Exploring the practice of organisational Security Patch Management from a socio-technical perspective

Using a Mixed Methods Approach to investigate IT-practitioners' decision-making and patch activity

Master thesis submitted to Delft University of Technology in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in **Complex Systems Engineering and Management**

Faculty of Technology, Policy, and Management

by

Yves van Engelen

Student number: 4680480

To be defended in public on August 25th 2022

Graduation committee

Chairperson	:	Prof.dr. M.J.G. van Eeten, Section Organisation & Governance
First supervisor	:	Dr. S.E. Parkin, Section Organisation & Governance
Second supervisor	:	Prof.dr.ir. M.F.W.H.A. Janssen, Section ICT

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Preface

This thesis research explores the practice of organisational security patch management from a socio-technical perspective by using a mixed methods approach to investigate IT-practitioners' decision-making and patch activity. With great pleasure, I present you the final report of this thesis, marking the ending of my Master in Complex Systems Engineering & Management at Delft University of Technology. The last six months have been an enriching experience where I was allowed to work with both industry professionals and academic experts.

While following the course on Economics of Cybersecurity given by my first supervisor Simon Parkin in the first semester of this academic year, I became more interested in the social and governance aspect of cybersecurity and was determined to find a research topic closely related to this. I would like to thank Simon Parkin for the valuable discussions and great supervision of the project. Furthermore, for always making time for me when I needed your perspective on something. Your enthusiasm and positivity throughout the project helped me stay focused and motivated. I would like to thank my chairperson Michel van Eeten for introducing me to the field of software security patching and helping me shape the context and scope of this research. Furthermore for your fruitful contributions of your extensive knowledge and expertise throughout the meetings. I would like to thank Marijn Janssen for helping me realise my full thesis committee and providing helpful feedback from a fresh and different perspective. I would like to express my gratitude and respect toward my entire thesis committee for their guidance and support along the way.

This work has been carried out with an external organisation, for which I would like to thank all IT practitioners who helped me with their useful insights and great cooperation throughout the project. Furthermore, I appreciate the help of two IT practitioners in particular, who helped me with the exploration of collecting quantitative data, which has not been easy. Additionally, a special thanks to my point of contact in the organisation for providing the utmost assistance and for your genuine enthusiasm throughout the project. I hope the findings of this thesis will provide valuable insights for future security patch processes.

Lastly, I would like to express my appreciation to my family and friends for supporting me throughout the process, helping me take my mind off it with the well-needed distractions, and checking in on how I was doing. I want to thank my parents and sister for their unfailing support and continuous encouragement during all my study endeavours. Special thanks to Annie, with whom I have spent the last five years studying closely together, for your daily positivity and inspiration.

Yves van Engelen
The Hague, August 2022

Abstract

In the current digitalised society keeping assets secure is one of the most prominent challenges organisations face. In the ongoing arms race between attackers and defenders, software security patching is a well-recognised and effective strategy to mitigate vulnerabilities in software products. However, organisations struggle with the best practice to “patch early and often”, resulting in vulnerabilities in software being exposed for much longer than desired. Prior research indicates the socio-technical nature of this practice forms the core of delays in software patch management. Developing a deeper understanding of the decision-making of IT practitioners and what socio-technical factors play a role in this process allows organisations to address the ineffectiveness of their security patch process. The main research question in this explorative research is: *What socio-technical factors influence the effectiveness and timeliness of the security patching process in organisations?* This Mixed Methods research combines qualitative data from interviews with IT practitioners, with a quantitative data exploration of the meaningfulness of organisational measurements. Findings show that IT practitioners go through a funnel of decision-making that influences the decision of what to patch, and when to patch. The presence and interplay of different socio-technical factors related to four main aspects of this decision (i.e., security, applicability, operability, and availability) result in tensions and trade-offs influencing the decision space of IT. Furthermore, this study indicates the interrelations between the significance of socio-technical factors, which is reduced by certain coping strategies applied by IT practitioners. This research reveals that having some measurement in place helps to understand the existence of challenges and the working of coping strategies, therefore contributing to an understanding of socio-technical challenges. However, it also reveals several limitations to the quality of existing data and difficulties in coming to measurements that provide meaningful information, due to socio-technical factors. The main contribution of this research is a better understanding of how socio-technical factors influence the decision-making process of IT practitioners. This research is limited in the way it uses quantitative data to understand patching activity. Future research is recommended to compare the potential discrepancy between what IT practitioners state influences the effectiveness of their security patch process and what the actual patching activity of IT practitioners reveals about the effectiveness of patching. This research furthermore hypothesises that not all socio-technical factors have the same level of significance. It is recommended to investigate the possibilities of quantification of the importance of each of the socio-technical challenges identified in this explorative study.

Table of Contents

Glossary

IT Practitioner – someone who practices and is specialised in IT, and is involved in making IT-related decisions through their daily work

IT System – “a discrete set of information resources organized expressly for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information” (NIST, 2022b, p. 1)

Security Patch – a piece of code that ‘fixes’ an identified problem in software programming.

Security Patch Management – “the process for identifying, acquiring, installing, and verifying patches for products and systems” (Souppaya & Scarfone, 2013, p. vi)

Socio-technical – “where human and technological interactions are tightly coupled, such that the success of software security patch management significantly depends on the effective collaboration of humans with the technical systems” (Dissanayake, Jayatilaka, Zahedi, & Babar, 2022, p. 2)

Software Security Patching – is the practice of installing fixes to security vulnerabilities in software products and systems deployed in an organisation's IT environment (Dissanayake et al., 2022)

Vulnerability – A weakness in an asset’s protections such that a threat source may be able to adversely affect the security requirements (confidentiality, integrity or availability) of an asset (Alexander & Panguluri, 2017)

List of Acronyms

ACSC	Australian Cyber Security Centre
AVG	Algemene Verordening Gegevensbescherming (NL)
BIR	Baseline Informatiebeveiliging Rijksdienst (NL)
BYOD	Bring Your Own Device
CAP	Change Advisory Board
CFS	Critical Success Factor
CIA-principle	Confidentiality, Integrity, Availability
CoSEM	Complex Systems Engineering and Management
CIS	The Center for Internet Security
CMMI	Capability Maturity Model Integration
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration score
DDoS	Denial of Service Attack
DPA	Dutch Data Protection Authority
DWP	UK's Department for Work & Pension
EDPB	European Data Protection Board
GDPR	General Data Protection Regulation
HA	High-Availability
HPC	High-performance Computer
IEC	International Electrotechnical Commission
ISO	International Organisation for Standardisation
IT	Information Technology
ISMS	Information Security Management System
ITIL	Information Technology Infrastructure Library
KPI	Key Performance Indicator
NCSC	The Dutch National Cyber Security Centre
NIST	National Institute of Standards and Technology
NVD	National Vulnerability Database
OS	Operating System
OTA	Ontwikkel (Develop), Test (Test), Acceptatie (Acceptation) (NL)
SA	System Administrator
SaaS	Software-as-a-Service
SANS	SysAdmin, Audit, Network and Security Institute
SLA	Service-level Agreement
TBS	Treasury Board of Canada Secretariat
UCISA	Universities and Colleges Information Systems Association (NL)
WODC	Wetenschappelijk Onderzoek- en Documentatiecentrum (NL)
WSUS	Windows Server Update Services

List of figures

List of tables

1 | INTRODUCTION

"While the issue of patch management has technology at its core, it's clear that focusing only on technology to solve the problem is not the answer" – Chan (2004)

This chapter aims to get a better understanding of the problem domain and research objective of this study. It aims to discuss the nature of the problem, why it needs to be addressed, and in what way. The motive for this research will be discussed in section 1.1. The research problem will be explored in more detail in section 1.2. Here, an overview of prior research is presented (1.2.1), the scientific and societal knowledge gaps are identified and the research's objective and scope are discussed (1.2.2), and the main research question and sub-questions are stated (1.2.3). The research relevance is argued for in section 1.3, including the link to the program of study this MSc Thesis is written for. Finally, section 1.4 presents a reading guide for the following chapters.

1.1 Research Background

"Cybercrime is the greatest threat to every company in the world", spoke IBM's former CEO Ginni Rommety in 2015 (Birch, 2015, par. 4). While eight years have passed, the claim has only become more substantial. An estimation of the worldwide company costs for cybercrime is set to be USD 10.5 trillion annually by 2025, which was *only* USD 3 trillion in 2015 (Morgan, 2020). Organisations of all sorts are attractive targets for threat actors to be harmed, varying from disruption of systems by denial-of-service attacks to financial gains by phishing attacks. Currently, it is getting increasingly easy to attack, e.g., malware can be bought off-the-shelf and used without any coding expertise (Eriksen-Jensen, 2013). At the same time, IT security practitioners face increasingly complex and more aggressive security threats (Eriksen-Jensen, 2013). This development, where it becomes easier and easier to attack while simultaneously the threats become more complex, is alarming.

Perhaps even more worrisome, according to a survey by McKinsey & Company, only 16% of risk managers state their company is well prepared to deal with a cyber risk (Poppensieker & Riemenschnitter, 2018). For companies, it is crucial to implement security measures and be well-prepared for potential threats. A security incident's consequences can impact asset and financial losses, productivity losses, reputation damage, and legal liabilities (Seemma, Nandhini, & Sowmiya, 2018). Dependent on the type of incident, the aftermath can last for weeks or even months. However, it is certainly not easy for organisations to keep their assets secure as their digital environment comprises multiple facets. Organisations increasingly depend on software to process data, optimise workflows, or provide services to employees. As IT is often a facilitating part of organisations' core businesses, third-party software and its maintenance is regularly used to bring in expertise and keep costs and other resources low. Although third-party software makes it easier for organisations to manage their IT systems, it is not the silver bullet to carefree operations. Within software products, defects, flaws, or glitches can be present that create opportunities for threat actors to exploit. With software becoming increasingly complex, the number of newly identified vulnerabilities is on the rise ((IBM, 2022), see Figure 1).

