly use the term "lawful hacking" but instead speaks of an "investigation in a computer".[122] This chapter examines the issues surrounding an investigation into a computer, more specifically, the use of vulnerabilities under this power and discusses what police are allowed to do under this authority and analyse what this power actually means in practice.

3.1 Exploiting vulnerabilities according to the Dutch CCP

Very limited police hacking powers were expanded in 2018 with the passing of a new law, the Computer Crime III Act (Wet Computercriminaliteit III). The Computer Crime III Act (hereinafter: Dutch CCP) has officially entered into force as of this year and has formally introduced "lawful hacking" as an investigatory power into the Dutch Code of Criminal Procedure. The hacking law as it is often called, details when law enforcement can actually go about entering into computers, the procedures that have to be followed and which safeguards are in place.

3.1.1 Formal requirements

In terms of formal requirements, Article 126nba(1) of the Dutch CCP starts off with an exhaustive list of purposes for which the new investigative powers can be used amongst which are determining the characteristics (e.g. location, identity) of a computer or user, recording

[120] Ivan Škorvánek, Bert-Jaap Koops, Bryce Clayton Newell, and Andrew Roberts, "'My Computer Is My Castle': New Privacy Frameworks to Regulate Police Hacking' (2019) Pre-Press Draft, forthcoming in BYU Law Review, p. 6.
[121] Investigatory Powers Act 2016, Code of Practice, Equipment Interference (2018) <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/715479/Equipment_Interference_Code_of_Practice.pdf> accessed 10 September 2019.
[122] Dutch CCP, article 126nba.

confidential communications, systematic observation and rendering data inaccessible. Police are only allowed to use these hacking techniques for investigation into severe crimes that seriously breach the rule of law and when investigation requires the use of these powers urgently.[123] To this end, the bill refers to a list of crimes in Article 67 of the Dutch CCP which broadly speaking carry a period of a maximum of four and some even eight years of imprisonment. The fact that the hacking powers may only to be exercised for these serious crimes illustrates the severity of intrusion of these investigatory powers.

Furthermore, investigators are allowed to enter into a computer (with or without a technical aid) only after an order has been obtained from the public prosecutor.[124] Before a public prosecutor can deliver such an order, a written request will need to be submitted to an investigative judge to obtain prior authorisation. After evaluation and approval of this written request, together with authorisation from a Central Exam Committee and after determining that there is no conflict with principles of proportionality and subsidiarity, the investigative judge will present a written authorisation detailing the different components of the order and the period in which it can be used.[125] In order to adequately assess whether the request meets the principles of proportionality and subsidiarity (to meet urgent need), the law requires that the order will need to contain certain specifics or information, namely[126]: the suspected criminal act, the computer into which entry is sought by means of an identifying number if possible (e.g. MAC address, IP address, IMEI-number), which part of the system is being hacked, how the power is going to be used, the exercise period and the categories of data that are targeted.[127] This information not only ensures the technique is proportional and subsidiary, but also that it is targeted and not just aimed at bulk surveillance.[128] Widening the scope of proportionality, then, the law prescribes that not only a device that is directly used by a suspect, but also devices of relatives or acquaintances that a suspect uses regularly (i.e. generally more than two times)

[123] Dutch CCP, article 126nba(1).
[124] Ibid.
[125] Dutch CCP, article 126nba(4).
[126] *Kamerstukken II 2015/16, 34 372*, no. 3, at 30.
[127] Dutch CCP, article 126nba(2)(a)(h).
[128] Mirja Gutheil, Aurélie Heetman, James Eager, Max Crawford, Quentin Liger, 'Legal Frameworks for Hacking by Law Enforcement: Identification, Evaluation and Comparison of Practices' (2017) Policy Department for Citizens' Rights and Constitutional Affairs <http://www.europarl.europa.eu/RegData/etudes/STUD/2017/583137/IPOL_STU(2017)583137_EN.pdf> accessed 11 September, p. 50.

are open for investigation.[129] Finally, the investigatory power can be applied for a period of four weeks (which can be prolonged repeatedly each time for a period of four weeks).[130]

3.1.2 Procedural requirements

First and foremost, any hacking power that is allowed by the Dutch CCP can only be carried out by investigators that are part of the technical team.[131] In other words, these powers are allowed to be exercised by special technical investigators that are specialists in the area of information- and communication technology. One important safeguard the law seeks to achieve here sees to the separation between technical investigators and investigators that are involved in the tactical/operation part of the investigation.[132] Data that is collected by the technical investigators will be analysed by the tactical investigators and not by the technical investigators themselves, to ensure that the whole process does not lead to a one-sided investigation by having the whole process brought under one team that not only is responsible for conducting the hacking technique but also evaluating it.[133] The software that is being used to gain access, then, also needs to meet certain technical requirements. In fact, these are the same requirements that are already in place for the conventional wiretap.[134] Furthermore, and perhaps most important in the context of oversight, all the investigative activities that are performed under article 126nba of the Dutch CCP must be logged.[135]

The Computer Crime III Act also stipulates certain safeguards for when the investigatory operation has ended. The hacking technique, or more specifically, the software that is used to carry out the investigation must be removed from the suspect's computer.[136] In the case it is not possible to (completely) remove or when removal creates risks to the functioning of the computer that was the target of investigation, the public prosecutor shall inform the administrator of the computer about this and additionally provide sufficient information to enable complete removal of the hacking tool after the investigation has ended.[137]

---

[129] Ivan Škorvánek, Bert-Jaap Koops, Bryce Clayton Newell, and Andrew Roberts, "'My Computer Is My Castle': New Privacy Frameworks to Regulate Police Hacking' (2019) Pre-Press Draft, forthcoming in BYU Law Review, p. 19.
[130] Dutch CCP, article 126nba(4).
[131] *Kamerstukken II 2015/16, 34 372*, no. 3, at 9.
[132] Ibid, p. 31.
[133] Ibid, p. 30; Ivan Škorvánek, Bert-Jaap Koops, Bryce Clayton Newell, and Andrew Roberts, "My Computer Is My Castle": Ivan Škorvánek, Bert-Jaap Koops, Bryce Clayton Newell, and Andrew Roberts, "'My Computer Is My Castle': New Privacy Frameworks to Regulate Police Hacking' (2019) Pre-Press Draft, forthcoming in BYU Law Review, p. 19.
[134] Dutch CCP, article 126ee.
[135] Dutch CCP, article 126nba(8)(b).
[136] Dutch CCP, article 126nba(6).
[137] Ibid.

Moreover, the Computer Crime III Act also introduces several mechanisms that require to notify simply any target that has been targeted by the police hacking hereby arguably ensuring a right to an effective remedy for these targets.[138]

3.2 Police hacking by exploitation of vulnerabilities in detail

As the purpose of this thesis is to research the power to exploit vulnerabilities, the next sections will explain what the previously mentioned rules and safeguards actually mean in practice. The use of vulnerabilities by law enforcement has received substantial criticism over the years the Computer Crime III Act was being discussed in parliament (criticism relating to both the use of 0-days and existing vulnerabilities). Dutch NGO, Bits of Freedom, in particular has been raising attention to the use of 0-days by police.[139] On many occasions it reported that this use potentially leads to less security for society as government agencies will be less inclined to share the discovery of an unknown vulnerability with the appropriate vendor as such a vulnerability allows police to keep gaining access to target devices as long as this weakness isn't patched.[140] Simultaneously, the longer it takes for a certain piece of software to be patched, the more opportunity other attackers have to find and exploit the same vulnerability. These discussions in and outside parliament have eventually led to this issue being addressed in the final version of the law, as an amendment was made to it which introduced an obligation for police to notify vendors about unknown vulnerabilities that it has discovered during an investigation.[141] Only in the case when an investigation urges this, police are allowed to delay such notification for which prior authorisation will need to be obtained.[142]

3.2.1 The vulnerability market: control of vulnerabilities

Even though the amendment seems to resolve the issue surrounding the disclosure of 0-days by law enforcement, careful analysis of the matter still allows for certain concerns to be raised. The Minister of Justice and Security - when asked about ways to acquire 0-days - stated that

---

[138] Mirja Gutheil, Aurélie Heetman, James Eager, Max Crawford, Quentin Liger, 'Legal Frameworks for Hacking by Law Enforcement: Identification, Evaluation and Comparison of Practices' (2017) Policy Department for Citizens' Rights and Constitutional Affairs <http://www.europarl.europa.eu/RegData/etudes/STUD/2017/583137/IPOL_STU(2017)583137_EN.pdf> accessed 11 September, p. 52.

[139] Rejo Zenger, 'Onbekende kwetsbaarheden als disruptive technology' (*Bits of Freedom*, 5 July 2017) <https://www.bitsoffreedom.nl/2017/07/05/onbekende-kwetsbaarheden-als-dispruptive-technology/> accessed 10 September 2019.

[140] Bart Jacobs, 'Policeware' (2009) 39 Nederlands Juristenblad, p. 2763-2764.

[141] *Kamerstukken II 2016-17, 34 372*, no. 14.

[142] Dutch CCP, article 126ffa.

the Act (though not explicitly mentioned in it) does not allow police forces to directly purchase 0-days from third parties on the market.[143] This means that when examining law enforcement's use of 0-days there are thus two different scenarios that are legally possible: police researchers can set out to discover a 0-day by themselves and then exploit such a 0-day after which it will have to disclose the weakness to the appropriate vendor. The consequence of this being that police thereby closes this window of opportunity for any future use as the vendor accordingly will release a patch which closes the 0-day. Another way, however, for law enforcement to make use of a 0-day (indirectly) without obstructing the possibility to make use of the 0-day once again in the future, is when it acquires a hacking tool that contains an exploit for a vulnerability that has not been reported to the vendor yet.[144] As in such a case the police will not know how the tool actually functions from the ground up as it wasn't the police who developed the tool but some third party, and as in most cases the police won't be able to retrieve the exact details of the exploit and vulnerability, it will be difficult to report or disclose about the details of the exploit or vulnerability.[145] The San Bernardino case is an example of a case in which a law enforcement agency resorted to the vulnerability market and acquired a tool that contained a 0-day giving it access to suspect's iPhone.[146]

Some researchers have argued that by allowing this practice, police in a way incentivises or perhaps cultivates a market for hacking tools that exploit 0-days.[147] There is a growing concern that companies will principally recourse to the business of finding 0-days, not with the intent of reporting these to the appropriate vendor so that a patch can be developed, but generally more so to gain financial benefits with law enforcement and intelligence agencies as active customers.[148] Some have even proposed that governments actually drive these vulnerability markets.[149]

---

[143] *Handelingen TK 2016/17*, no 34, item 26, at 38-39.
[144] Ibid.
[145] Ibid.
[146] Ellen Nakashima, 'FBI paid professional hackers one-time fee to crack San Bernardino iPhone' (*Washington Post*, 12 April 2016) <https://www.washingtonpost.com/world/national-security/fbi-paid-professional-hackers-one-time-fee-to-crack-san-bernardino-iphone/2016/04/12/5397814a-00de-11e6-9d36-33d198ea26c5_story.html> accessed 21 September 2019.
[147] Riana Pfefferkorn, 'Security Risks of Government Hacking' (2018) The Center for Internet and Society <http://cyberlaw.stanford.edu/files/publication/files/2018.09.04_Security_Risks_of_Government_Hacking_Whitepaper.pdf> accessed 21 September 2019, p. 5.
[148] Riana Pfefferkorn, 'Security Risks of Government Hacking' (2018) The Center for Internet and Society <http://cyberlaw.stanford.edu/files/publication/files/2018.09.04_Security_Risks_of_Government_Hacking_Whitepaper.pdf> accessed 21 September 2019, p. 6.
[149] Martin C. Libicki, Tim Webb, *The Defender's Dilemma: Charting a Course Toward Cybersecurity* (RAND Corporation 2015), p. 53.