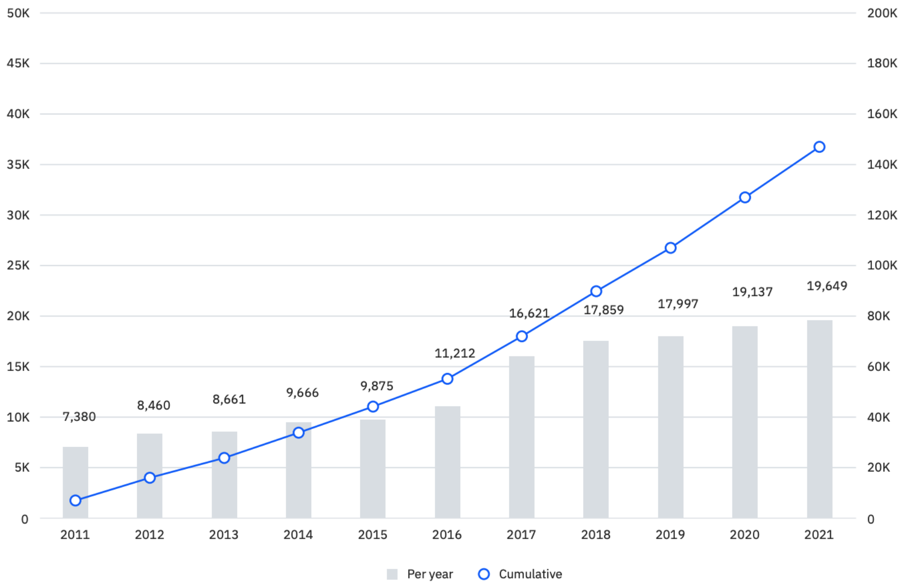


Figure 1 - New vulnerabilities identified each year (cumulative) (Figure from (IBM, 2022))

Vulnerabilities become a problem when threat actors use them to exploit organisations' IT systems. Over the course of 2021, multiple severe vulnerabilities were known to be exploited by threat actors. An example is the ‘ProxyLogon’ that allows threat actors to bypass the authentication of a Microsoft Exchange Server to impersonate an admin. This exploit could result in the extraction of sensitive data (IBM, 2022). Another example is the ‘Log4j’ vulnerability often used for web applications that allows threat actors to capture control of systems and infect them with malicious software (IBM, 2022). A more well-known example is the ‘WannaCry’ attack in 2017, which caused organisations to spend billions of dollars (USD) due to “productivity losses, mitigation efforts, paid ransom, and lost files” (Berr, 2017 in August, Dao, & Kim, 2019). The ‘success factor’ of the impact and size of the attack was made possible by the vast number of non-patched computer systems. The software vendor, Microsoft, released a patch to fix this vulnerability, which needed to be installed by the organisations themselves. However, even two months after the patch was released, the WannaCry attack “struck more than 200,000 computers across more than 150 countries that had not yet patched” (Greenberg, 2017; Lohr and Alderman, 2017 in August et al., 2019). Furthermore, one month and a significant amount of media attention later, the ‘NotPetya’ ransomware attack was exploiting organisations for the exact same vulnerability (Microsoft, 2017a in August et al., 2019).

This illustrates that the support of software vendors to have a securely functioning IT environment can only go so far, and many practices still need to be carried out by the organisation itself. One of the 14 best practices to be used by organisations to accomplish cyber resilience is to “ensure all software is up-to-date” (ENISA & CERT-EU, 2022). Keeping software up to date is also called *security patching* or *patch management*: the practice of installing fixes to security vulnerabilities in software products and systems deployed in an organisation’s IT environment (Dissanayake et al., 2022). Patching is a well-recognised and effective strategy to mitigate software vulnerabilities (Dissanayake et al., 2022). This process requires multiple steps, including identifying, acquiring, testing, installing, and verifying security patches (Dissanayake et al., 2022). In a survey administered by the Ponemon Institute to understand how organisations respond to software vulnerabilities, IT security professionals state that 60% of their breaches in 2019 involved vulnerabilities that were unpatched (Ponemon Institute, 2020). Besides, IBM states in their 2021 Cost of a Data Breach Report that to identify and contain a data breach, it took organisations on average 287 days (IBM, 2021).

The beforementioned examples and statistics signal the importance of accurate security patch management within organisations. Unfortunately, this turns out to be a difficult and complex task. The 2020 Cyber Hygiene Report[1] showed that more than half of organisations included in the study are not able to fix critical vulnerabilities within 72 hours after the patch is released (Automox & AimPoint Group, 2020). Furthermore, around 15% of organisations are unable to patch the vulnerabilities within 30 days (Automox & AimPoint Group, 2020). In a survey administered by Frost & Sullivan, 79% of IT security leaders state that they are "… "extremely" or "moderately" concerned that patches are missing from their endpoints" (Suby, 2018, p. 4). Another study states that the average time to patch a vulnerability is 102 days, while at the same time, the time it takes for attackers to weaponize a known vulnerability can be as less as seven days (Willis, 2020, par. 10). While timely patching is critical to keep their digital assets secure, organisations appear to struggle with the best practice to 'patch early and often' (Dissanayake et al., 2022). The question then emerges, why is this so difficult?

1.2 Problem definition

1.2.1 Prior Research

Investigating prior research (see Table 1) can help get a better understanding of that question. August et al. (2019) state that "the growing reality is that security is not [solely] a technical problem; it's an economic one" (p. 4576). The difficulty lies in the fact that security patching is like chasing a moving target (Eriksen-Jensen, 2013), wherein each of the activities, different socio-technical factors play a role that impact the decision-making of IT practitioners and challenge the timeliness of patching (Dissanayake et al., 2022). Werlinger, Hawkey, and Beznosov (2008)' categorisation of socio-technical factors that influence security patching is taken to analyse prior research, which consists of human, organisational, and technological challenges.

Table 1 - Overview of publications and authors used in explorative literature review

Author(s)	**Publication**
Andrew (2005)	The five Ps of patch management
August et al. (2019)	Market Segmentation and Software Security: Pricing Patching Rights
Cavusoglu, Cavusoglu, and Zhang (2006)	Economics of Security Patch Management
Dissanayake et al. (2022)	Software security patch management – A systematic literature review of challenges, approaches, tools and practices
Dissanayake, Zahedi, Jayatilaka, and Babar (2021)	A Grounded Theory of the Role of Coordination in Software Security Patch Management
Eriksen-Jensen (2013)	Holding back the tidal wave of cybercrime
Gerace and Mouton (2004)	The challenges and Successes of Implementing an Enterprise Patch Management Solution
Gianini, Cremonini, Rainini, Cota, and Fossi (2015)	A Game Theoretic approach to Vulnerability Patching
Li, Rogers, Mathur, Malkin, and Chetty (2019)	Keepers of the machines: Examining how system administrators manage software updates for multiple machines
Sihvonen and Jäntti (2010)	Improving Release and Patch Management Processes: An Empirical Case Study on Process Challenges
Tiefenau, Häring, Krombholz, and Von Zezschwitz (2020)	Security, Availability, and Multiple Information Sources: Exploring Update Behavior of System Administrators
Walkowski, Krakowiak, Oko, and Sujecki (2020).	Distributed Analysis Tool for Vulnerability Prioritisation in Corporate Networks
Werlinger et al. (2008)	Human, Organisational and Technological Challenges of Implementing IT Security in Organisations

[1] A survey of 560 IT operations and security professionals at enterprises with between 500 and 25,000 employees, across more than 15 industries

Starting with the *technical factors* of patching. Organisations face hundreds of patches within an average week (McGhie, 2003 in Cavusoglu et al., 2006). Every new patch requires multiple steps to be taken. Untested patches cannot be trusted as they might conflict with other parts of the system (Cavusoglu et al., 2006). This demonstrates the technical dependencies of software and hardware within an organisation (Dissanayake et al., 2021). Failures and side effects from patch deployment, such as faulty configuration and dependency changes, can lead to server downtime and service interruptions for employees (Dissanayake et al., 2022). Furthermore, it is possible that patches can break in unforeseen ways or that patches are not adequate for the intended vulnerability. As it is a utopia to always patch all vulnerabilities within an organisation (Andrew, 2005), organisations need to make trade-offs of which patch to apply and when. Besides, as not all vulnerabilities are targeted by threat actors, it would be a waste of resources to patch everything (Gianini et al., 2015). Often, this trade-off is based on risk predictions; vulnerabilities with the highest risk of being attacked will be patched first (Walkowski et al., 2020). Consequently, many IT practitioners often delay or refuse patches with low risks, causing the continuation of using outdated software (Dissanayake et al., 2021). Technical challenges, therefore, influence the way IT practitioners need to make decisions.

Likewise, *organisational factors* play a role in forming decisions during patching. Security is often not a priority in organisations, and a balance must be found between maintaining software security and complying with organisational policies to minimise service interruptions (Dissanayake et al., 2022). Li et al. (2019) found an emerging importance in patching decisions from organisation's policies and management. The authors state that a shift in company culture that recognizes the importance of patching would help benefit system administrators to carry out their job. Furthermore, security resources are often scarce. However, Werlinger et al. (2008) found that when support from top-level management is greater, the more is spent on preventive resources, and the more effective security is. Additionally, organisations, together with different external stakeholders (e.g., vendors and end-users), make security patching a collaborative effort (Dissanayake et al., 2022). This does not go without conflict, as opposing interests and interdependencies can be present (e.g., delays in patch release). Effective communication is needed to reach a mutual understanding (Werlinger et al., 2008). Besides, organisations often have an enormous number of connected devices within their networks that are constantly added, renamed, or removed; making patching difficult (Gerace & Mouton, 2004). Bring your own device (BYOD) policies make it even more challenging to control the devices connected to the organisational network (Eriksen-Jensen, 2013).

Moreover, *human factors* are influencing the security patching process. For example, a lack of expertise and knowledge can cause IT practitioners to rely heavily on third parties (Tiefenau et al., 2020). Automating parts of the process could help IT practitioners cope with the complexity (e.g., tools that monitor update processes). However, Dissanayake et al. (2022) argue that the 'human-in-the-loop' is inevitable in the dynamic environment. Manual deployment is more prone to human errors, potentially increasing the timeframe of patching (Li et al., 2019). Additionally, there are internal stakeholder dependencies, where each team member has its own role and responsibilities within the organisation, e.g., security managers, engineers, administrators. As Sihvonen and Jäntti (2010) argue, it is often unclear who is responsible for what and who needs to be informed at which stage of the process. Collaboration, coordination, and communication are essential for timely patching (Dissanayake et al., 2022).

1.2.2 Research Gap and Research Objective

It can be summarised that security patch deployment in organisations is time consuming and resource intensive. Moreover, the interplay of different socio-technical factors makes it troublesome, inconvenient, and ineffective. This results in ineffective patch deployment, making vulnerabilities exposed for a longer time than desired from a security point of view.