Continuing, one of the prime reasons why law enforcement will not easily be able to retrieve the substantial details behind the actual exploit and vulnerability that underlie these hacking tools, is that many of these companies that are in the market of selling such tools generally want to prevent the vulnerabilities they have found to be publicly known and thus to keep them as secret as possible in order to be able to keep trading the hacking tool.[150] Though this by itself does not really cause any issues, from a more analytical point of view, however, this can become problematic if you provide a broader perspective and include the power of police to acquire tools that incorporate (0-day) vulnerabilities without a need to disclose.

In the San Bernardino case, the FBI's Executive Assistant Director for Science and Technology explained how "the FBI purchased a method from an outside party in order to be able to unlock the suspect's device, but not with that purchase the rights to technical details about how the method functions, or the nature and extent of any vulnerability upon which the method may rely in order to operate".[151] This brings to light a serious loophole: purchasing a hacking tool containing a 0-day and not the rights to the technical details of the vulnerability itself, and accordingly having this contractual relationship protected by for instance an NDA, means that where police buys such a tool and later finds out about the details of the vulnerability, it is forbidden to inform the appropriate vendor. This raises questions as to whether law enforcement should be allowed to engage in this practice. The amendment that is made to the Dutch CCP clearly aims at providing a means for law enforcement to gain access to devices for the purpose of solving crime, whilst at the same time addressing security by enforcing disclosure after its use.[152]

Though police might elevate the issue to "simply not having knowledge about the technical workings of the tool and weakness(es) it utilises" or raise the argument that "buying just the rights to the use of a tool is significantly less expensive than purchasing rights to both the use and the technical details", however, in a way this distinctly resembles a path to circumvent the disclosure provisions contained in the Dutch CCP and to thus not have to report

---

[150] Joseph Menn, Mark Hosenball, 'Apple iPhone unlocking maneuver likely to remain secret' (*Reuters*, 14 April 2016) <https://www.reuters.com/article/us-apple-encryption-whitehouse-idUSKCN0XB05D?feedType=RSS&feedName=technologyNews&utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+reuters%2FtechnologyNews+%28Reuters+Technology+News%29> accessed 21 September 2019.

[151] Eric Geller, 'FBI says it won't submit tool used to hack San Bernardino iPhone for disclosure review' (*The Daily Dot*, 27 April 2016) <https://www.dailydot.com/layer8/fbi-san-bernardino-iphone-exploit-no-vulnerability-disclosure/> accessed 21 September 2019.

[152] *Handelingen TK 2016/17*, no 34, item 26, at 26.

about a method that allows for access.[153] Police can in any case actively question whether the tool it is about to purchase contains a 0-day. In this regard, the law could mitigate this issue by prescribing that law enforcement along with acquiring such a tool carry out an in-depth investigation on the hacking tool to uncover details about the vulnerability and exploit allowing it to disclose this information to the appropriate vendors after its use. However, no existing research has been found to support such a measure and the issue of contractual relationships preventing disclosure remains to exist in this scenario (forbidding the acquisition of such contracts could then again address that problem). In sum, this practice asks for more proactive behaviour from law enforcement to ensure this indeed does not become a legitimate loophole.

Another aspect from a more ethical perspective can be found by zooming in on the vulnerability seller. Many of these companies that are in the business of selling 0-days are questioned as they do not only sell these vulnerabilities to governments (law enforcement and intelligence services) and other legitimate private companies, but also to "human rights-violating nations, organised crime or other abusive actors".[154] When the Italian hacking firm "Hacking Team", fatefully, got hacked by a hacker who goes by the pseudonym of "Phineas Fisher", the public got to witness what it means when governments do business with such companies whose sole purpose is to just sell vulnerabilities for simply the highest price, regardless of what happens with these tools.[155] Fisher, in a statement months after the hack, explained that his purpose was to stop the company as it "abused human rights at a global scale" and with leaking the massive amount of internal documents and e-mails he wanted to showcase how different tools and services that were being sold to police and intelligence, weren't necessarily used for the legitimate purpose of solving crime.[156] Contrarily, as the leak shows, many of the tools that Hacking Team sold were used against journalists and activists amongst which the attack on human rights defender Ahmed Mansoor (who was even targeted on three different occasions, in one attack by using a 0-day and in the two others an existing

---

[153] Jason Healey, 'The U.S. Government and Zero-Day Vulnerabilities: From Pre-Heartbleed to Shadow Brokers' (2016) Journal of International Affairs <https://jia.sipa.columbia.edu/sites/default/files/attachments/Healey%20VEP.pdf> accessed 21 September 2019, p. 13.

[154] Robert K. Knake, 'FBI to Apple: We Would Probably Disclose the iPhone Flaw if We Knew What It Was' (*Council on Foreign Relations*, 29 March 2016) <https://www.cfr.org/blog/fbi-apple-we-would-probably-disclose-iphone-flaw-if-we-knew-what-it-was> accessed 21 September 2019.

[155] Violet Blue, 'How spyware peddler Hacking Team was publicly dismantled' (*Engadget*, 9 July 2015) <https://www.engadget.com/2015/07/09/how-spyware-peddler-hacking-team-was-publicly-dismantled/?guccounter=2> accessed 21 September 2019.

[156] Lorenzo Franceschi-Bicchierai, 'The Vigilante Who Hacked Hacking Team Explains How He Did It' (*Motherboard*, 15 April 2016) <https://www.vice.com/en_us/article/3dad3n/the-vigilante-who-hacked-hacking-team-explains-how-he-did-it> accessed 21 September 2019.

vulnerability and social engineering).[157] Hacking Team in this regard isn't alone as the Gamma Group, the NSO Group and many more organisations were and still are in the same business of selling tools incorporating all kinds of vulnerabilities to simply the highest bidder.[158]

Documents published by Wikileaks disclose how Dutch police has been in contact with Hacking Team and though no exact details have been documented concerning this relationship, the publications indicate that Dutch police was likely to buy a hacking tool ("RCS Galileo") from this company.[159] Other Wikileaks documents make clear that it was not the first time the police has been in contact with such a vendor as in the period of 2012 until 2014, the police had acquired several licenses for a number of FinFisher applications from Gamma Group.[160] Up until today, however, the police keep denying having had any contractual relationships with such vendors as several public disclosure requests made by Buro Jansen received the reply that no contract has been concluded or used vague wordings such as "it cannot be confirmed nor denied that any relationship exists between the two parties" (leaving open the possibility for these tools to have been acquired by a subsidiary).[161]

3.2.2 Losing control and possession of vulnerabilities

Whereas many discussions in the past have tended to focus on 0-days - and for that matter still do[162] - already known (i.e. existing) vulnerabilities which are used by law enforcement are equally important to take into consideration in this analysis. Experts have argued that in order to actually be able to gain access to devices it is not necessary to make use of 0-days as almost every device contains vulnerabilities that are already known and for which a vendor might have already developed a patch, but for reasons like people not updating their systems or not in due

---

[157] Bill Marczak, John Scott-Railton, 'The Million Dollar Dissident - NSO Group's iPhone Zero-Days used against a UAE Human Rights Defender' (*Citizenlab*, 24 August 2016) <https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/> accessed 21 September 2019.

[158] Dan Sabbagh, 'Israeli firm linked to WhatsApp spyware attack faces lawsuit' (18 May 2019) <https://www.theguardian.com/world/2019/may/18/israeli-firm-nso-group-linked-to-whatsapp-spyware-attack-faces-lawsuit> accessed 21 September 2019.

[159] Buro Janssen, 'De Nederlandse politie en Hacking Team - Flirten met de tools van de dictator' (*Buro Janssen*, 23 January 2017) <https://www.burojansen.nl/observant/de-nederlandse-politie-en-hacking-team-flirten-met-de-tools-van-de-dictator/> accessed 21 September 2019, p. 4.

[160] Buro Janssen, 'Gamma Group en de politie; FinFisher trojan in de Nationale politie' (*Buro Janssen*, 22 January 2017) <https://www.burojansen.nl/observant/gamma-group-en-de-politie-finfisher-trojan-in-de-nationale-politie/> accessed 22 September 2019.

[161] Ibid.

[162] Olaf van Miltenburg, 'Overheid gaat gebruik zero-days door AIVD en MIVD toetsen' (*Tweakers*, 15 March 2018) <https://tweakers.net/nieuws/136335/overheid-gaat-gebruik-zero-days-door-aivd-en-mivd-toetsen.html> accessed 23 September 2019.

time, remain open for exploitation.[163] Security companies and intelligence agencies have stipulated that in almost all their daily operations they encounter and rely more on existing vulnerabilities that either directly or indirectly allow for access.[164] The NSA's chief hacker disclosed at a security conference that "a lot of people think that nation states are running their operations on zero days, but it's not that common. For big corporate networks, persistence and focus will get you in without a zero day; there are so many more vectors that are easier, less risky, and more productive".[165]

In this regard, when examining the Dutch CCP, it becomes clear that there is one vital issue that the law does not consider in any way which is the possibility of law enforcement losing possession and thereby the control over vulnerabilities or tools that incorporate existing vulnerabilities.[166] That this is fact rather than fiction was witnessed when the loss of the so-called BlueKeep vulnerability by the US intelligence agency, led to other third parties examining the tool and using the vulnerability in different tools.[167] Law enforcement agencies can lose possession over a vulnerability for instance because of an employee or insider leaking information about it or simply because of not having properly secured the infrastructure in which the vulnerability information is kept (i.e. getting hacked).[168] In such a case, other threat actors are able to make use of this information for as long as the vendor hasn't developed a patch (again stressing the urge for swift disclosure). Important in this analysis is to realise that government agencies just like any other entity or person in possession of a certain vulnerability never has full control and thus is vulnerable and perhaps even prone to attacks that can have a major impact on not only national security, but also the judicial process and integrity of evidence as will be discussed in chapter 4.[169]

---

[163] 'Position paper Fox-IT t.b.v. hoorzitting/rondetafelgesprek Computercriminaliteit III d.d. 11 februari 2016' (2016) Tweede Kamer 2016D06073 <https://www.tweedekamer.nl/downloads/document?id=19cb6a18-cb5f-4a39-8cf9-1defac7920ae&title=Position%20paper%20Fox-IT%20t.b.v.%20hoorzitting%2Frondetafelgesprek%20Computercriminaliteit%20III%20d.d.%2011%20februari%202016.pdf> accessed 24 September 2019.

[164] Ibid.

[165] Bruce Schneier, *We Have Root: Even More Advice from Schneier on Security* (John Wiley & Sons, Inc. 2019), p. 270.