Looking at the *scientific gap*, current studies in the field of security patch management focus on a specific aspect of the problem, whether it is from a technical solution-oriented perspective on the possibilities of patch management automation of cloud applications (Hafeez, Karve, Dumba, Gandhi, & Zeng, 2019) or from a coordinative problem-oriented perspective on how the relationship between different teams is of great importance (Brandman, 2005). Tiefenau et al. recommend that "future work needs to take a more holistic view and investigate technical and social factors in the update process"(2020, p. 247). Looking at software security patch management from a socio-technical perspective is relatively new. In recent years, two significant publications with such a perspective came out: Dissanayake et al. (2021) and Dissanayake et al. (2022). Where Dissanayake et al. (2021) focus on the role of coordination, Dissanayake et al. (2022) conducted a systematic literature review of the current state of security patch management to identify social-technical challenges and frame directions for future research. One of these recommendations is the "need for more investigation on the less explored software security patch management phases" (2022, p. 13). Where most of the studies in the literature review focus on the patch deployment phase, little attention has been given to other phases such as information retrieval, testing, and post-deployment. Second, a "need for focus on socio-technical aspects in software security patch management" (2022, p. 14) is recommended. While their findings from existing literature reveal that the role of coordination, communication, and collaboration doubtlessly has a negative impact on the timeliness of vulnerability remediations, it is only limitedly studied. The authors argue for "a need for more research on the roles and effects of such socio-technical aspects in software security patch management" (2022, p. 14).

Looking at the *societal gap*, current studies lack "real-world, rigorous evaluations" in the form of field experiments and case studies. As the authors state: "the low percentage of the studies with industry-related evaluation highlights the need for researchers to work with practitioners to improve the state of the practice of rigorously evaluating research outcomes" (Dissanayake et al., 2022, p. 15). By actively collaborating with the industry, "the quality and transferability of the research outcomes to industrial adoption" (2022, p. 15) will be improved. Involving IT practitioners as data source has been suggested to be useful (e.g., (Dissanayake et al., 2021; Li et al., 2019)). In addition, the relevance and inclusion of 'contextual factors' is often lacking in current studies (Dissanayake et al., 2022; Tiefenau et al., 2020). When these are explicitly taken into account, the credibility and quality of the research will increase (Dissanayake et al., 2022). A one-size-fits-all strategy does not fit the complex environment organisations are in. Interesting to explore is the gap between theory and industry, and whether the discrepancies are as big as they seem to be.

These recommendations sum up the scientific and societal gap and lead to the *research objective* of developing a deeper understanding of security patching behaviour in organisations and explore what socio-technical factors play a role in the effectiveness of this process. To make valuable contributions to the existing research field, a different approach will be used that targets this multi-disciplinary gap. Where qualitative data (e.g., literature review, interviews) is needed to understand the interplay between socio-technical factors and the role of coordination, communication, and collaboration in decision-making, it can only come so far. As the actual deployment of a patch is merely technical and effectiveness is quantifiable, the usage of quantitative data is well suited to provide insights into patching activity. Therefore, the research strategy is a *Mixed Methods* approach, which is highly suitable when one type of data source is insufficient (Creswell & Clark, 2017). Mixed Methods research combines both qualitative and quantitative data in the same research (Creswell & Clark, 2017). Section 2.1 will dive deeper into the research design which explains the aim of comparing and synthesising both qualitative and quantitative data.

1.2.3 Research Questions and Scope

The main research question is formulated as follows:

> *What socio-technical factors influence the effectiveness and timeliness of the security patching process in organisations?*

Several sub-questions need to be answered first to answer the main research question. The flow of these sub-questions is presented in the next chapter (Chapter 2).

- **Sub-question 1**: What existing standards, frameworks, or guidelines on security patching are available for organisations?
- **Sub-question 2**: What decisions and trade-offs are being made by IT practitioners in organisational security patching, and what does this process look like?
- **Sub-question 3**: What is the quality of organisations' logs to determine patch activities?
- **Sub-question 4**: How does measurement of patching activity relate to the socio-technical factors causing the tensions and dependencies found in the patching decisions?
- **Sub-question 5**: How can the findings be used in practice, and how viable are the results to speed up an effective patching process?

1.3 Research Relevance

1.3.1 Scientific relevance

The scientific relevance aims to contribute to the socio-technical aspects of security patch management, a perspective that is only limitedly researched (Dissanayake et al., 2021). This research aims to better understand the decision-making process IT practitioners go through that influence the effectiveness of security patching. It aims to contribute to the understanding of how different socio-technical factors interact with this decision-making process, and whether the significance of these can be identified by the use of quantitative data on patch activity. Moreover, it aims to explore inter-team relations in the security patch process by conducting organisation-wide interviews. In addition to qualitative data, the possibilities for usage of quantitative data are explored, possibly encouraging further research to contribute to this type of research.

It furthermore aims to contribute to the exploration of contextual factors of educational organisations. To increase the credibility and transferability of the research outcomes, the scope of this research is focused on a large organisation (>1000 employees). The reason for this is two-fold, for one, large organisations and their networks are assumed to have a mature patch policy in place. This will increase the availability of data. Two, the role of coordination, communication, and collaboration ought to be more complex in large organisations, wherefore a deeper understanding of the socio-technical factors can be explored. The selection of this organisation will be further argued for in section 2.1.2.

1.3.2 Societal relevance

It is clear that ineffective patching leads to negative consequences in organisations. There is an urgency for this problem to be addressed, as can be concluded from the statistics in section 1.1: most organisations are nowhere near mature patch management practices. Important to mention is that the majority of organisations have the best intention to do patch management right; they however often do not know if what they are doing is 'right'. The aim is to develop an overview that shows what socio-technical factors play a role in organisational patch management for organisations to be used to reflect on their own practice. Where large organisations with a patch management process in place can use the outcomes to reflect on their existing practises, small and medium organisations without a patch management process in place can use the research outcomes to design or alter their process. Next to organisations, two other interest groups can benefit from this research. First, software vendors could use the findings to integrate a socio-technical aspect to the development process of patches (Dissanayake et al., 2022). With more knowledge of organisations' struggles, patch release and deployment alignment could be accomplished. Second, when organisations improve the effectiveness and timeliness of security patching, end-users presumably benefit from more secure services.

1.3.3 Link to CoSEM Study Programme

The study program of Complex Systems Engineering and Management (CoSEM) focuses on design solutions for large and complex socio-technical systems. Characteristics of these systems are that these are often systems of systems, embedded and connected to other systems, consisting of large-scale technical and physical components. Security patching management in organisations includes an extensive amount of hardware and physical IT networks that are embedded within bigger networks. Security is not only embedded in the larger range of tasks from an organisation’s IT department, but at the same time the organisation itself is embedded in the larger ecosystem. Moreover, the working of these systems requires coordination and cooperation in a dynamic environment between many different types of stakeholders with diverging interests and interdependencies. In security patch management, these can be seen internally between different teams and end-users of IT devices, but simultaneously externally with software vendors, third parties, and governmental organisations. To tackle these socio-technical problems, technical, institutional, economic, and social knowledge must be considered. As Islam et al. argue (in Dissanayake et al., 2022) the effectiveness of patching is the result of the interplay between technical challenges and constraints, institutional standards and policies, and the interactions of humans with technical solutions. It is a great example of having to make non-optimal choices due to the complexity of the decision-making process.

1.4 Reading Guide

This chapter discussed the problem scope and research gap that will be addressed throughout the following chapters. Chapter 2 will dive deeper into the methodology and different methods that will be used to answer the main research question and related sub-questions. A literature review in Chapter 3 will delineate the problem domain in more depth and will analyse the different stakeholders involved. Here, sub-question 1 will be answered by analysing the literature on guidelines and recommendations. Chapter 4 will dive deeper into the practice of security patch management and will explore difficulties and relevant factors by interpreting the qualitative data of interviews, thus answering sub-question 2. Chapter 5 will explore the possibilities of quantitative measurements and evaluate the potential challenges, answering sub-question 3. The results found in both Chapter 4 and Chapter 5 will be combined and compared in Chapter 6 to provide an answer on sub-question 4. Here recommendations will be presented based on the synthesis of the findings. Chapter 7 will discuss and reflect on the results and the novelty of this research, and with that answer sub-question 5. Here, the limitations of this research and recommendations for further research will be discussed. Finally, Chapter 8 will provide a conclusion on the main research question by combining the answers to all sub-questions.

2 | METHODOLOGY

"Most people are starting to realize that there are only two different types of companies in the world: those that have been breached and know it and those that have been breached and don't know it"
– Software and security expert Ted Schlein (Reynolds, 2014, p. 83)

Where Chapter 1 already hinted at the suitable approach to take in this research, the aim of this chapter is to provide justification of the design choices made and a clarification of the type of research to be conducted. Furthermore, as there is a need for "real-world, rigorous evaluations" this study will make use of a case study in collaboration with a large organisation, which will be discussed in this chapter. The final aim of this chapter is to give insight into how the research approach, the different methods, and the sub-questions are linked together.

Section 2.1 will discuss what mixed methods research entails and what design fits the research objective most. Furthermore the case study selection and type of organisation will be discussed in this section. Section 2.2 will explain the different methods that are needed to be able to answer the sub-questions. An overview of the research structure will be presented in section 2.3, where the different research steps and methods will be linked to the corresponding flow of sub-questions.

2.1 Research Design

2.1.1 Mixed Methods Research

Johnson, Onwuegbuzie, and Turner (2007) offer the following definition: "Mixed methods research is the type of research in which a researcher or team of researchers combines elements of qualitative and quantitative research approaches (e.g., use of qualitative and quantitative viewpoints, data collection, analysis, inference techniques) for the purposes of breadth and depth of understanding and corroboration" (2007, p. 123). The main benefit of this research approach is to use two different research approaches in a complementary way to develop an in-depth and in-breadth understanding of security patch management. This allows to make up for the weaknesses in both qualitative (e.g., subjective measures) and quantitative (e.g., lack of context) research.

Within mixed methods research, there are three core research designs (Creswell & Clark, 2017). In this study, the *convergent design* is used. Here the purpose of using both qualitative and quantitative data is to bring these together to combine and compare the results to understand the research problem better (Creswell & Clark, 2017). This is ideal when a more complete understanding of the problem is wanted, and to validate the results of one data source with the other (Creswell & Clark, 2017). These two data sources are concurrent but separated, in a way that "one does not depend on the results of the other" (p. 127). This is particularly useful as the different data types can explore different perspectives and underlying relationships of the same research subject (Fernandez & Azorin, 2011). Creswell and Clark (2017) describe four steps in convergent design (see Figure 2) Step 1 includes the design and collection of qualitative and quantitative data. The data analysis in Step 2 is also conducted separately, with each its own quantitative or qualitative method(s). The data is merged in Step 3, to interpret these together in Step 4 eventually.