[166] Scott Shane Nicole Perlroth, David E. Sanger, 'Security Breach and Spilled Secrets Have Shaken the N.S.A. to Its Core' (*NY Times*, 12 November 2017) <https://www.nytimes.com/2017/11/12/us/nsa-shadow-brokers.html> accessed 8 September 2019.

[167] Andy Greenberg, 'The Strange Journey of an NSA Zero-Day—Into Multiple Enemies' Hands' (*WIRED*, 7 May 2019) <https://www.wired.com/story/nsa-zero-day-symantec-buckeye-china/> accessed 8 September 2019.

[168] Riana Pfefferkorn, 'Security Risks of Government Hacking' (2018) The Center for Internet and Society <http://cyberlaw.stanford.edu/files/publication/files/2018.09.04_Security_Risks_of_Government_Hacking_Whitepaper.pdf> accessed 21 September 2019, p. 7.

[169] Valarie Findlay, 'Cyber Threats Against Police' (*National Police Foundation*) <https://www.policefoundation.org/cyber-threats-a-glocal-problem-for-law-enforcement/> accessed 25 November 2019.

Even though the Dutch CCP was developed after the Dutch government witnessed that losing vulnerabilities is genuinely a risk for law enforcement and intelligence services[170], there hardly seems to be any realisation of this (at least from a more formal perspective). In the situation possession is lost, law enforcement arguably should develop a way to know that this vulnerability information is leaked in order to be able to supply the appropriate vendor of this knowledge in due time, so that the vendor can develop a patch and urge its customers to install it as soon as possible. Including a provision that explicitly requires that the appropriate vendor is informed as soon as police discovers that control over a vulnerability or tool has been lost might seem redundant; however, looking at behaviour of other agencies perhaps does make this necessary.[171]

3.2.3 Vulnerability patch dynamics

Nevertheless, timely as police might be, in some cases timely informing a vendor about a leak still doesn't prevent severe damage from occurring. In the case of the BlueKeep vulnerability the hacker group that got hold of the tool and exploited this weakness had made public, before it released it, that it had possession of this tool allowing the NSA to notify Microsoft to develop a patch.[172] Where in this case the patch was developed quickly upon finding out, this attack shows how there is another dynamic that should also be taken into account, namely user adoption. Users will need to update their software (i.e. install a patch) to fix the weakness that it contains, meaning that generally there is no way for a patch to be installed remotely without any user (inter)action. Research by IBM[173] showed already back in 2016 that humans (i.e. human error) in this regard are the weakest link in the chain of security and cause up to 95% of cybersecurity incidents.[174] So, because users are not always aware or simply too "lazy" to directly patch, attackers often long after a patch has been developed (sometimes even years) are still able to exploit the vulnerability and cause damage.

---

[170] Lorenzo Franceschi-Bicchierai, "Hackers Hit 'Some' Cisco Customers With Leaked NSA Hacking Tools" (*Motherboard*, 19 September 2016) <https://www.vice.com/en_us/article/3dajyn/hackers-hit-cisco-customers-leaked-nsa-hacking-tools-shadow-brokers> accessed 24 September 2019.

[171] Huib Modderkolk, *Het is oorlog, maar niemand die het ziet* (Podium Amsterdam 2019), p. 204.

[172] Riana Pfefferkorn, 'Security Risks of Government Hacking' (2018) The Center for Internet and Society <http://cyberlaw.stanford.edu/files/publication/files/2018.09.04_Security_Risks_of_Government_Hacking_Whitepaper.pdf> accessed 21 September 2019, p. 9.

[173] IBM, 'IBM 2015 Cyber Security Intelligence Index' (*IBM*, 2016) <https://essextec.com/wp-content/uploads/2015/09/IBM-2015-Cyber-Security-Intelligence-Index_FULL-REPORT.pdf> accessed 24 September 2019.

[174] Curtis Franklin Jr. BC, *Securing the Cloud: Security Strategies for the Ubiquitous Data Center* (Auerbach Publications 2019), p. 178.

To complicate the situation even further, many vendors struggle to develop an adequate fix for a vulnerability or, as explained in chapter 2, in the case of hardware vulnerabilities generally are not even capable of doing so. Furthermore, they are often under an enormous time pressure to deliver a patch because of the severity of many of the vulnerabilities that exist nowadays.[175] Also, in some cases there might be a patch for a certain device which is then still deemed "un-patchable", not from a technical perspective, but from a management/business perspective. What is meant by this is that users or owners sometimes apply a more risk-based approach as they have to choose between on the one hand "running the machine that contains the vulnerability" and on the other hand "halting their operations by taking the machine out of service" in order to be able to update it.[176] As the latter case could lead to the loss of some significant income, especially when a company is dependent on a number of key important machines, some businesses and users decide to keep running a device knowing that it contains the vulnerability (thus deciding that the security risks don't weigh up against the financial risks). This all is vital in understanding many dynamics surrounding the use of vulnerabilities by police forces and more importantly underlines the necessity of adequate safeguards and measures.

3.2.4 Security of devices

The Dutch CCP does not differentiate between the type of vulnerability it exploits or, more important, the computer or device (i.e. "automated work") it seeks to gain access to. The notion of an automated work is defined broadly in the Dutch CCP and can encompass almost every electronic device ranging from a smartwatch to a router.[177] Law enforcement can set out to discover or find a vulnerability in basically all these devices without limiting or defining in further detail the type of device. Discussions have been raised as to whether law enforcement should be allowed or not to look for a vulnerability in certain types of systems or devices, for instance in cars. When examining the current legislation, the provisions in the Dutch CCP seem to allow law enforcement to exploit a vulnerability in a car system potentially allowing it to not only extract essential information, but also to perform certain actions like slowing it down

[175] Zak Doffman, 'Apple Fixes Serious iOS 13, iPadOS 13 And Catalina Security Issues: Update Your Devices Now' (*Forbes*, 31 October 2019) <https://www.forbes.com/sites/zakdoffman/2019/10/31/apple-patches-serious-ios-13-and-catalina-security-issues-update-your-devices-now/#63c8c8992c2a> accessed 25 November 2019.
[176] Riana Pfefferkorn, 'Security Risks of Government Hacking' (2018) The Center for Internet and Society <http://cyberlaw.stanford.edu/files/publication/files/2018.09.04_Security_Risks_of_Government_Hacking_Whitepaper.pdf> accessed 21 September 2019, p. 11.
[177] Dutch CCP. article 80sexies.

or shutting it off completely. Schönfeld (Chief Innovation Officer at the Dutch police) has confirmed this practice as he reported that police is already experimenting with this new movement as it has been trying to find out whether they can stop or drive a car to a certain location".[178] Schönfeld explains that they have tested several vehicles from different car manufactures including Mercedes, Tesla and Toyota and this testing has been done "in collaboration with these car companies because this information is valuable to them, too. If the police can hack into their cars, others can as well".[179]

Where the Dutch police seems to approach this issue from a more experimental perspective, a Wikileaks disclosure[180] shows how other agencies have potentially been developing tools and exploiting car system vulnerabilities as an actual method in practice to gain access to the computer systems contained in a suspect's vehicle.[181] Bruce Schneier, security expert, gives an excellent representation of the role of vulnerabilities in different consumer devices which goes beyond the scope of this thesis, but the central issue he touches upon which was also addressed during the Computer Crime III Act's consultation, is that the law arguably should make a distinction between the different computer devices that law enforcement is allowed to exploit.[182] In other words, allowing law enforcement agencies to find and exploit vulnerabilities in simply "any automated work" without having any procedural or technical requirements setting out a baseline or boundaries, gives police an umbrella power that does not take into consideration the global infrastructure of connected devices and the impact on this infrastructure this might have. This is especially important as there are agencies already considering entering into the domain of medical devices like a pacemaker.[183]

3.3 Chapter conclusions

When examining the power of police to exploit vulnerabilities, without a doubt, this technique offers an extremely effective way to gain access to devices. Law enforcement agencies in this regard can rely on their own research and investigations to find vulnerabilities and develop

---

[178] Anouk Vleugels, 'Police can remotely drive your stolen Tesla into custody' (*The Next Web*, 19 November 2018) <https://thenextweb.com/the-next-police/2018/11/19/police-control-your-self-driving-cars/> accessed 24 September 2019.
[179] Ibid.
[180] Wikileaks, 'Branch Direction Meeting notes' (*Wikileaks*, 23 October 2014) <https://wikileaks.org/ciav7p1/cms/page_13763790.html> accessed 24 September 2019.
[181] Bruce Schneier, *Click Here to Kill Everybody: Security and Survival in a Hyper-connected World* (W. W. Norton & Company 2018), p. 86.
[182] Ibid.
[183] Jenna McLaughlin, 'NSA Looking To Exploit Internet of Things, Including Biomedical Devices, Official Says' (*The Intercept*, 10 June 2016) <https://theintercept.com/2016/06/10/nsa-looking-to-exploit-internet-of-things-including-biomedical-devices-official-says/> accessed 24 September 2019.

accustomed exploits or they can resort to the vulnerability market to acquire these flaws. Important in this regard is that Dutch law seeks to prevent the trade in 0-days because of their severity and importance when it comes to security, but also to stimulate vulnerability disclosure. Conversely, the law has created an exception (i.e. amendment) to the rule that prevents the acquisition of 0-days as police are allowed to purchase hacking tools that make use of 0-days. The government in this regard is increasingly becoming an important, stimulating actor on the vulnerability market which raises many concerns. As the law does not instil any options to audit law enforcement agencies that buy these hacking tools, there is little supervision that guarantees that police aren't actively using this exception as a circumvention. Police participation in the vulnerability market along with this lack of transparency therefore exposes them to many risks and arguably extend these risks to the public (e.g. BlueKeep exploitation). This mainly has to do with the fact that law enforcement in the end for a great part is dependent on the vendors and users of devices and software to fix any misuse of a vulnerability they have called into existence.

Finally, vulnerability exploitation, or lawful hacking, is by some regarded as an alternative to encryption backdoors as it allows police to target specific individuals and does not necessitate to alter software or hardware for this purpose (the latter affecting many more people than the intended targets).[184] However, when examining the above, it seems that vulnerability exploitation under the current regime, tends to shift back the risks and effects to the larger public instead of specific individuals. Agencies arguably should take responsibility for whatever happens when tools get leaked and that they put in all the work to mitigate the damage resulting from such actions (simply reporting the vulnerability to a vendor to have it patched is not sufficient). [185] Other ways these effects and risks can be mitigated is by implementing (i.e. legally enforcing) special technical measures. These measures then are also necessary for law enforcement to not be equally challenged in court and to preserve integrity of gathered digital evidence. Chapter 4 will explore why this otherwise might be the case.

[184] Alan Z. Rozenshtein, 'Wicked Crypto' (2018) 9 UC Irvine Law Review 1181, p. 1207

[185] Joseph Cox, 'Your Government's Hacking Tools Are Not Safe' (*Motherboard*, 14 April 2017) <https://www.vice.com/en_us/article/d7bvxa/your-governments-hacking-tools-are-not-safe> accessed 24 September 2019.

4 Controls seeking to guard the chain of evidence

The previous chapter touched upon the controls that exist to prevent misuse and proliferation. Several situations have been examined which demonstrated that these controls in a lot of cases are not sufficient or inadequate. This chapter seeks to shine a light on measures that can be implemented technically which consequently can also be enforced legally thereby creating safeguards that ensure certain guarantees. After describing how some of these measures contribute to a fair investigation, the latter part of this chapter will discuss what omitting such measures could mean for the integrity of evidence that is gathered and how suspects could potentially misuse this knowledge.

4.1 (Technical) means to prevent proliferation

4.1.1 Automated exit in the case of not targeted devices

The Stuxnet attack that was described in the first chapter, fundamentally shows the importance of including certain security or technical measures when law enforcement exploits a device. More specifically, incorporating a measure to "exit" a computer system the police are targeting. What this means in practice can be described by examining the Stuxnet attack that exploited several vulnerabilities. Stuxnet was able to infect Windows computer systems that were located in Iranian nuclear plants (centrifuges) by abusing four different weaknesses: the first allowed the Stuxnet worm to be installed onto a machine via a USB-stick, one allowed the worm to spread from that machine to others on the network and finally, the other two vulnerabilities allowed the attackers to gain certain rights or privileges on the infected machines making it possible to enter commands onto the computer systems.[186] When the attack was carried out on the centrifuges, the second vulnerability (at that time a 0-day) abused Step 7 software (Siemens software that was used to program and control the machines) in order to infect the systems.[187] Important was that in order to abuse the software, Stuxnet would first try each and every computer to check whether it contained this software and only after it encountered the software on a machine, it would run the rest of the malware and thus exploit the final vulnerability.[188] If no Step 7 software was found, the malware would "silently exit".

[186] Chwan-Hwa Wu, J. David Irwin, *Introduction to Computer Networks and Cybersecurity* (CRC Press 2013), p. 31.
[187] Ibid.
[188] Nicolas Falliere, Liam O. Murchu & Eric Chien, 'W32.Stuxnet Dossier' (*Symantec Security Response*, February 2011)

Similarly, law enforcement should in the exploits they develop that take advantage of a vulnerability include a similar process. Police could, for instance, incorporate in their attack that only when a certain MAC address matches the MAC address to which the warrant is written out, the attack will be carried out.[189] The Dutch CCP requires police to include, amongst other details, the MAC address of their target in the request for authorisation to an investigative judge (article 126nba(a)(h)), meaning that this information is already known at the time of creating a way to exploit a vulnerability. The safeguard here, however, would be to instate that the exploit that is developed or acquired should also incorporate some piece of code that prevents spreading to non-targeted machines similarly to how Stuxnet only infected those machines that it properly identified.[190] This allows for a proactive as opposed to a reactive approach and prevents malicious third parties, by and large, from discovering the government exploit and subsequently creating their own tools to create a new exploit or simply repurposing the exploit to target other victims.[191]

4.1.2 Use a dropper with an encrypted payload

A vulnerability is the most vital part of an exploit or hacking tool, in other words, having knowledge about a vulnerability analogously means knowing which door in a certain house has a faulty lock that can be opened. Therefore, law enforcement agencies would want to protect discovery of a vulnerability it has at hand as best as possible, even in the stage after which it is used. A technical measure that could be implemented to prevent this from happening as suggested by scholars is by obliging encryption of all the tools that are used, or more technically, to use a so-called "dropper" to exploit a vulnerability.[192] In practice, after law enforcement has developed an exploit for a certain vulnerability and has successfully gained access, it will for instance want to install a keylogger or other piece of malware to intercept logins or communications. To not give away any details of the vulnerability, law enforcement

<https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf> accessed 1 November 2019, p. 3.

[190] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 52.
[191] Riana Pfefferkorn, 'Security Risks of Government Hacking' (2018) The Center for Internet and Society <http://cyberlaw.stanford.edu/files/publication/files/2018.09.04_Security_Risks_of_Government_Hacking_Whitepaper.pdf> accessed 21 September 2019, p. 10.
[192] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 52.

can, or should, use a dropper which – as the name suggests – drops the piece of malware on the system of the suspect.[193]

What this means in practice is the following. A dropper consists of two stages: firstly, it exploits a vulnerability and thus provides access to the system. Secondly, it drops the malicious code onto a system. In the case of law enforcement exploiting a vulnerability, it should build a dropper that contains an encrypted payload (i.e. the code that carries the instructions that constitutes an attack) so that details about how access is gained cannot be easily detected or re-used by criminals and, equally important, ensures that the payload targets the appropriate device.[194] The way this can be achieved, again stressing the importance of specifying the importance of certain technical details like a MAC address in the warrant, is to have an identifier as the "key to encrypt and decrypt the payload".[195]

As discussed in chapter 2, in the reconnaissance phase, police will have picked up such an identifier and can use it as a cryptographic key that not only protects its exploits against other malicious actors, but also prevents it from gaining access to machines that it should not enter as decryption will not be possible. This technical measure alongside the requirement in the Dutch CCP to specify certain technical information in the request to an investigative judge, creates an important form of oversight or safeguard during the investigation phase. This approach has been used by US government agencies on multiple occasions (for instance when it used the Regin malware that exploited the Belgian telecommunication provider Proximus (previously Belgacom), but also in a different form through the so-called Gauss malware).[196] Furthermore, legally enforcing to encrypt these processes of exploitation is already established in German law and arguably would also serve Dutch investigations in terms of establishing adequate safeguards.[197]

---

[193] Ibid.
[194] Alex Matrosov, Eugene Rodionov, Sergey Bratus, *Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats* (No Starch Press 2019), p. 199.
[195] Mohammed Abbas Fadhil Al-Husainy, 'MAC Address as a Key for Data Encryption' (2013) 11 IJCSIS 83.
[196] Symantec, 'Regin: Top-tier espionage tool enables stealthy surveillance' (*Symantec*, 27 August 2015) <https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/regin-top-tier-espionage-tool-15-en.pdf> accessed 19 November 2019; Kim Zetter, *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon* (Broadway Books 2015), p. 515.
[197] Chen-Yu Li, Chien-Cheng Huang, Feipei Lai, San-Liang Lee, Jingshown Wu, 'A Comprehensive Overview of Government Hacking Worldwide' (2018) IEEE Access 18174826 <https://ieeexplore-ieee-org.tilburguniversity.idm.oclc.org/document/8470931> accessed 1 November, p. 55060.

4.1.3 Self-destructing payload

Another measure that law enforcement can implement is including a self-destruct option on the exploit (i.e. payload) it has created or acquired.[198] The payload can be programmed to for instance self-destruct after a certain time-limit meaning that it is deleted automatically after it has accomplished what it has been designed to achieve.[199] It has been suggested that in this regard, the time-limit contained in the warrant that law enforcement has obtained, could serve as an appropriate time-limit that makes sure that after a vulnerability has been exploited, it restores the target device to its so-called "pre-exploit state" and erases itself and simultaneously all evidence of it having ever been on the machine.[200] This would thus actually achieve that the time-limit that is legally assigned to police is technically enforced, and arguably also improve public security as vulnerabilities would be reported much quicker as a result of exploiting vulnerabilities being bound to a strict time-limit. It should be noted, however, that this might not always serve law enforcement as in some cases certain malware that is being deployed will have to remain on a system for a longer period to extract the evidence that is necessary making time-limits less useful. In these cases, a self-destruct option that is activated after certain data or evidence is extracted should be more appropriate.[201] The essence is to include a self-destruct option based on a time-period or goal which decreases the opportunity for proliferation.

4.2 Integrity of evidence

The previous section examined measures that take important identifiers contained in a warrant to legally enforce some important safeguards that guarantee a proper investigation. However, aside from these safeguards protecting against proliferation, there are also some concerns as to the evidence that is gathered after successful vulnerability exploitation. It could and already has been disputed that the chain of custody is compromised as suspects - more often illegitimately than legitimately – have argued that the digital evidence that has been gathered after a vulnerability has been exploited, could have been tampered with.[202] This mainly has to

---

[198] Jairo E. Serrano, Juan Carlos Martínez-Santos, *Advances in Computing: 13th Colombian Conference, CCC 2018, Cartagena, Colombia, September 26-28, 2018, Proceedings* (Springer 2018), p. 115.
[199] Kim Zetter, *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon* (Broadway Books 2015), p. 515.
[200] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1, p. 43.
[201] Kim Zetter, *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon* (Broadway Books 2015), p. 515.
[202] Aswin Gopalakrishnan, Emanuele Vineti, Ashok Kumar Mohan, M. Sethumadhavan, 'The Art of Piecewise Hashing: A Step Toward Better Evidence Provability' (2018) 7 Journal of Cyber Security and Mobility 1-2

do with the fact that vulnerabilities that exist for law enforcement simultaneously exist for every other party that discovers the same error in a piece of software code or hardware meaning that any message, photo or other piece of data that is stored on the system, arguably, could have been altered by this third party. This not only seems to be caused by the difficulty of the bits and bytes behind software and how computer systems operate, but perhaps more so because of a lack of procedures for collecting digital evidence on computer systems, not only in the Netherlands but internationally as well.[203]

4.2.1 Lack of legally enforced procedures

Research has set out that police searches that are conducted remotely (or on seized computer systems) generally are carried out twofold.[204] On the one hand, there is forensic research that is ascribed to detectives specialised in collecting digital evidence and the National Forensic Institution (NFI). These examinations can be described as high-quality technical investigations that consist of well-documented procedures.[205] On the other hand, there are numerous police authorities that lapse into a more practical hands-on approach meaning that they apply non-computer, every-day investigative and forensic procedures on computer systems (thus not focusing on the different treatment computer systems should receive as opposed to physical offline examinations).[206] Marvis et. al, continue to explain that these more every-day searches result in some law enforcement agencies paying less attention to the "validity, collection and processing" of any found data (i.e. evidence).[207] So generally in these cases there are few internal policies that see to the "processing and gaining access to data, storing of the data and the decryption of any encrypted data" found on a device.[208] Or, phrased differently, there is no uniform approach as to how law enforcement handles evidence extraction when it comes to exploitation of vulnerabilities in systems.

<https://www.riverpublishers.com/journal/journal_articles/RP_Journal_2245-1439_719.pdf>, accessed 2 November 2019, p. 110.

[203] Ibid.

[204] Paul A.M. Mevis, Joost H.J. Verbaan, Barbara A. Salverda, 'Onderzoek aan in beslag genomen elektronische gegevensdragers en geautomatiseerde werken ten behoeve van de opsporing en vervolging van strafbare feiten' (2016) Erasmus Universiteit Rotterdam -School of Law, WODC < https://www.wodc.nl/binaries/2598-volledige-tekst_tcm28-74084.pdf> accessed 2 November 2019, p. 78.

[205] Ibid.

[206] Ibid.

[207] Ibid.

[208] Ibid.

4.2.1.1 Telegram hack

To illustrate how problematic integrity of data in the case of vulnerabilities can be, the alleged Telegram-hack case in which the German investigative police force (Bundeskriminalamt (BKA)) gained access to devices of several suspects is illustrative. In this case, the German BKA obtained a warrant which allowed it to cooperate with a telecommunication provider and demand to route all communications of the suspects to the BKA. As this also allowed police to receive any authentication codes sent by a certain application when setting it up, this allowed the BKA to install the Telegram chat application on an investigative device, sign in with the suspect's accounts, receive the authentication codes, enter these and consequently receive and read almost all the encrypted communications to and from the Telegram accounts of the suspects.[209] This communication with plain interception of the communication data would not have been readable (i.e. understandable), because Telegram encrypts this traffic.