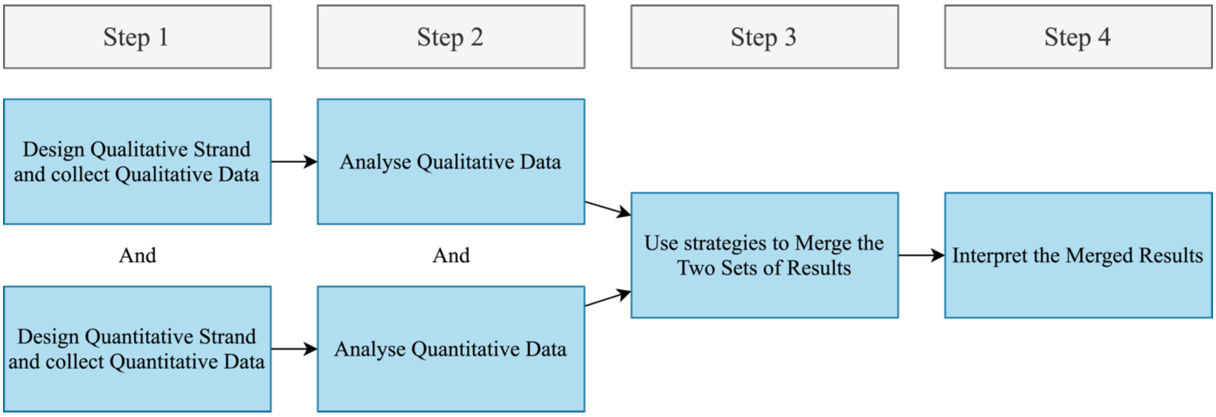

Step 1
Step 2
Step 3
Step 4
Design Qualitative Strand and collect Qualitative Data
Analyse Qualitative Data
And
And
Design Quantitative Strand and collect Quantitative Data
Analyse Quantitative Data
Use strategies to Merge the Two Sets of Results
Interpret the Merged Results

Figure 2 - Steps in convergent design (based on (Creswell & Clark, 2017, p. 127))

With any research method, limitations are present. A limitation of using a mixed methods approach is that it is more time-consuming, and it can be more complex to evaluate the outcomes than other approaches (Halcomb, 2019). For example, it is challenging to combine a qualitative data set in the form of text and a quantitative data set in the form of numbers (Creswell & Clark, 2017). These difficulties are reduced by putting the focus of exploration on qualitative data and use quantitative data to get a deeper understanding of a certain identified aspect. This does not limit the explorative nature of the quantitative data as both data sources are still used in a convergent manner, however it does make the research more manageable. Furthermore, as this research aims to explore the behaviour and the factors of influence of security patching, the potential challenge of combining qualitative and quantitative data will be a valuable finding in itself. Chapter 5 will discuss the quantitative data collection and analysis in more detail.

2.1.2 Case Study Selection

In 2019, Maastricht University (UM) was hit by a ransomware attack where the attackers were able to compromise 267 servers, including email servers, file servers, and backup servers (Dijkstra & van Dantzig, 2020). This led to the university paying the threat actors around €200,000 worth of bitcoin to restore their systems (Schouten & Bomers, 2021). The initial access to the university's network was gained through a phishing email, leading up to the compromission of servers on October 17th, 2019 (Dijkstra & van Dantzig, 2020). Investigations of Fox-IT show that the particular servers were running on a no longer supported operating system from Microsoft, where a certain MS17-0104 patch was not installed (Dijkstra & van Dantzig, 2020). Presumably, this patch would have fixed the possibility of the EternalBlue exploit, with which attackers could spread malware through the network (Dijkstra & van Dantzig, 2020). One of the recommendations provided by Fox-IT to the university after the event took place, is to improve the vulnerability and patch management processes.

Universities and educational organisations are providing a large number of digital facilities to their students and employees. Particularly interesting is that university networks are typically large and open, making them highly vulnerable (Singh, Joshi, & Gaud, 2016). Additionally, universities have a tendency toward decentralisation, making it more challenging for the central IT department to keep control over their network and to enforce policies (Al Maskari, Saini, Raut, & Hadimani, 2011). In a study by data analytics company Embroker (2022), it is shown what the threat environment looks like for the educational sector (see Figure 3). While DDoS attacks are surely the most common cyber incident, other types of attacks where the risk of occurrence could be reduced by proper security patching are also greatly present (e.g., Crimeware, Cyber-espionage, Web Applications, and Other).

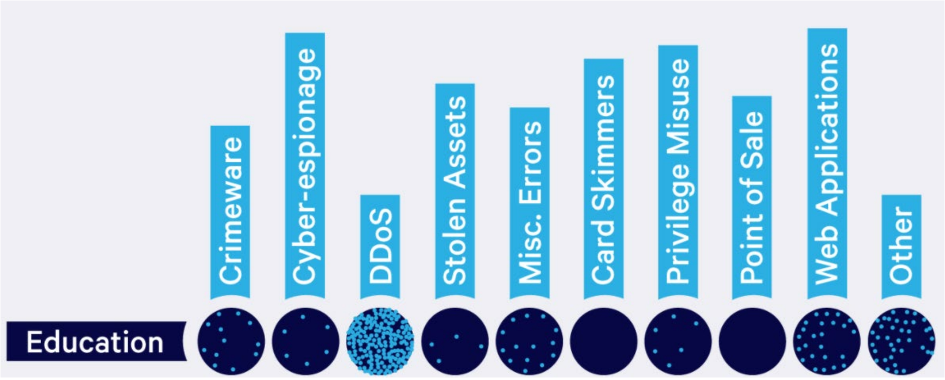


Figure 3 - Cyber Incidents in the education sector (Figure from (Embroker, 2022))

In this research, the partner organisation from which data will be collected is an educational organisation with over 30.000 students, researchers, and personnel. This case study allows for an in-depth, multi-faceted exploration of a complex issue in a real-life setting (Crowe et al., 2011). In the remaining of this proposal, *the organisation* will be referring to this educational organisation. The IT environment of this organisation is split roughly into two parts (see Figure 4). There are centralised IT systems that provide central services to students, researchers, and employees (e.g., email, digital learning environments, (virtual) computers). Different teams operate different parts of ICT, e.g., Systems, Infrastructure, Applications, Data Management. These different teams are all responsible for patching their own systems. Additionally, there are decentralised IT systems that are connected to IT's network but are owned by researchers of faculties. With the ownership of these systems come the decision rights as well, wherefore, these owners are also in charge of security patching. These systems are also referred to as *faculty-managed systems* in the remaining of this research.

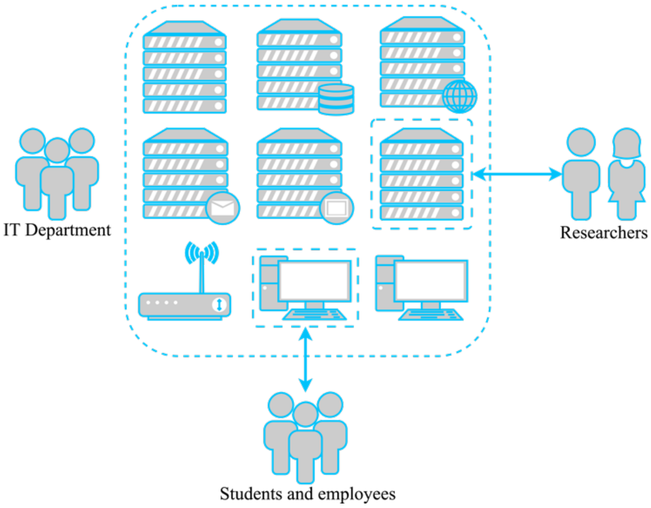
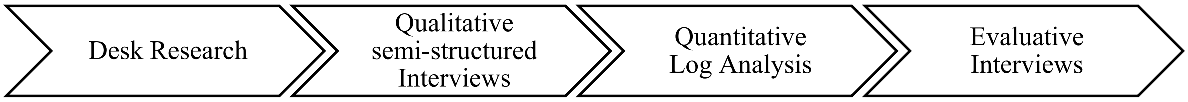


Figure 4 - Abstract overview of IT environment of case study organisation

2.2 Methods in sub-questions

Throughout this study, multiple research methods will be combined to provide a comprehensive analysis. In this section, the different methods will be discussed together with their aim and which sub-question they help to answer. Four main research methods will be used, as summarised in Figure 5.


Desk Research

Qualitative semi-structured Interviews

Quantitative Log Analysis

Evaluative Interviews

Figure 5 - Research methods

2.2.1 Desk research

> *Sub-question 1: What existing literature, standards, frameworks, or guidelines on security patching are available for organisations on security patch management?*

The first sub-question is descriptive and is focused on getting an overview of the aspects of the security patching process. The goal is to understand what according to theory effective patch management entails, and what is recommended for organisations to accomplish this. Hence *desk research* is the chosen research method for answering this sub-question. Here the secondary data gathered are scientific articles and grey literature (e.g., industry reports). The focus will lay on combining multiple different sources from standardisation, advisory, and governmental bodies (e.g., ISO, NIST, SANS), to get a better understanding of the existing guidance for organisations. The data will be clustered and analysed based on content. This sub-question has a scoping purpose as the deliverable (i.e., an overview of the information available for organisations) will be used as input for the following sub-question to analyse the patching process within the organisation. The findings are used to get an initial idea of what authoritative sources see as 'good security patch management' and what these recommend to acquire this.

2.2.2 Qualitative Interviews

> *Sub-question 2: What decisions and trade-offs are being made by IT practitioners in organisational security patching and what does this process look like?*

This sub-question is explorative and is focused on getting an overview of the decisions and trade-offs being made by IT practitioners during the patching process. The goal is to understand how decisions are made from different perspectives within the IT department, and whether certain themes can be identified. The research method for this sub-question is by conducting *semi-structured interviews* with 14 IT practitioners. Semi-structured interviews are a well-recognised method to explore thoughts and coping strategies in decision-making processes. All participants are employed at the same organisation but have different roles and responsibilities, and are part of different teams (e.g., Security officers, System Administrators, Coordinators, Managers). This allows for an in-breadth understanding of the interaction between different stakeholders within the IT department. The participants are selected through *purposeful sampling*; all participants are actively involved in decision-making with security patching, whether as advisor on the severity of a patch, change coordinator, or at an operational level of patch implementation. All respondents are approached via one employee, who serves as the point of contact with the organisation throughout the entire research period.

Semi-structured interviews will help to guide the interviewees in a certain direction (Flick, Von Kardorff, & Steinke, 2004). Questions that will be asked are mainly open-ended. However, some closed-ended questions are posed to open the opportunity for open-ended questions (Adams, 2015). For example, the question 'in your judgement, do you think security patching takes longer than needed?' can be followed up by 'why is that?' or 'when does that happen?'. Details can additionally be obtained by asking for examples. An advantage for the purpose of answering this sub-question is that it allows asking the participants the same questions, while still being flexible to ask questions outside the pre-defined set based on the response of the participant (Dearnley, 2005). This is particularly useful when the roles and responsibilities of participants slightly differ. For example, when speaking to a participant in a coordinating function, the reasoning behind organisational protocols and guidelines could be explored in more detail than when speaking to a participant in an operating role, who might see these more as a limiting factor that hinders certain choices. It allows to identify one's independent thoughts in a group (Adams, 2015). Another advantage of semi-structured interviews is that it allows exploring a wide range of issues that is unknown beforehand by the researcher (Dearnley, 2005).