At the same moment in time, however, there circulated a so-called "SS7 vulnerability" that existed in the code that made up the Telegram application.[210] This vulnerability basically allowed any attacker to obtain the same type of access and to intercept the same communications as the BKA, when they enlisted the telecommunication provider and routed the communication traffic. Though not challenged in court, any (more technically skilled) defence attorney could have argued that the gathered evidence was tampered with by for instance stating that these were created by third parties and not the suspect (and thus question the chain of evidence).[211] As proposed by some researchers, however, this issue can perhaps be solved by electronically signing all digital evidence that is gathered on computer systems.[212] In this regard, no documentation of Dutch police already doing so has been encountered during this research (which does not imply that this practice is not implemented in some investigations).

---

[209] Sven Herpig, 'Government Hacking: Computer Security vs. Investigative Powers paper' (*Stiftung Neue Verantwortung*, 21 June 2017) <https://www.stiftung-nv.de/sites/default/files/snv_tcf_government_hacking-problem_analysis_0.pdf> accessed 4 November 2019, p. 9.
[210] CHEF-KOCH, 'How to Hack Whatsapp & Telegram Using SS7 Flaw', (Github, 2015) <https://gist.github.com/CHEF-KOCH/07ad6b8d3cd3d11435cc6dffb7b33d85> accessed 2 November 2019.
[211] Sven Herpig, 'Government Hacking: Computer Security vs. Investigative Powers paper' (*Stiftung Neue Verantwortung*, 21 June 2017) <https://www.stiftung-nv.de/sites/default/files/snv_tcf_government_hacking-problem_analysis_0.pdf> accessed 4 November 2019, p. 9.
[212] Ibid.

4.2.2 Legal measures to preserve integrity

Continuing, even though there is a lack of legally enforced procedures when it comes to extracting and handling data from computer systems after these have been exploited by police, the law does contain some measures that aim at preserving the reliability of this practise, or at least, the tools that are used. More specifically, these measures aim to prevent misuse by third parties from whom tools have been acquired by police in order to preserve the integrity of data that is gathered with the help of such tools. This mainly has to do with the fact that any tool acquired by police can contain a vulnerability (just like how the tool itself exploits a vulnerability) that in return can be misused by the vendor of that tool.[213] The Dutch CCP therefore establishes that any tool that exploits a vulnerability needs to be protected against alteration of the tool, against alteration of the data it registers (i.e. evidence) and against unauthorised parties gaining knowledge about these data.[214] Furthermore, to ensure a manufacturer is not able to tamper with a tool externally, authentication measures need to be implemented that ensure that any external communication with the tool is not possible.[215] And, perhaps more important in light of preserving integrity, it also prescribes measures that see to a proper extraction of data to the technical infrastructure in use by the police.

From a more analytical point of view, when examining the parliamentary piece that describes the implementation of all these measures, what stands out is that all these measures apply to any "technical aid" that is acquired or used. The text seems to have been written from the perspective of using tools to gain access to a system and does not actually differentiate between using special hacking tools and exploiting vulnerabilities on the other hand. Article 1(f) describes a technical aid as "any software application that detects, registers and transports data and that allows for investigation techniques to be carried out as warranted".[216] Exploiting vulnerabilities, as has been described in chapter 2, however, does not always take place by using a special tool designed specifically or especially for the purpose of gaining access. What actually distinguishes exploiting vulnerabilities is that it often comes down to discovering or causing errors in software and hardware which consequently allow for the escalation of privileges (i.e. to take over a system or intercept communications). Discovering or causing errors in computer systems isn't always achieved by running special tools, but by and large comes down to wrongly written code and misusing these found errors by talking to the

[213] *Kamerstukken II 2015/16, 34 372,* no. 3, at 5.
[214] *Kamerstukken II 2017/18, 34 372*, no. G, at 13.
[215] Ibid.
[216] *Stb.* 2018, 340.

application or machine (i.e. by using commands). Recently, for instance, a vulnerability was found in the Whatsapp chat-client which could be exploited by sending a modified mp4 video file which consequently allowed for a backdoor to be installed on a recipient's device.[217] This example clearly illustrates that knowledge about how a certain application or system works (i.e. errors in the code) is much more fundamental to vulnerability exploitation than using special applications or tools. Similarly, the spyware that the NSO Group had developed which allowed for remote spying on thousands of Whatsapp users in the fall of 2019 was possible due to a similar vulnerability existing in Whatsapp.[218]

This does not mean that it is impossible for vulnerability exploitation to be covered by these safeguards, but the fundamental issue here is that this power is subsumed under the general provision for hacking by law enforcement aimed at the use of special tools, whilst not taking into account how attackers actually get to exploitation. This means that these measures that seek to preserve integrity, in theory, might be overlooked. Thus, the question becomes whether these safeguards in practice actually matter or are a mere pretext when it comes to vulnerabilities. Therefore, from a more analytical point of view, concerns could be raised as to whether vulnerability exploitation will cause several evidentiary issues in the future as integrity of the investigation is not aimed at specifically exploiting weaknesses.

4.2.3 Transparency of investigation

Lastly, there is the issue of secrecy surrounding the use of special investigative techniques. Any of the "special investigatory powers" techniques or tactics that police use to gain access to a system (including the exploitation of vulnerabilities), do not need to be disclosed to the public when police are faced with a case that is in the interest of a severe investigation.[219] This means that in these situations law enforcement is not mandated to communicate which tools they possess that support their investigation, from whom these tools were acquired and so on. This evidently has to do with the fact that disclosing this information would result in people gaining knowledge about techniques or tactics, reducing the merit these tools hold as people

[217] National Vulnerability Database, 'CVE-2019-11931' (*National Vulnerability Database*, 14 November 2019) <https://nvd.nist.gov/vuln/detail/CVE-2019-11931> accessed 14 November 2019.
[218] Swati Khandelwal, 'Facebook Sues Israeli NSO Spyware Firm for Hacking Whatsapp Users' (*The Hacker News*, 29 October 2019) <https://thehackernews.com/2019/10/whatsapp-nso-group-malware.html> accessed 14 November 2019.
[219] Dutch CCP, article 187(1)(b).

would know how police operate.[220] Whether the interest mentioned in article 187d(1)(b) of the Dutch CCP is met depends on the concrete circumstances of the case.[221] Strikingly, there is little research on what these concrete circumstances look like. The law does provide some safeguards to ensure that this doesn't lead to an arbitrary process as the public prosecutor will need to obtain authorisation from an investigatory judge before it is allowed to omit technical details[222].

Defendants who do not have access to certain files containing information about the investigatory techniques and tactics used against them has been a controversial issue for years and some have even argued that this practice goes against the right to a fair trial.[223] However, the European Court of Human Rights in several cases has ruled that there is no right to an absolute disclosure of all details surrounding an investigation. [224] The Dutch Court of Appeal has ruled in several cases along the same lines that not all technical details have to be disclosed by police. In the Gimli case it was sufficient that the police described that a beacon was used to intercept communications without going into the technicalities of that beacon.[225] Questions, however, keep surfacing as to whether defendants in the case of vulnerability exploitation, should actually have a right to know about how police have operated given the many concerns surrounding integrity. As exploiting flaws in software and hardware is a delicate practice, much more sensitive to misuse or abuse than other tools and techniques police has at hand, disclosure of details surrounding the process behind a certain vulnerability that is exploited in court might prove to be appropriate to guarantee a fair trial.[226]

One way to perhaps address this issue – without exposing police techniques completely - could be to shift this "control" to the National Cyber Security Centrum (NCSC) or establish a department within the NCSC. Similarly, Germany cooperates with its cyberorganisation

---

[220] Eelco M. Moerman, 'Inburgeren in de opsporing: Over de juridische positie van de burger in de opsporing van strafbare feiten' (2016) Erasmus University Rotterdam <https://repub.eur.nl/pub/94119/Proefschrift-Moerman_.pdf> accessed 14 November 2019, p. 117-119.
[221] C.P.M. Cleiren, M.J.M. Verpalen, *Strafvordering. Tekst & Commentaar. De tekst van het Wetboek van Strafvordering en enkele aanverwante wetten voorzien van commentaar* (Kluwer 2011), p. 863.
[222] Dutch CCP, article 149b.
[223] Eelco M. Moerman, 'Inburgeren in de opsporing: Over de juridische positie van de burger in de opsporing van strafbare feiten' (2016) Erasmus University Rotterdam <https://repub.eur.nl/pub/94119/Proefschrift-Moerman_.pdf> accessed 14 November 2019, p. 117-119.
[224] *Jasper v. United Kingdom* App no 27052/95 (ECHR, 16 February 2000) para. 52.
[225] Gerechtshof 's-Hertogenbosch 8 december 2006, LJN AZ4219 (Gimli), para. C.3.
[226] The Center for Internet and Society, 'Government Hacking: Evidence & Court Disclosure of Vulnerabilities' (*The Center for Internet and Society*, 5 April 2017) <https://cyberlaw.stanford.edu/events/government-hacking-evidence-court-disclosure-vulnerabilities> accessed 14 November 2019.

ZiTis[227] which supports law enforcement "by developing methods, products, and strategies to fight criminality and terrorism on the internet.".[228] Phrased differently, the NCSC could provide assistance in issues surrounding the technicalities of vulnerabilities and in that sense would act as an authority providing "technical oversight" which would not endanger the use of vulnerability exploitation as an investigative method. As the NCSC is already equipped with the task of overseeing the policy for the coordinated vulnerability disclosure program in the Netherlands, this additional role could perhaps provide more guarantees in the process of preserving integrity not only for the purpose an investigation, but also for the purpose of a fair trial.

4.3 Chapter conclusions

This chapter described how certain technical security measures that are based on identifying information that is written down in a warrant obtained by police, can enforce safeguards that establish a fair investigation. Some of these measures are already in place in other jurisdictions which demonstrates that there is a need for technical safeguards that actually see to the exploitation of vulnerabilities aside from mere procedures. This necessity is further explored in the latter part of this chapter by examining how the integrity of evidence might be attacked when such measures are absent. Tech-savvy lawyers might misuse knowledge about existing vulnerabilities and argue for evidence tampering. Equally important in this regard is how the law makes it difficult for defendants to actually say something about the way a certain device is exploited as the exact technique that is used won't necessarily have to be disclosed in court.

[227] Bundesministerium des Innern für Bau und Heimat, 'Startschuss für ZITiS' (*Bundesministerium des Innern für Bau und Heimat*, 21 January 2017) <https://www.bmi.bund.de/SharedDocs/pressemitteilungen/DE/2017/01/zitis-vorstellung.html> accessed 14 November 2019.

[228] Pierluigi Paganini, 'ZITiS is the new German Government cyber unit in wake of terror attacks (*Security Affairs*, 15 August 2016) <https://securityaffairs.co/wordpress/50297/terrorism/zitis-german-cyber-unit.html> accessed 3 December 2019.

5 Conclusion

In the run towards the introduction of the Computer Crime III Act, many criticized the provisions establishing that law enforcement agencies will be allowed to hack into computer systems of suspects. These provisions are widely regarded as a response to the increasing use of encryption in many of the applications and hardware end-users have accustomed to their daily routines, and also something criminals have resorted to to create obstacles for police in uncovering their criminal intentions. Literally and figuratively speaking, this has led to the government trying to find holes in this approach to not end up in a so-called dark age in which police powers have been made redundant. Exploiting weaknesses in software and hardware, then, allows police to have a more specific 'hacking power' allowing it to overcome some of the hurdles introduced with this increased use of encryption by criminals.

Existing literature has mostly focused on the different (malware) tools police could potentially obtain to gain access to systems and whether the safeguards that are introduced are adequate enough to prevent misuse, or phrased differently, to prevent the creation of a super umbrella power. Little has been written however about the actual use of exploiting flaws in software and hardware in practice, more specifically about how to safeguard this process in order to prevent obstruction of evidence.

The research question in this thesis as to why (technical) safeguards to law enforcement exploiting vulnerabilities should be introduced to preserve the integrity of gathered evidence was answered by examining three different areas. First, the underlying processes of exploiting vulnerabilities were explored. Discussed in this regard is how vulnerabilities in software and hardware are errors in the code that make up these systems. Consequently, abusing these flaws means that attackers will need to find them which is done by a careful and extensive investigation or by resorting to the acquisition of vulnerabilities that have been found by others and are sold on so-called vulnerability markets. A description of some of the most common vulnerabilities concluded the first part of this research to demonstrate how a vulnerability is exploited after one has been found or purchased.

The second area of study in this research consisted of an analysis of the law that contains the provisions that allow for the exploiting of vulnerabilities by police. It focused on the legal measures that have been introduced and identified what this power actually allows for. This part of the research meant for a more critical examination to dive into some of the details behind the use of vulnerabilities by law enforcement agencies. Public security and proliferation

fundamentally dictated the narrative in this chapter as these two issues demonstrated what severe consequences can come with using vulnerabilities as a means to gain access to systems.

Finally, the last area of research elaborated on technical procedures that the current legal framework lacks. Most important, it describes how implementing such procedures is not only desirable from the standpoint of having appropriate guarantees, but mostly establishes that legally enforcing such procedures would preserve integrity and benefit the investigations of law enforcement agencies. A lack of a uniform approach to procedures regarding the handling of evidence after exploitation strengthens this need to prevent any compromise of the evidence. Furthermore, the issue of defendants perhaps needing to know about how law enforcement operates when it exploits a weakness has been touched upon which raises questions as to whether vulnerabilities should be fully disclosed during court proceedings or not.

One of the main findings of this research is that fundamentally exploiting vulnerabilities can be a very effective method to gain access to computer systems. However, one of the most important issues that the current legal framework seems to encounter is that this method is simply subsumed under the bigger notion of 'police hacking'. All the safeguards that have been introduced are not likely to have been written specifically with exploiting vulnerabilities in mind, but more so with running certain tools that allow for access in a much more automated way. Conducting technical reconnaissance, gathering suspect's devices and the networks they connect with, crosschecking all the software and hardware in use against existing vulnerabilities and so on focus much on an individual tailormade process that paradoxically holds implications for the masses when exposed. This study keeps finding that exploiting vulnerabilities should be written down in a separate provision and the safeguards that the law has introduced should be rewritten to address this power.

In sum, because the current legal framework is written in this fashion, exploiting vulnerabilities now exposes not only the public, but also law enforcement to many risks. Firstly, proliferation seems to not be adequately addressed and in the wake of cybersecurity attacks and data breaches occurring in great numbers, this issue remains to be alarming. Secondly, Article 126nba of the Dutch CCP seems to go past what vulnerability exploitation does to maintaining the integrity of data on a system and how it is almost inherent to the nature of this practice to result in integrity becoming compromised. There is a serious lack of research in this domain from a government perspective, but also in the legal scholarly field, concerning the creation of an adequate framework for how to deal with data and evidence after access has been gained by exploiting a flaw. And also, what does this do to the evidentiary position in terms of existing weaknesses that were not central to the investigation into a suspect but were merely

present in a specific device or application, as was the case with the SS7 vulnerability in the Telegram hack by the German BKA?

This answer to the research question does not ask for removal of this power from the toolbox of the police, but it asks for the legal framework to separately and in more detail address the exploitation of vulnerabilities. Government should arguably investigate deeper into how vulnerabilities in practice are actually exploited (i.e. often not the case of entering some details in an application and running that application) and desirably think of establishing a separate team that oversees the whole process of vulnerability exploitation. Under the current establishment, criminal suspects could take advantage of the holes legislation creates by not setting out more comprehensive procedures on how to deal with evidence gathered specifically through the exploitation of a flaw, or the other way around, police can utilise this same knowledge to take this power further than is necessary and abuse it. This study thus asks for more research on safeguards from a more technical nature that are aimed at preserving integrity to rule out any arbitrary (mis)use during investigations, and also during court procedures.

In conclusion, exploiting vulnerabilities is a necessary tool for law enforcement to gain access to computer systems which otherwise becomes extremely difficult in the current age. Nevertheless, practices worldwide, especially those in the United States, have illustrated how this power can create severe implications on public security. Some safeguards have been implemented to mitigate these risks as much as possible, however, as the focus has been primarily on almost solely these implications, there has not been much rhetoric on integrity and disclosure during court proceedings. As there is almost no way around using vulnerabilities, it is time to shift the discussion to evidence and court disclosure now before investigations will lose merit because of the inherent nature of vulnerabilities.

6 Bibliography

6.1 Literature

1. Abdullah M. Algarni, Yashwant K. Malaiya, 'Software Vulnerability Markets: Discoverers and Buyers' (2014) 8 International Journal of Computer, Information Science and Engineering 3
2. Alan Z. Rozenshtein, 'Wicked Crypto' (2018) 9 UC Irvine Law Review 1181.
3. Alana Maurushat, *Disclosure of Security Vulnerabilities: Legal and Ethical Issues* (Springer; 2013).
4. Apostolos P. Fournaris, Lidia P. Fraile, Odysseas. Koufopavlou, 'Exploiting Hardware Vulnerabilities to Attack Embedded System Devices: a Survey of Potent Microarchitectural Attacks' (2017) 6 (52) Electronics <https://www.mdpi.com/2079-9292/6/3/52/pdf> accessed 23 November 2019.
5. Arthur Conklin, Daniel Shoemaker, *CSSLP Certification All-in-One Exam Guide* (McGraw-Hill Education 2013).
6. Arthur Conklin, Gregory White, *Principles of Computer Security: CompTIA Security+ and Beyond* (McGraw-Hill Education 2018).
7. Aswin Gopalakrishnan, Emanuele Vineti, Ashok Kumar Mohan, M. Sethumadhavan, 'The Art of Piecewise Hashing: A Step Toward Better Evidence Provability' (2018) 7 Journal of Cyber Security and Mobility 1-2 <https://www.riverpublishers.com/journal/journal_articles/RP_Journal_2245-1439_719.pdf>, accessed 2 November 2019.
8. Bart Jacobs, 'Policeware' (2009) 39 Nederlands Juristenblad.
9. Bruce Duyshart, *The Digital Document: A Reference for Architects, Engineers and Design Professionals* (Routledge 2013).
10. Bruce Schneier, *Click Here to Kill Everybody: Security and Survival in a Hyper-connected World* (W. W. Norton & Company 2018)
11. Bruce Schneier, *We Have Root: Even More Advice from Schneier on Security* (John Wiley & Sons, Inc. 2019)
12. The Center for Internet and Society, 'Government Hacking: Evidence & Court Disclosure of Vulnerabilities' (*The Center for Internet and Society*, 5 April 2017) <https://cyberlaw.stanford.edu/events/government-hacking-evidence-court-disclosure-vulnerabilities> accessed 14 November 2019.

13. C.P.M. Cleiren, M.J.M. Verpalen, *Strafvordering. Tekst & Commentaar. De tekst van het Wetboek van Strafvordering en enkele aanverwante wetten voorzien van commentaar* (Kluwer 2011)
14. Chen-Yu Li, Chien-Cheng Huang, Feipei Lai, San-Liang Lee, Jingshown Wu, 'A Comprehensive Overview of Government Hacking Worldwide' (2018) IEEE Access 18174826 <https://ieeexplore-ieee-org.tilburguniversity.idm.oclc.org/document/8470931> accessed 1 November 2019.
15. Chwan-Hwa Wu, J. David Irwin, *Introduction to Computer Networks and Cybersecurity* (CRC Press 2013).
16. Ciprian Rusen, *IC3: Internet and Computing Core Certification Computing Fundamentals Study* (Sybex 2015).
17. Curtis Franklin Jr. BC, *Securing the Cloud: Security Strategies for the Ubiquitous Data Center* (Auerbach Publications 2019).
18. Eelco M. Moerman, 'Inburgeren in de opsporing: Over de juridische positie van de burger in de opsporing van strafbare feiten' (2016) Erasmus University Rotterdam <https://repub.eur.nl/pub/94119/Proefschrift-Moerman_.pdf> accessed 14 November 2019.
19. Enrico Perla, Massimiliano Oldani, *A Guide to Kernel Exploitation: Attacking the Core* (Syngress 2010).
20. Garth O. Bruen, *WHOIS Running the Internet: Protocol, Policy, and Privacy* (Wiley 2015).
21. Gedare Bloom, Eugen Leontie, Bhagirath Narahari and Rahul Simha, *Handbook on Securing Cyber-Physical Critical Infrastructure* (Morgan Kaufmann 2012).
22. Gerard Johansen, Lee Allen, Tedi Heriyanto, Shakeel Ali, *Kali Linux 2 – Assuring Security by Penetration Testing* (Packt Publishing 2016).
23. Giorgio Franceschetti, Marina Grossi, *Homeland Security Technology Challenges: From Sensing and Encrypting to Mining and Modeling* (Artech House Publishers 2008).
24. Harold F. Tipton, Micki K. Nozaki, *Information Security Management Handbook* (CRC Press 2007).
25. Huib Modderkolk, *Het is oorlog, maar niemand die het ziet* (Podium Amsterdam 2019).
26. ITL Education Solutions Limited, *Introduction to Computer Science* (Pearson Education India 2004).