An important thing to keep in mind is that semi-structured interviews are "time-consuming, labour intensive, and require interviewer sophistication" (Adams, 2015, p. 493). This drawback can be reduced by conducting fewer interviews. This then has the consequence that it is unlikely to involve a large enough sample for the validity of the research (Adams, 2015). In this study, a balance is found by conducting >15 interviews. This can be justified for two reasons; one, IT practitioners are professionals and are limited in their time outside their day-to-day job. Second, as a mixed methods approach is used, potential shortcomings might be covered by quantitative data. A limitation to the nature of interviews is that the participants might be biased towards presenting themselves in a certain way by giving socially acceptable answers. For example, participants might feel pressure from higher-level management or other teams in the organisation. This is potentially solved by the inclusion of quantitative data, but cannot be ruled out entirely.

The tool used to analyse the interview transcripts will be Taguette[2], an open-source qualitative research tool that helps to conduct *thematic analysis*. An advantage of this type of analysis is that it can be used to provide a summary of key concepts in a large body of data where similarities and differences across the data set can be identified (Braun & Clarke, 2006). Another advantage is that it allows for unanticipated insights, as new themes can emerge throughout the analysis. Section 4.1 will provide more explanation of what the interview process entails and how the interviews are analysed with thematic analysis. The deliverable of this sub-question (i.e., an overview of socio-technical factors influencing the patching decisions) will be used in the following sub-question to investigate the patching activity.

2.2.3 Quantitative Log Analysis

> *Sub-question 3: What is the quality of organisations' logs to determine patch activities?*

The third sub-question is explorative and is focused on understanding the quality of organisation's logs to determine patch activity of the organisation. The goal here is two-fold, for one, to explore the possibilities of collecting quantifiable patch metrics and the potential difficulties that come with this. The possibilities for data gathering will be in collaboration with two different teams. This step does not include data analysis yet but aims to investigate the steps leading up to a potential data analysis. The method used here to get more insight into this process is through *discussions* with IT practitioners.

The second goal is to investigate how quantitative data can be useful and if so, what it entails. Where the qualitative data of interviews in Sub-question 2 is useful to explore relationships and factors of influence of decision-making, quantitative data could be used to provide more in-depth insights into the significance of these factors. The possibilities for quantitative data to verify the existence of challenges and the possibilities to use quantitative data as source for decision-making is explored. The quantitative data will be collected by the available logs that can help explore relevant metrics of security patching. The data sources to be used will be dependent on which data is available. With the data that is available, *descriptive analysis* will be conducted to analyse the patching behaviour. With regards to the exploratory purpose of this sub-question, no statistical tests will be performed as it is merely an investigation of how quantitative and qualitative data show interrelations. Section 5.2 will provide more explanation of the scope of data collection and the type of metrics used.

> *Sub-question 4: How does measurement of patching activity relate to the socio-technical factors causing the tensions and dependencies found in the patching decisions?*

[2] https://www.taguette.org/, a free and open-source qualitative data analysis tool

The fourth sub-question is focused on combine the findings of Sub-question 2 and Sub-question 3. The aim here is to synthesise the findings and get an overview of the initial results. No new method is used here, as it builds upon the earlier used methods in the corresponding sub-questions. This synthesis will lead to practical recommendations for organisations.

2.2.4 Evaluative interviews

> *Sub-question 5: How can the findings be used in practice, and how viable are the results to speed up an effective patching process?*

The fifth and final sub-question is evaluative and aims to look at the practical use of the findings. The aim here is to evaluate and interpret the results with using additional empirical data, placing these in context. The method to collect this data is through *open interviews* with or *written feedback* from IT practitioners of different layers of the organisation, including higher-level management. The goal of this validation is two-fold: it aimed to explore opinions and reactions on the findings and reflect on the goal of the IT department regarding patching to bring the findings in context. The content and shape of the discussion will be based on the findings of the previous sub-questions. This can be seen as a last iterative feedback step before answering the main research question.

2.3 Research flow and structure

An overview of the activities of the research approach, the methods used to answer each sub-question and the relation between different sub-questions is presented in a research flow diagram in Figure 6.

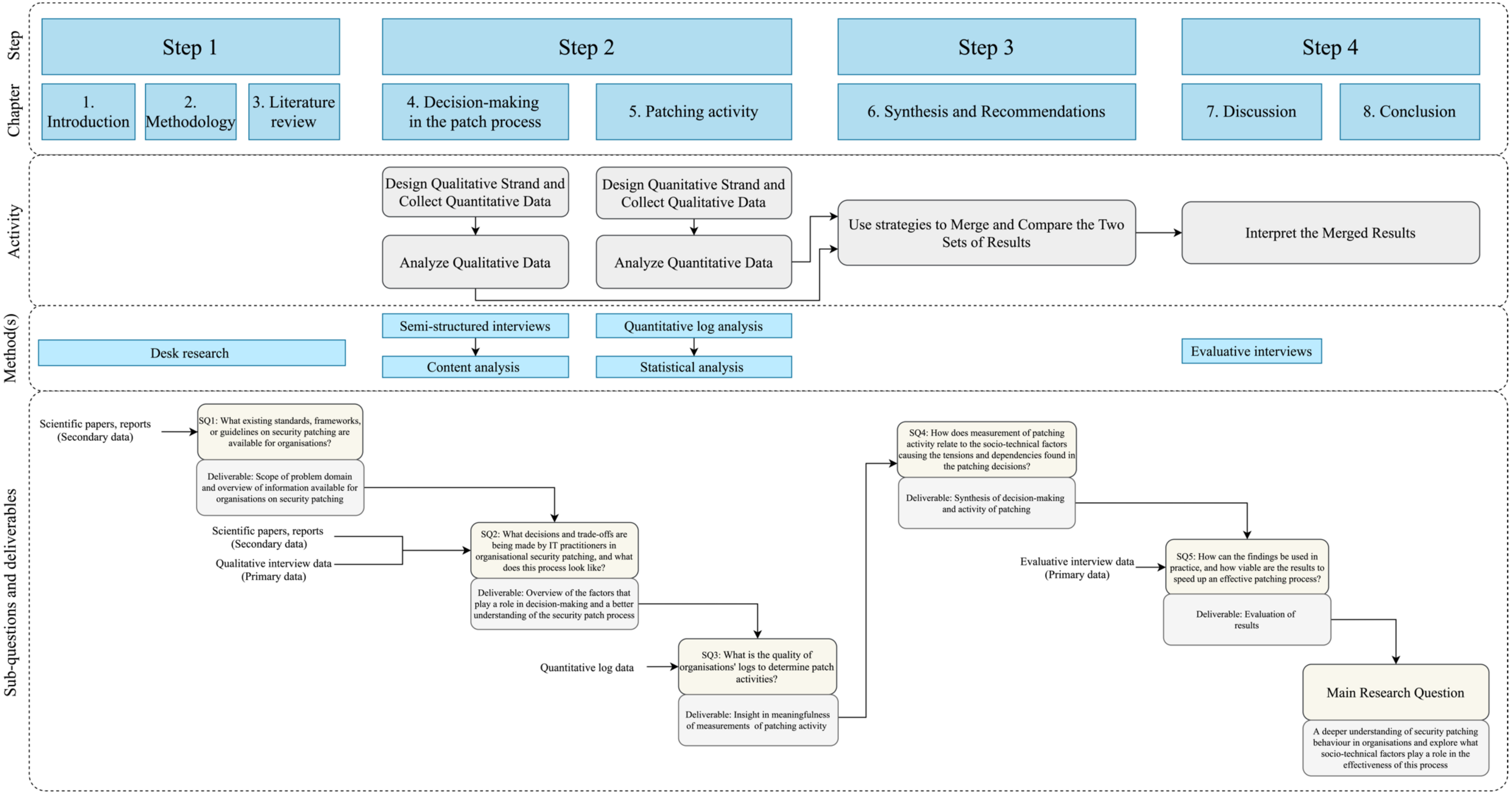

Step
Chapter
Activity
Method(s)
Sub-questions and deliverables
Step 1
Step 2
Step 3
Step 4
1. Introduction
2. Methodology
3. Literature review
4. Decision-making in the patch process
5. Patching activity
6. Synthesis and Recommendations
7. Discussion
8. Conclusion
Design Qualitative Strand and Collect Quantitative Data
Design Quantitative Strand and Collect Qualitative Data
Analyze Qualitative Data
Analyze Quantitative Data
Use strategies to Merge and Compare the Two Sets of Results
Interpret the Merged Results
Desk research
Semi-structured interviews
Content analysis
Quantitative log analysis
Statistical analysis
Evaluative interviews
Scientific papers, reports (Secondary data)
SQ1: What existing standards, frameworks, or guidelines on security patching are available for organisations?
Deliverable: Scope of problem domain and overview of information available for organisations on security patching
Scientific papers, reports (Secondary data)
Qualitative interview data (Primary data)
SQ2: What decisions and trade-offs are being made by IT practitioners in organisational security patching, and what does this process look like?
Deliverable: Overview of the factors that play a role in decision-making and a better understanding of the security patch process
Quantitative log data
SQ3: What is the quality of organisations' logs to determine patch activities?
Deliverable: Insight in meaningfulness of measurements of patching activity
SQ4: How does measurement of patching activity relate to the socio-technical factors causing the tensions and dependencies found in the patching decisions?
Deliverable: Synthesis of decision-making and activity of patching
Evaluative interview data (Primary data)
SQ5: How can the findings be used in practice, and how viable are the results to speed up an effective patching process?
Deliverable: Evaluation of results
Main Research Question
A deeper understanding of security patching behaviour in organisations and explore what socio-technical factors play a role in the effectiveness of this process

Figure 6 - Research Flow Diagram

3 | LITERATURE REVIEW

"Benjamin Franklin once said that "an ounce of prevention equals a pound of cure." Patch and vulnerability management is the "ounce of prevention" compared to the "pound of cure" that is incident response"
– Standardisation body NIST (Mell, Bergeron, & Henning, 2005, p. 2)

Where Chapter 1 sets out the problem, this chapter aims to better understand the context organisations are in. This consists of both getting a better understanding of what security entails for organisations as well as the internal and external interactions in the ecosystem of security practices. Furthermore, as the title hints at, this chapter aims to investigate existing literature for organisations on how to deal with security and patch management in particular. This aims to understand what 'good' patch management entails and whether the provided guidance is aligned with the complexity of the organisation's environment.