27. Ivan Škorvánek, Bert-Jaap Koops, Bryce Clayton Newell, and Andrew Roberts, "'My Computer Is My Castle': New Privacy Frameworks to Regulate Police Hacking' (2019) Pre-Press Draft, forthcoming in BYU Law Review.
28. Jairo E. Serrano, Juan Carlos Martínez-Santos, *Advances in Computing: 13th Colombian Conference, CCC 2018, Cartagena, Colombia, September 26-28, 2018, Proceedings* (Springer 2018)
29. James Graham, Ryan Olson, Richard Howard, *Cyber Security Essentials* (Routledge 2010)
30. James Forshaw, *Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation* (No Starch Press 2017).
31. Jason Andress, *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice* (Syngress 2011).
32. Jason Healey, 'The U.S. Government and Zero-Day Vulnerabilities: From Pre-Heartbleed to Shadow Brokers' (2016) Journal of International Affairs <https://jia.sipa.columbia.edu/sites/default/files/attachments/Healey%20VEP.pdf> accessed 21 September 2019.
33. Jidong Xiao, Hai Huang, Haining Wang, *Security and Privacy in Communication Networks: 11th EAI International Conference, SecureComm 2015, Dallas, TX, USA, October 26-29, 2015, Proceedings* (Springer International Publishing 2015).
34. Jordi Mongay Batalla, George Mastorakis, Constandinos X. Mavromoustakis, Evangelos Pallis, *Beyond the Internet of Things: Everything Connected* (Springer 2017).
35. Joseph M. Kizza, *Guide to Computer Network Security* (Springer 2015).
36. Kim Zetter, *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon* (Broadway Books 2015).
37. Kristin Finklea, 'Law Enforcement Using and Disclosing Technology Vulnerabilities' (2017) (Congressional Research Service R44827 <https://fas.org/sgp/crs/misc/R44827.pdf> accessed 9 June 2019.
38. Lech J. Janczewski, Andrew M. Colarik, *Cyber Warfare and Cyber Terrorism* (Information Science Reference 2019).
39. Lillian Ablon, Martin C. Libicki, Andrea A. Golay, *Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar* (Santa Monica, CA: RAND Corporation, 2014).
40. Martin C. Libicki, Tim Webb, *The Defender's Dilemma: Charting a Course Toward Cybersecurity* (RAND Corporation 2015)

41. Michael E. Whittman, Herbert J. Mattord, David Mackey, Andrew Green, *Guide to Network Security* (Cengage Learning 2012).
42. Mirja Gutheil, Aurélie Heetman, James Eager, Max Crawford, Quentin Liger, 'Legal Frameworks for Hacking by Law Enforcement: Identification, Evaluation and Comparison of Practices' (2017) Policy Department for Citizens' Rights and Constitutional Affairs <http://www.europarl.europa.eu/RegData/etudes/STUD/2017/583137/IPOL_STU(2017)583137_EN.pdf> accessed 11 September.
43. Nancy Sehgal, *Let's Log In: A Textbook for Introductory Information Technology* (Dorling Kindersly (India) Pvt. Ltd. 2006).
44. Paul A.M. Mevis, Joost H.J. Verbaan, Barbara A. Salverda, 'Onderzoek aan in beslag genomen elektronische gegevensdragers en geautomatiseerde werken ten behoeve van de opsporing en vervolging van strafbare feiten' (2016) Erasmus Universiteit Rotterdam -School of Law, WODC < https://www.wodc.nl/binaries/2598-volledige-tekst_tcm28-74084.pdf> accessed 2 November 2019
45. Riana Pfefferkorn, 'Security Risks of Government Hacking' (2018) The Center for Internet and Society <http://cyberlaw.stanford.edu/files/publication/files/2018.09.04_Security_Risks_of_Government_Hacking_Whitepaper.pdf> accessed 21 September 2019
46. Sagar Rahalkar, *Metasploit for Beginners: Create a threat-free environment with the best-in-class tool* (Packt Publishing 2017).
47. Sean Oriyano, *Penetration Testing Essentials* (Sybex 2016).
48. Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, 'Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet' (2014) 12 Nw. J. Tech. & Intell. Prop. 1.
49. Sven Herpig, 'Government Hacking: Computer Security vs. Investigative Powers paper' (*Stiftung Neue Verantwortung*, 21 June 2017) <https://www.stiftung-nv.de/sites/default/files/snv_tcf_government_hacking-problem_analysis_0.pdf> accessed 4 November 2019.
50. The New York Times Editorial Staff, *Hacking and Data Privacy: How Exposed Are We?* (The Rosen Publishing Group 2018).
51. Trent Jaeger, *Operating System Security* (Morgan and Claypool Publishers 2008).
52. Zach Lerner, 'A Warrant to Hack: An Analysis of the Proposed Amendments to Rule 41 of the Federal Rules of Criminal Procedure' (2016) 18 Yale J.L. & Tech 1.

53. Zhen Yan, Refik Molva, Wojciech Mazurczyk, Raimo Kantola, *Network and System Security: 11th International Conference, NSS 2017, Helsinki, Finland, August 21–23, 2017, Proceedings (Lecture Notes in Computer Science)* (Springer 2017).

6.2 Jurisprudence

54. *Jasper v. United Kingdom* App no 27052/95 (ECHR, 16 February 2000)
55. Gerechtshof 's-Hertogenbosch 8 december 2006, LJN AZ4219 (Gimli)

6.3 Legislation

56. Dutch Code of Criminal Procedure
57. German Code of Criminal Procedure (StPO)
58. *Handelingen TK 2016/17*, no 34, item 26.
59. Investigatory Powers Act 2016, Code of Practice, Equipment Interference (2018).
60. *Kamerstukken II 2015/16, 34 372,* no. 3
61. *Kamerstukken II 2016-17, 34 372,* no. 14.
62. *Kamerstukken II 2017/18, 34 372,* no. G.
63. *Staatsblad* 2018, 340.
64. Wet op de Computer Criminaliteit III
65. 'Position paper Fox-IT t.b.v. hoorzitting/rondetafelgesprek Computercriminaliteit III d.d. 11 februari 2016' (2016) Tweede Kamer 2016D06073 <https://www.tweedekamer.nl/downloads/document?id=19cb6a18-cb5f-4a39-8cf9-1defac7920ae&title=Position%20paper%20Fox-IT%20t.b.v.%20hoorzitting%2Frondetafelgesprek%20Computercriminaliteit%20III%20d.d.%2011%20februari%202016.pdf> accessed 24 September 2019.

6.4 Other sources

66. Anouk Vleugels, 'Police can remotely drive your stolen Tesla into custody' (*The Next Web*, 19 November 2018) <https://thenextweb.com/the-next-police/2018/11/19/police-control-your-self-driving-cars/> accessed 24 September 2019.
67. Andy Greenberg, 'Meet the Hackers Who Sell Spies The Tools To Crack Your PC (And Get Paid Six-Figure Fees)' (*Forbes*, 21 May 2012) <https://www.forbes.com/sites/andygreenberg/2012/03/21/meet-the-hackers-who-

sell-spies-the-tools-to-crack-your-pc-and-get-paid-six-figure-fees/> accessed 20 July 2019.

68. Andy Greenberg, 'Triple Meltdown: How So Many Researchers Found a 20-Year-Old Chip Flaw at the Same Time (*Wired*, 1 July 2018) <https://www.wired.com/story/meltdown-spectre-bug-collision-intel-chip-flaw-discovery/> accessed 24 November 2019.
69. Andy Greenberg, 'The Strange Journey of an NSA Zero-Day—Into Multiple Enemies' Hands' (*WIRED*, 7 May 2019) <https://www.wired.com/story/nsa-zero-day-symantec-buckeye-china/> accessed 8 September 2019.
70. Andy Greenberg, 'Meltdown Redux: Intel Flaw Lets Hackers Siphon Secrets from Millions of PCs' (*WIRED*, 14 May 2019) <https://www.wired.com/story/intel-mds-attack-speculative-execution-buffer/> accessed 25 June 2019.
71. Axi0mX, 'ipwndfu' (*Github*, 1 October 2019) <https://github.com/axi0mX/ipwndfu> accessed 24 November 2019.
72. Bernardo Lustosa, "Apple's iOS update frequency has increased 51% under Cook's management" (*VentureBeat*, 28 February 2018) <https://venturebeat.com/2018/02/28/apples-ios-update-frequency-has-increased-51-under-cooks-management/> accessed 24 November 2019.
73. Bill Horne, 'Hardware Security: Why Fixing Meltdown & Spectre Is So Tough' (*Dark Reading* 26 January 2018) <https://www.darkreading.com/risk/hardware-security-why-fixing-meltdown-and-spectre-is-so-tough/a/d-id/1330908> accessed 25 June 2019.
74. Bill Marczak, John Scott-Railton, 'The Million Dollar Dissident - NSO Group's iPhone Zero-Days used against a UAE Human Rights Defender' (*Citizenlab*, 24 August 2016) <https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/> accessed 21 September 2019.
75. Bruce Schneier, 'WannaCry and Vulnerabilities' (*Schneier on Security*, 2 June 2016) <https://www.schneier.com/blog/archives/2017/06/wannacry_and_vu.html> accessed 9 June 2019.
76. Bundesministerium des Innern für Bau und Heimat, 'Startschuss für ZITiS' (*Bundesministerium des Innern für Bau und Heimat*, 21 January 2017) <https://www.bmi.bund.de/SharedDocs/pressemitteilungen/DE/2017/01/zitis-vorstellung.html> accessed 14 November 2019.

77. Buro Janssen, ‘Gamma Group en de politie; FinFisher trojan in de Nationale politie’ (*Buro Janssen*, 22 January 2017) <https://www.burojansen.nl/observant/gamma-group-en-de-politie-finfisher-trojan-in-de-nationale-politie/> accessed 22 September 2019.
78. Buro Janssen, ‘De Nederlandse politie en Hacking Team - Flirten met de tools van de dictator’ (*Buro Janssen*, 23 January 2017) <https://www.burojansen.nl/observant/de-nederlandse-politie-en-hacking-team-flirten-met-de-tools-van-de-dictator/> accessed 21 September 2019.
79. Cat Rutter Pooley, ‘Cyber security efforts turn proactive after sophisticated attacks’ (*Financial Times*, 15 November 2018) < https://www.ft.com/content/68a9398a-d065-11e8-9a3c-5d5eac8f1ab4> accessed 9 June 2019.
80. CB Insights, ‘How Big Tech Is Finally Tackling Cybersecurity’ (*CB Insights*, 27 March 2019) < https://www.cbinsights.com/research/facebook-amazon-microsoft-google-apple-cybersecurity/> accessed 9 June 2019.
81. CERT-EU, ‘WannaCry Ransomware Campaign Exploiting SMB Vulnerability’ (2017) CERT-EU Security Advisory 2017-012 <https://cert.europa.eu/static/SecurityAdvisories/2017/CERT-EU-SA2017-012.pdf> accessed 20 November 2019.
82. CHEF-KOCH, ‘How to Hack Whatsapp & Telegram Using SS7 Flaw’, (Github, 2015) <https://gist.github.com/CHEF-KOCH/07ad6b8d3cd3d11435cc6dffb7b33d85> accessed 2 November 2019.
83. Dan Sabbagh, ‘Israeli firm linked to WhatsApp spyware attack faces lawsuit’ (18 May 2019) <https://www.theguardian.com/world/2019/may/18/israeli-firm-nso-group-linked-to-whatsapp-spyware-attack-faces-lawsuit> accessed 21 September 2019.
84. Daniel Miessler, ‘Public Vulnerability Database Resources’ (*Daniel Miessler*, 5 December 2018) <https://danielmiessler.com/study/vulnerability-database-resources/> accessed 3 July 2019.
85. Dawn Kawamoto, “’WannaCry’ Rapidly Moving Ransomware Attack Spreads to 74 Countries” (*Dark Reading*, 5 December 2017) <https://www.darkreading.com/attacks-breaches/wannacry-rapidly-moving-ransomware-attack-spreads-to-74-countries/d/d-id/1328874> accessed 20 November 2019.
86. Ellen Nakashima, ‘FBI paid professional hackers one-time fee to crack San Bernardino iPhone’ (*Washington Post*, 12 April 2016) <https://www.washingtonpost.com/world/national-security/fbi-paid-professional-

hackers-one-time-fee-to-crack-san-bernardino-iphone/2016/04/12/5397814a-00de-11e6-9d36-33d198ea26c5_story.html> accessed 21 September 2019.

87. Eric Geller, 'FBI says it won't submit tool used to hack San Bernardino iPhone for disclosure review' (*The Daily Dot*, 27 April 2016) <https://www.dailydot.com/layer8/fbi-san-bernardino-iphone-exploit-no-vulnerability-disclosure/> accessed 21 September 2019.
88. Gordon Haff, 'Why the operating system matters even more in 2017' (*opensource*, 7 December 2016) <https://opensource.com/16/12/yearbook-why-operating-system-matters> accessed 27 June 2019.
89. IBM, 'IBM 2015 Cyber Security Intelligence Index' (*IBM*, 2016) <https://essextec.com/wp-content/uploads/2015/09/IBM-2015-Cyber-Security-Intelligence-Index_FULL-REPORT.pdf> accessed 24 September 2019.
90. Jack Nicas, Apple to Close iPhone Security Hole That Law Enforcement Uses to Crack Devices (*NY Times*, 13 June 2018) <https://www.nytimes.com/2018/06/13/technology/apple-iphone-police.html> accessed 20 July 2019.
91. James B. Comey, 'Going Dark: Are Technology, Privacy, and Public Safety on a Collision Course?' (*FBI*, 16 October 2014) <https://www.fbi.gov/news/speeches/going-dark-are-technology-privacy-and-public-safety-on-a-collision-course> accessed 9 June 2019.
92. Joseph Menn, Mark Hosenball, 'Apple iPhone unlocking maneuver likely to remain secret' (*Reuters*, 14 April 2016) <https://www.reuters.com/article/us-apple-encryption-whitehouse-idUSKCN0XB05D?feedType=RSS&feedName=technologyNews&utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+reuters%2FtechnologyNews+%28Reuters+Technology+News%29> accessed 21 September 2019.
93. Inside Battelle, 'Hardware vs. Software Vulnerabilities' (*Inside Battelle*, 18 January 2018) <https://inside.battelle.org/blog-details/hardware-vs.-software-vulnerabilities> accessed 25 June 2019.
94. Jenna McLaughlin, 'NSA Looking To Exploit Internet of Things, Including Biomedical Devices, Official Says' (*The Intercept*, 10 June 2016) <https://theintercept.com/2016/06/10/nsa-looking-to-exploit-internet-of-things-including-biomedical-devices-official-says/> accessed 24 September 2019.

95. Joseph Cox, ‘Your Government's Hacking Tools Are Not Safe’ (*Motherboard*, 14 April 2017) <https://www.vice.com/en_us/article/d7bvxa/your-governments-hacking-tools-are-not-safe> accessed 24 September 2019.
96. Josh Fruhlinger, ‘What is Stuxnet, who created it and how does it work?’ (*CSO*, 22 August 2017) <https://www.csoonline.com/article/3218104/what-is-stuxnet-who-created-it-and-how-does-it-work.html> accessed 9 June 2019.
97. Josh Fruhlinger, ‘Spectre and Meltdown explained: What they are, how they work, what’s at risk’ (*CSO* 15 January 2018) <https://www.csoonline.com/article/3247868/spectre-and-meltdown-explained-what-they-are-how-they-work-whats-at-risk.html> accessed 25 June 2019.
98. Kevin Poulsen, ‘Feds Are Suspects In New Malware That Attacks TOR Anonymity’ (*WIRED*, 8 May 2013) <https://www.wired.com/2013/08/freedom-hosting/> accessed 9 June 2019.
99. Lorenzo Franceschi-Bicchierai, ‘The Vigilante Who Hacked Hacking Team Explains How He Did It’ (*Motherboard*, 15 April 2016) <https://www.vice.com/en_us/article/3dad3n/the-vigilante-who-hacked-hacking-team-explains-how-he-did-it> accessed 21 September 2019.
100. Lorenzo Franceschi-Bicchierai, “Hackers Hit ‘Some’ Cisco Customers With Leaked NSA Hacking Tools” (*Motherboard*, 19 September 2016) <https://www.vice.com/en_us/article/3dajyn/hackers-hit-cisco-customers-leaked-nsa-hacking-tools-shadow-brokers> accessed 24 September 2019.
101. Matt Blaze, ‘When Should the Government Disclose "Stockpiled" Vulnerabilities?’ (*Matt Blaze*, 2017) <https://www.mattblaze.org/blog/between_immediately_and_never/> accessed 15 July 2019.
102. Michael Sulmeyer, Kate Miller, ‘Indicting Hackers and Known Vulnerabilities’ (*Lawfare*, 27 May 2016) <https://www.lawfareblog.com/indicting-hackers-and-known-vulnerabilities> accessed 15 July 2019.
103. Microsoft, ‘MS09-042: Vulnerability in Telnet could allow remote code execution’ (*Microsoft*, 17 April 2018) <https://support.microsoft.com/en-us/help/960859/ms09-042-vulnerability-in-telnet-could-allow-remote-code-execution> accessed 24 November 2019.

104. National Vulnerability Database, 'CVE-2019-11931' (*National Vulnerability Database*, 14 November 2019) <https://nvd.nist.gov/vuln/detail/CVE-2019-11931> accessed 14 November 2019.

105. Nicolas Falliere, Liam O. Murchu & Eric Chien, 'W32.Stuxnet Dossier' (*Symantec Security Response*, February 2011) <https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf> accessed 1 November 2019

106. Nicole Perlroth & Scott Shane, 'In Baltimore and Beyond, a Stolen N.S.A. Tool Wreaks Havoc (*NY Times*, 25 May 2019) <https://www.nytimes.com/2019/05/25/us/nsa-hacking-tool-baltimore.html> accessed 9 June 2019.

107. NSA, 'Discovering IT Problems, Developing Solutions, Sharing Expertise' (*NSA*, 30 October 2015) <https://www.nsa.gov/News-Features/News-Stories/Article-View/Article/1625787/infographic-discovering-it-problems-developing-solutions-sharing-expertise/> accessed 20 July 2019.

108. Olaf van Miltenburg, 'Overheid gaat gebruik zero-days door AIVD en MIVD toetsen' (*Tweakers*, 15 March 2018) <https://tweakers.net/nieuws/136335/overheid-gaat-gebruik-zero-days-door-aivd-en-mivd-toetsen.html> accessed 23 September 2019.

109. Paulo Garcia, 'Don't trust your hardware: Why security vulnerabilities affect us al' (*The Conversation*, 1 November 2018) <https://theconversation.com/dont-trust-your-hardware-why-security-vulnerabilities-affect-us-all-105773> accessed 25 June 2019.

110. Pierluigi Paganini, 'ZITiS is the new German Government cyber unit in wake of terror attacks (*Security Affairs*, 15 August 2016) <https://securityaffairs.co/wordpress/50297/terrorism/zitis-german-cyber-unit.html> accessed 3 December 2019.

111. Rejo Zenger, 'Onbekende kwetsbaarheden als disruptive technology' (*Bits of Freedom*, 5 July 2017) <https://www.bitsoffreedom.nl/2017/07/05/onbekende-kwetsbaarheden-als-dispruptive-technology/> accessed 10 September 2019.

112. Robert Donovan, 'Are Some Security Vulnerabilities Too Complex to Fix?' (*InfoSecurity Magazine*, 28 May 2019) <https://www.infosecurity-magazine.com/infosec/security-vulnerabilities-1-1-1-1/> accessed 25 June 2019.

113. Robert K. Knake, 'FBI to Apple: We Would Probably Disclose the iPhone Flaw if We Knew What It Was' (*Council on Foreign Relations*, 29 March 2016) <https://www.cfr.org/blog/fbi-apple-we-would-probably-disclose-iphone-flaw-if-we-knew-what-it-was> accessed 21 September 2019.

114. Rupert Goodwins, 'Sony hacked again in Lulzsec breach' (*ZDNet*, 3 June 2011) <https://www.zdnet.com/article/sony-hacked-again-in-lulzsec-breach/> accessed 25 July 2019.

115. Scott Shane Nicole Perlroth, David E. Sanger, 'Security Breach and Spilled Secrets Have Shaken the N.S.A. to Its Core' (*NY Times*, 12 November 2017) <https://www.nytimes.com/2017/11/12/us/nsa-shadow-brokers.html> accessed 8 September 2019.

116. Swati Khandelwal, 'Facebook Sues Israeli NSO Spyware Firm for Hacking Whatsapp Users' (*The Hacker News*, 29 October 2019) <https://thehackernews.com/2019/10/whatsapp-nso-group-malware.html> accessed 14 November 2019.

117. Symantec, 'Regin: Top-tier espionage tool enables stealthy surveillance' (*Symantec*, 27 August 2015) <https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/regin-top-tier-espionage-tool-15-en.pdf> accessed 19 November 2019

118. Thomas P. Bossert, "It's official: North-Korea Is Behind WannaCry" (*WSJ*, 18 December 2017) <https://www.wsj.com/articles/its-official-north-korea-is-behind-wannacry-1513642537> accessed 20 November 2019.

119. Thomas Reed, 'New iOS exploit checkm8 allows permanent compromise of iPhones' (*Malwarebytes*, 27 September 2019) <https://blog.malwarebytes.com/mac/2019/09/new-ios-exploit-checkm8-allows-permanent-compromise-of-iphones/> accessed 24 November 2019.

120. Valarie Findlay, 'Cyber Threats Against Police' (*National Police Foundation*) <https://www.policefoundation.org/cyber-threats-a-glocal-problem-for-law-enforcement/> accessed 25 November 2019.

121. Violet Blue, 'How spyware peddler Hacking Team was publicly dismantled' (*Engadget*, 9 July 2015) <https://www.engadget.com/2015/07/09/how-spyware-peddler-hacking-team-was-publicly-dismantled/?guccounter=2> accessed 21 September 2019.

122. WebIMX, ‘Penetration Testing’ (*WebIMX*) <http://www.webinfomatrix.com/penetration-testing.html> accessed 5 July 2019.

123. Wikileaks, ‘Branch Direction Meeting notes’ (*Wikileaks*, 23 October 2014) <https://wikileaks.org/ciav7p1/cms/page_13763790.html> accessed 24 September 2019.

124. Zack Whittaker, ‘Two years after WannaCry, a million computers remain at risk’ (*TechCrunch*, 12 May 2019) < https://techcrunch.com/2019/05/12/wannacry-two-years-on/> accessed 9 June 2019.

125. Zak Doffman, ‘Apple Fixes Serious iOS 13, iPadOS 13 And Catalina Security Issues: Update Your Devices Now’ (*Forbes*, 31 October 2019) <https://www.forbes.com/sites/zakdoffman/2019/10/31/apple-patches-serious-ios-13-and-catalina-security-issues-update-your-devices-now/#63c8c8992c2a> accessed 25 November 2019.