Section 3.1 will determine the scope of IT security and will define the core concepts used in this study. Section 3.2 discusses the interactions of IT practitioners both inside and outside the organisation. This section will also discuss the context influencing these interactions and argues for the need for IT governance. Section 3.3 will dive deeper into the existing literature on guidelines, frameworks, and standards. The applicability of these will be explored and evaluated in section 3.4, providing an answer to sub-question 1.

3.1 IT Security in organisations

Every organisation collects, processes, stores, and transmits information. Information and the related processes, systems, networks, and people are significant *assets* for organisations to achieve their business objectives (ISO/IEC, 2018). An organisation's assets can range from tangible assets (e.g., hardware, software) to intangible assets (e.g., reputation, intellectual property, information). However, the presence of *threats* introduces *risks*, which can disrupt the operation of these assets when vulnerabilities are exploited. The practice of dealing with information technology risks posed to an organisation is *risk management*, "…the systematic application of management policies, procedures, and practices to the task of establishing the context, identifying, analyzing, evaluating, treating, monitoring, and communicating information security risks" (Watson & Jones, 2013, p. 120). Inevitably, organisations accept risks to a certain extent by the usage of software that could potentially include vulnerabilities. Not all vulnerabilities become known, and furthermore, not all vulnerabilities form a threat. However, when vulnerabilities do become publicly known, the risk increases as threat actors are more likely to exploit that certain vulnerability (Souppaya & Scarfone, 2022).

There are roughly four strategies organisations can apply to deal with the risks posed by *software vulnerabilities,* as discussed by Souppaya and Scarfone (2022). Organisations can choose to *accept* the risk, for example, when either the impact or the probability of the threat to occur is estimated to be low. The risk can be *transferred* by sharing the consequences the risk entails with another party through security insurance or using software-as-a-service (SaaS). The risk can be *avoided*; this reduces the probability of a threat by minimizing the attack surface, for example, by shutting down systems or uninstalling the vulnerable software. Finally, the risks can be *mitigated*, where the risk of a threat is reduced or eliminated by deploying security *controls*, for example, by vulnerability patching or using firewalls to isolate vulnerable assets (Souppaya & Scarfone, 2022). Figure 7 represents the relations between these concepts.

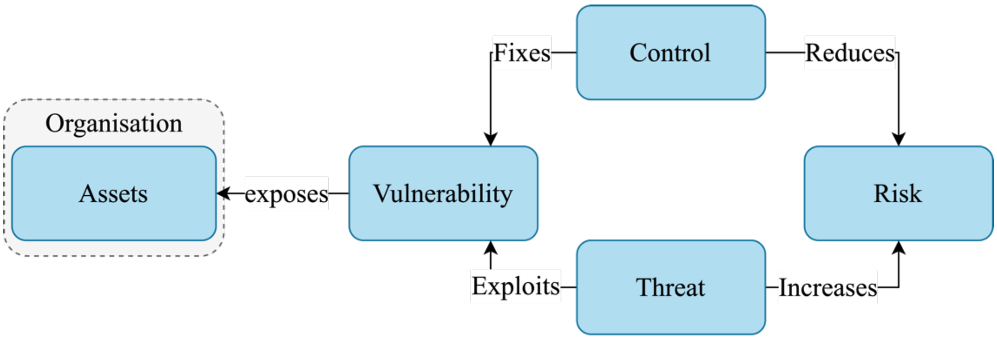


Figure 7 - Relation between core concepts assets, vulnerability, threat, control, risk

All *security*, in essence, is "… about the protection of assets from the various threats posed by certain inherent vulnerabilities" (Von Solms & Van Niekerk, 2013, p. 100). *Cybersecurity* is perhaps the most commonly known term, where "…information and ICT are the underlying cause of the vulnerability" (Von Solms & Van Niekerk, 2013, p. 100). Cybersecurity involves a broad range of intangible assets, of which humans and their interests is the main asset to protect, as illustrated in Figure 8 (Von Solms & Van Niekerk, 2013). Examples of use cases that fall under this broad definition of cybersecurity are cyberbullying, cyber terrorism, cyber fraud, and cyber scams.

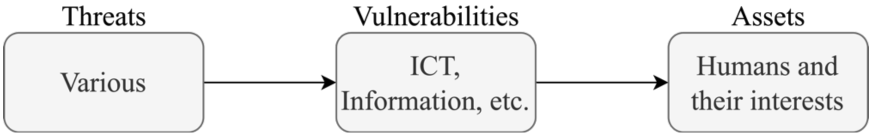


Figure 8 - Definition of Cybersecurity

Information and Communication Technology Security "…deals with the protection of the actual technology-based systems on which information is commonly stored and/or transmitted" (Von Solms & Van Niekerk, 2013, p. 98), where the focus lays on the actual ICT infrastructure. Here the asset itself is ICT, as displayed in Figure 9.

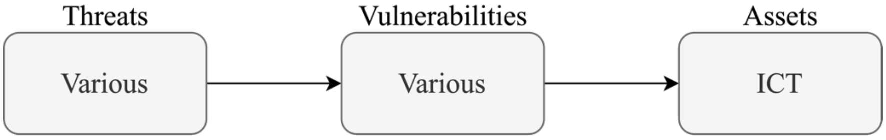


Figure 9 - Definition of ICT security

Information security focuses on ensuring the confidentiality, availability, and integrity of information (ISO/IEC, 2018). Whitman & Herbert define this as "the protection of information and its critical elements, including the systems and hardware that use, store, and transmit that information" (2009, p. 8). Confidentiality is reached when information assets are protected from unauthorised disclosure. Integrity is reached when information assets are protected from unauthorised alteration. Availability is reached when information assets are ensured to be accessible by authorised individuals in a timely and reliable way. Here the vulnerabilities are present in the ICT systems that can expose the assets of information (Von Solms & Van Niekerk, 2013), as illustrated in Figure 10.

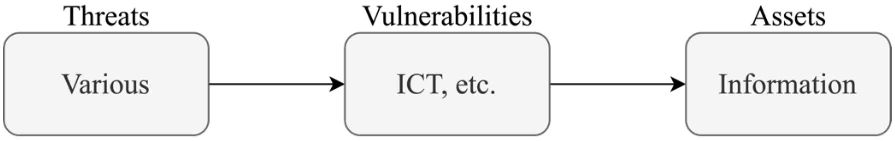


Figure 10 - Definition of Information security

Based on these definitions and the purpose and scope of this research, the definition of *cybersecurity* becomes too broad, as the ICT infrastructure and information are not the underlying cause of vulnerabilities. The definition of *ICT security* becomes too narrow, as the asset to protect is not the ICT infrastructure itself, but vulnerabilities in the ICT infrastructure need to be fixed in order to protect information assets. Therefore, in this research, *information security* suits the objective best and is the perspective used to observe security. *Software Security Patching* can then be seen as a mitigating control to deal with risks on the level of information security, and can be defined as the practice of installing fixes to security vulnerabilities in software products and systems deployed in an organisation's IT environment (Dissanayake et al., 2022). According to BIR (2014) the aim of security patching is twofold:

- To provide insight into the current state of vulnerabilities and applied patches within the managed IT system and;
- To create stable and secure information technology provision as efficiently as possible and with as few disruptions as possible (BIR, 2014).

Software Security Patching is the practice of installing fixes to security vulnerabilities in software products and systems deployed in an organisation's IT environment (Dissanayake et al., 2022)

The definition by Dissanayake et al. (2022) involves two important elements, the *practice* and the *fixes.* Starting with the latter, a *Software Security Patch* (in short *patch*) is a piece of code that 'fixes' an identified problem in software programming (Tracy, Jansen, Scarfone, & Butterfield, 2007). This 'repair job' is developed as a replacement for or an insertion in an existing 'flawed' piece of code (Tracy et al., 2007). This is a fairly straightforward action when looking at it on a technical level. Patches, therefore, aim to mitigate vulnerabilities in software to reduce the opportunity for exploitation. The applicability of patches ranges from operating systems to applications, and firmware to configurations; any piece of software. Patches are not solely used for security but can also be used for bug fixes or feature updates. Often these will be combined and released in one patch. For the scope of this study, when referred to a patch, a security patch is meant unless stated otherwise.

Software Security Patch (in short *patch*) is a piece of code that 'fixes' an identified problem in software programming (Tracy et al., 2007)

It becomes more complex when taking a governance perspective to look at the *practice*. NIST defines *Patch Management* as "the process for identifying, acquiring, installing, and verifying patches for products and systems" (Souppaya & Scarfone, 2013, p. vi). This requires multiple steps and goes way beyond fixing a piece of code, but involves many different aspects such as information retrieval, asset management, planning, and prioritising. This process is a combination of multiple facets and includes different socio-technical elements such as coordination, communication, and collaboration with other stakeholders. The next section will explore the context organisations are dealing with in the current age of software.

Patch Management is "*the process for identifying, acquiring, installing, and verifying patches for products and systems"* (Souppaya & Scarfone, 2013, p. vi)

3.2 Organisations in the ecosystem of IT security

Organisations do not operate in a vacuum; that would make the practice of information security nearly effortless. The choices and risk considerations organisations need to make are dependent on the interactions with their environment. This section aims to explore the root cause of the need for IT security and software security patching by examining the context and environment organisations are in.

3.2.1 Interactions outside the organisation

In the past, organisations' software mostly operated on internal networks, where multiple layers of network security controls kept it protected (Souppaya & Scarfone, 2022). In these days, although patching was considered important, it has not always been given priority. Nowadays, business operations demand dynamic IT and the usage of more internet-facing applications, such as mobile, cloud, and virtual machines. For organisations, it brings many benefits to use third-party software products to provide these services to their employees. Using third-party software is regularly cheaper and less time-intensive than developing software in-house. Additionally, third parties are specialised and have the expertise that brings value to the functioning of the IT systems. Besides, using third-party software can increase the scalability and flexibility of IT operations (Cáceres, Vaquero, Rodero-Merino, Polo, & Hierro, 2010).

However, with the rise of more internet-facing applications of third parties, the risk for systems to be compromised is much more significant, therefore increasing the priority of patching. As consumers (i.e., organisations) of software products, it is at the present time normalised that defect-free software is not a possibility. Apart from the inevitable that software comes with unforeseen bugs and flaws, software maintenance is a very lucrative activity for *software vendors*. Voas (2020) in his article 'the "patching" mentality' experienced a situation where software vendors bid on a contract to provide software and maintenance for an organisation. The interesting part was that even though the initial costs for the software vendor were greater than the revenue of the contract, many vendors brought out a bit. The reason for this is that the maintenance phase is where profit can be made to fix issues in the software product: "the lower the quality on the front end, the bigger the revenue/profits on the back end" (Voas, 2020, p. 12).

When it comes to patching, the main aspect of software maintenance is notifying organisations of vulnerabilities in their software products and providing the patch to be deployed by the organisation. Becoming aware of a vulnerability is the first step in determining the strategy for the actions that need to be taken. Apart from these software vendors, *(social) media* play an essential role in making organisations aware of the existence of vulnerabilities. Often, vendors bundle their patches; in this way, organisations are not overloaded with patches but are able to time, plan, and test accordingly (Souppaya & Scarfone, 2013). The most well-known example of this is Patch Tuesday, which takes place on the second Tuesday of each month, where vendors such as Microsoft, Adobe, and Oracle release patches for their software products[3]. Exceptions are those vulnerabilities where it is known that these are actively exploited. In these cases, a suitable patch is issued immediately, if available (Souppaya & Scarfone, 2013). In these situations of emergency, timing, planning, and testing are difficult, and sometimes even impossible.

This proves that organisations are highly dependent on the services vendors provide. To some extent, vendors can use this dependency to influence decision-making. For example, when maintenance becomes too costly, vendors generally stop to support older versions with patches (Souppaya & Scarfone, 2013), making it a better option for organisations to patch anyway. However, organisations have the 'patching rights' and therefore the decision-power to apply a patch or not. Certainly with newer software products, organisations (as the asset owner) have the ultimate say. According to August et al. this endowment of patching rights

[3] Microsoft's security update guide https://msrc.microsoft.com/update-guide (Microsoft, n.d.)

"lacks the incentive structure to induce better security-related decisions" (2019, p. 4575), resulting in a large group of organisations with unpatched systems.

When reasoning from a *threat actor* perspective, exploiting vulnerabilities can be valuable for many reasons. de Bruijne, van Eeten, Gañán, and Pieters (2017) updated the cyber actor typology commissioned by the WODC (Research and Documentation Centre) of the Ministry of Security and Justice of the Netherlands. The new typology is presented in Table 2, where the actor motivation of each of the actor types is added for a clearer overview. For educational organisations, the main types of threat actors are *extortionists, state (sponsored) actors*, and *insiders*.

Extortionists are economically driven and deploy attacks such as ransomware and DDoS. The example provided in section 2.1.2 Maastricht University shows how these attacks are deployed. Malware attacks are most commonly deployed against educational organisations when compared to other industries (e.g., professional services, retail, high tech) (Scholz, Hagen, & Lee, 2020). The main reason behind these attacks is for financial gains. In the case of ransomware attacks, this is directly demanded from the organisation that fell victim. An example given by Scholz et al. (2020) illustrates a university on the West Coast of the United States where extortionists were able to encrypt valuable research data of the school's medicines department. Eventually, $1.14 million USD was paid in cryptocurrency to receive a decryption key. In another example, a university in the United Kingdom fell victim to a ransomware attack but decided not to pay. This resulted in the shutdown of all IT systems, causing the start of a new academic term to be delayed (Scholz et al., 2020).

Table 2 - Threat actor typology (first two columns adopted from de Bruijne et al. (2017, p. 62))

CSAN actor typology	**TU Delft threat actor typology**	**Actor motivation**
Professional criminals	Extortionists Information brokers Crime facilitators Digital robbers Scammers and fraudsters	Economic (e.g., monetary)
Hacktivists Script kiddies	Hacktivists Crackers	Ideological (societal impact) Personal (e.g., fun, reputation)
Terrorists	Terrorists	Ideological (e.g., disruptive societal impact)
State actors	State actors State-sponsored network	Geo-political (e.g., espionage)
	Insiders	Personal (e.g., revenge, economic)
Private organisations Cyber researchers 'no actor'		

Another way of financial gain is by getting access to (sensitive) data of employees and students. As educational organisations hold a vast amount of personal data (e.g., names, addresses, birth dates), they become increasingly attractive to extortionists (Al-Alawi, Mehrotra, & Al-Bassam, 2020). The authors give an example of a data breach at Penn State University, resulting in the disclosure of the personal information of 18.000 students (Harris & Hammargren in (Al-Alawi et al., 2020)). At the North Dakota University, a database hack resulted in compromising personal data, including social security numbers of over 300.000 alumni. This information is used to exploit or extort individuals or organisations and can be sold on the illegal market.

Similarly, *state (sponsored) actors* are often interested in the intellectual property of educational organisations. Research data and results are valuable for gaining access to strategic information (Bresnick, 2021). In addition, other motives are the disruption of research work by excluding researchers from their data (Bresnick, 2021). Another form of disruption targets the entire functioning of the IT systems of the organisation (Scholz et al., 2020). Disruption can also be carried out by *insiders*; (former) employees who exhibit extensive knowledge of the IT environment and potential misconfigurations. Motives for insiders range from economic incentives to emotional incentives such as revenge. The presence and motives of these different threat actors increase the need for organisations to take fitting security measures.

3.2.2 Interactions within the organisation

Within organisations, security-related decisions are made by different teams within the *IT department*. Depending on the sector and core business of the organisation, this department often consists of developers, application support, Quality Assurance, Security, and system administrators. These teams need to work together to patch effectively, making it a real team effort (Brandman, 2005). When it comes to patching, *system administrators (SAs)* are most involved as their responsibilities lay with ensuring the operability, availability, and security of the systems (Tiefenau et al., 2020). SAs are accountable for managing the large and complex IT environment of the entire organisation. The *Security team* can be involved in assessing the criticality of a patch and advise on defence actions or indicate a timeframe to apply the patch (Brandman, 2005).

The decisions made by the IT department, therefore, directly impact other parts of the organisation, e.g., end-users and top-level management. When systems are not secure and a significant data breach occurs, the reputation of the organisation and top-level management will be affected. At the same time, the IT department is dependent on the decisions made by top-level management. *Top-level management* has control over financial resources, which are often scarce. Werlinger et al. (2008) found that when support from top-level management is greater, the more is spent on preventive resources, and the more effective security is (Werlinger et al., 2008). A conflict between the IT department and top-level management could be present when interests and objectives are not aligned. The classic example is the perceived urgency of security. An often-heard claim is that top-level management does not prioritise security as it does not directly contribute to business outcomes.

Organisations are just as much dependent on the security behaviour of their *end-users*, the ones that use the digital network and services of the organisation. There are two points of interest regarding this stakeholder. On the one hand, end-users value the availability of IT services, meaning no business interruptions and downtime. On the other hand, end-users value security and benefit from organisations patching effectively. Surprisingly, the update behaviour of end-users themselves leaves much to be desired (Tiefenau et al., 2020). Their security hygiene, e.g., using a strong password, backing up regularly, or using antivirus software, is often lacking. For one, end-users are often not aware of the link between their behaviour and the potential consequences for security, Tiefenau et al. (2020) found in their literature study. This is for example the case with installing updates, as it is not apparent how these relate to being secure. Additionally, implementing security measures is perceived as inconvenient as it takes time and causes work interruption (Tiefenau et al., 2020). These interactions indicate that decision-making by IT practitioners is influenced by internal stakeholders within the same organisation.

3.2.3 Contextual influence on interactions

To make it more manageable for organisations to deal with risks, many organisations aim to provide guidance on how to implement security. For instance, *Governmental bodies* are more and more aware of the importance of organisational cybersecurity and steer towards more guidance to organisations. The National Cyber Security Centre (NCSC) from the Dutch Ministry of Justice and Security informs and advises on threats and incidents for information systems (NCSC, n.d.). Examples of governmental bodies that published articles on patch management practices throughout the years are Baseline Informatiebeveiliging Rijksdienst (BIR) (NL), UK’s Department for Work & Pensions (UK), Australian Cyber Security Centre (ACSC) (AU), and Treasury Board of Canada Secretariat (TBS) (CA). Other than advisory, no laws and regulations regarding patching exist. Making patching mandatory is a delicate case. As stated in section 1.1, by far not all vulnerabilities are exploited. This would result in unbearable operations for organisations and many moments of business disruptions.

There are formal laws and regulations by governmental bodies that address data protection and data processing requirements for all organisations that deal with personal data in some way. *Supervisory bodies* have the role of ensuring the adherence to these. For example, the Dutch Data Protection Authority (Dutch DPA) or the European Data Protection Board (EDPB) contribute to the supervision of the application of (inter-)national data protection rules (e.g., Dutch AVG and European GDPR). Although these do not explicitly state patching practices, these do form requirements for organisations in the way they practice and design IT security. When a data breach occurs due to insufficient patching, organisations need to report this within 72 hours, or penalties are faced (Wolford, 2022).

Finally, *advisory and standardisation bodies* provide guidance and recommendations for IT security. When talking about IT standards, several organisations are relevant to consider. The National Institute of Standards and Technology (NIST) developed a cybersecurity framework to help organisations better understand and improve their management of cybersecurity risks (NIST, 2022a). Other relevant standardisation bodies are the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), which jointly published the well-known information security standards ISO/IEC 27000 series. SANS Institute is a cooperative for information security that provides training, certifications, programs, and resources to help cybersecurity professionals (SANS, 2022). The Center for Internet Security (CIS) is a non-profit organisation that serves as an information hub for security measures and recommendations throughout crowdsourcing (CIS, 2022). In section 3.3, the different guidelines and recommendations given by these bodies will be discussed.

3.2.4 The need for IT governance

Figure 11 displays the original security environment (as Figure 7), extended with relevant stakeholders and their corresponding influence on the organisation. The arrows represent the dependencies and interactions between the different stakeholders. One obvious stakeholder part of this ecosystem is *law enforcement*. As these are mainly targeting threat actors, this stakeholder is not explicitly considered in this research.

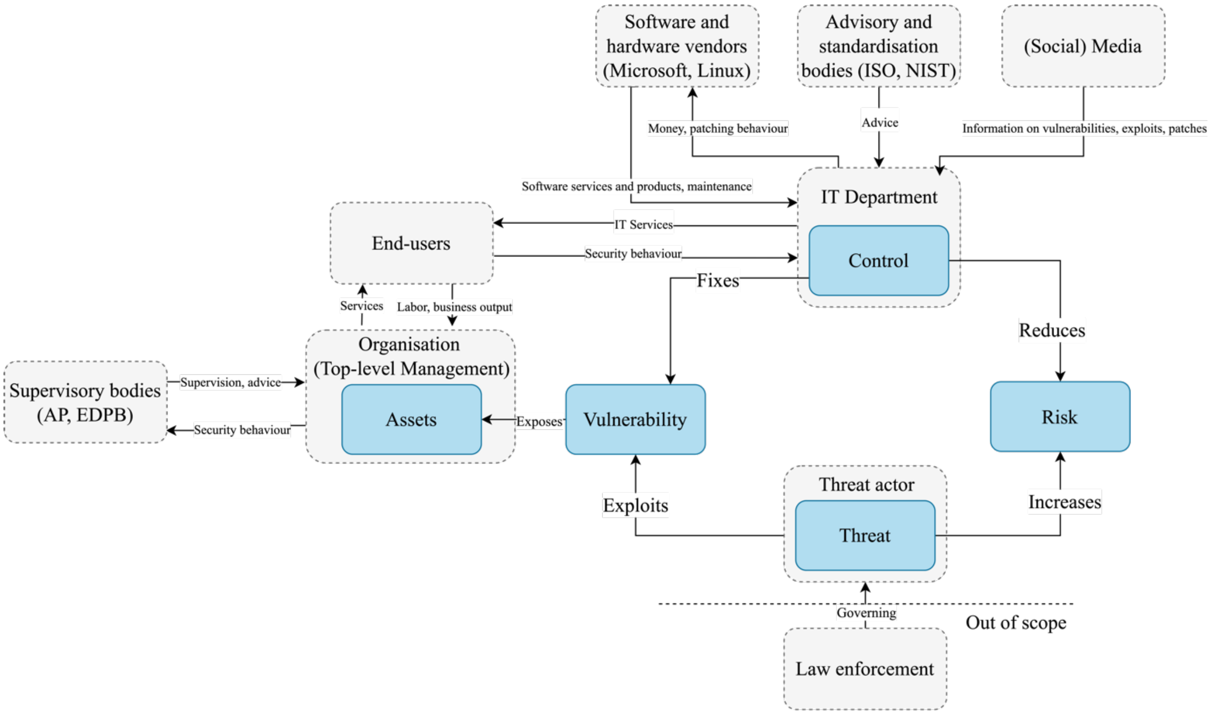

Software and hardware vendors (Microsoft, Linux)
Advisory and standardisation bodies (ISO, NIST)
(Social) Media
Money, patching behaviour
Advice
Information on vulnerabilities, exploits, patches
Software services and products, maintenance
IT Department
Control
End-users
IT Services
Security behaviour
Fixes
Services
Labor, business output
Organisation (Top-level Management)
Assets
Supervisory bodies (AP, EDPB)
Supervision, advice
Security behaviour
Exposes
Vulnerability
Reduces
Risk
Exploits
Threat actor
Threat
Increases
Governing
Out of scope
Law enforcement

Figure 11 - Overview of stakeholders in relation to organisational security

This shows that organisations are thus bound to interact with other stakeholders to practice IT security. There is a high level of dependency on external stakeholders for services, maintenance, and information. At the same time, there is a high level of dependency on internal stakeholders, which scopes the decision-making process of IT practitioners. The framework by Boehm, Merrath, Poppensieker, Riemenschnitter, and Stäle (2018) illustrates this complexity nicely by having multiple layers around organisations' assets, which form the centre to be protected. The first layer of *controls* is involved with the security measures to be taken by an organisation. When taking the example of patch management, the control is applying a patch to fix a vulnerability. The shell around these controls are the *Processes*, the activities and tasks needed to implement the control. In this example, the system might need to be shut down before applying the patch. Processes are established by *Organization*, the way different departments, teams, and employees cooperate and coordinate. One team might need to inform another team that a certain system is unavailable for a period of time. The outer shell is *Governance* which covers the policies, procedures, responsibilities, and decision rights throughout the organisation (see Figure 12).

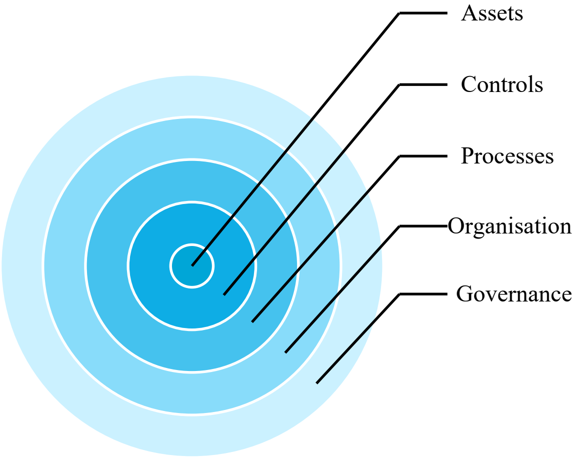


Figure 12 - The shells of IT risk management

The complex dynamic environment results in the need for IT governance to establish processes, policies, and other relevant activities to manage security effectively. There are many IT security frameworks organisations can use for IT risk management. Often, the choice for using one (or more) of the frameworks is based on the type of industry or compliance requirements. The following section (3.3) will present an overview of common bodies that published frameworks or guidelines related to security patching.

3.3 Overview of guidelines and recommendations

Multiple advisory, standardisation, and governmental bodies aim to help organizations with security patch management by publishing articles that describe guidelines and recommendations. The publications are presented in Table 3. Based on the availability of information, these publications will be used in the analysis of this study. Important to mention, however, is that this list is not exhaustive, and more organisations might have published similar articles involving patch management practices. An overview of all guidelines categorised per theme is available in Appendix A.

Table 3 - Publications of advisory, standardisation, and governmental bodies used in this chapter

	Date	Body	Author(s)	Publication Title
Standardisation	2002	NIST Special Publication (SP) 800-40	(Mell & Tracy, 2002)	Procedures for Handling Security Patches: Recommendations of the National Institute of Standards and Technology
	2005	NIST Special Publication (SP) 800-40 Version 2	(Mell et al., 2005)	Creating a Patch and Vulnerability Management Program.
	2013	NIST Special Publication (SP) 800-40 Revision 3	(Souppaya & Scarfone, 2013)	Guide to Enterprise Patch Management Technologies.
	2022	NIST Special Publication (SP) 800-40 Revision 4	(Souppaya & Scarfone, 2022)	Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology
	2013	ISO27002:2013, 2013	(ISO/IEC, 2013)	Information technology — Security techniques — Code of practice for information security controls
Advisory	2008	SANS Whitepaper	(Ruppert, 2008)	Patch Management
	2013	SANS Whitepaper	(Hoehl, 2013)	Framework for building a Comprehensive Enterprise Security Patch Management Program
	2009	CIS 20	(CIS, 2009)	Twenty Critical Controls for Effective Cyber Defense: Consensus Audit Guidelines
Governmental	2014	Baseline Informatiebeveiliging Rijksdienst (BIR)	(BIR, 2014)	Patch Management - Guideline 12.6
	2021	UK's Department for Work & Pension (DWP)	(DWP, 2021)	Security Standard – Security Patching (SS-033)
	2021	Australian Cyber Security Centre (ACSC)	(ACSC, 2021)	Assessing Security Vulnerabilities and Applying Patches
	2022	Treasury Board of Canada Secretariat	(TBS, 2022)	Patch Management Guidance

Some publications are designed to help organisations understand the basics of security patch management and stress the need for designing this the right way (CIS, 2009; DWP, 2021; Mell & Tracy, 2002; Ruppert, 2008; Souppaya & Scarfone, 2013; TBS, 2022). CIS (2009) for example, provides knowledge of common attacks and exemplifies how certain security controls are able to stop these from happening. DWP (2021) aims to help organisations get an understanding of what security requirements need to be met to adhere to the 'international best practice' for security patching. The nature of these guidelines is more explanatory. What can be seen by NIST's 800-40 series is that these are sequels and follow up on the content of the earlier publication. Where the first publication by Mell and Tracy (2002) aims to address all the basic elements of security patching, the latest publication by Souppaya and Scarfone (2022) assumes that this is known to the reader.

Other publications are more designed as a reference and guidance document for organisations to develop their own organisational policies and implement information security controls (ACSC, 2021; BIR, 2014; Hoehl, 2013; ISO/IEC, 2013; Mell et al., 2005). Hoehl (2013) for example aims to provide a framework for organisations to help organisations understand how to incorporate asset inventory control, risk management, and standardization in their policies. BIR (2014) describes policy principles regarding information security for the implementation of a patch management policy by organizations within the Dutch national government. The most recent publication of Souppaya and Scarfone (2022) takes a different perspective in the way they incorporate planning and aim to target all levels of the organisation, from management level to operational level. The nature of these guidelines is more advisory.

Assessing the publications by the year of publication, it can be concluded that, in general, the earlier publications focus more on providing insight into the concept of security patching. The most recent publications are more focused on how to implement security patching policies. All the publications are structured in a process-related way, mentioning different phases of the lifecycle of security patch management. Although all patch processes are dependent on the type of system the patch is implemented on and the context of the patch environment, it can be said that a common process consists of six phases. The main phases of the security patching process based on Dissanayake et al. (2022) are illustrated in Figure 13. The remaining of this section is therefore structured in a similar way.

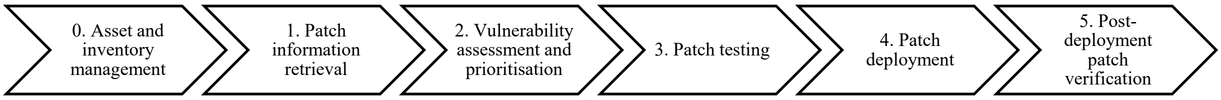

0. Asset and inventory management
1. Patch information retrieval
2. Vulnerability assessment and prioritisation
3. Patch testing
4. Patch deployment
5. Post-deployment patch verification

Figure 13 - Phases of a common patch process

3.3.1 Asset and inventory management

It is difficult to determine which systems require new patches when there is no clear understanding of what assets and inventory are within the organisation's scope (Hoehl, 2013). This is particularly useful to have a clear overview of the scope of the attack surface and what parts are impacted if a vulnerability is exploited (Hoehl, 2013). ISO/IEC (2013, p. 46) states, "a current and complete inventory of assets is a prerequisite for effective technical vulnerability management". If this is not present, it needs to be assumed that all assets need to be patched, resulting in in-effective patching as assumptions of risks and infrastructure resources are not accurately made. Relevant information in such an inventory is the software vendor, version numbers, the current state of deployment (e.g., what software is installed on what systems), the person(s) within the organisation responsible for the software, metadata (e.g., geographic location, data classification, and redundancy, details about compensating controls are valuable when performing initial risk assessment) and history of patches (DWP, 2021; ISO/IEC, 2013; Souppaya & Scarfone, 2013). A decision to be made here is what level of aggregation is necessary, as too much or too little information is not useful. It is recommended that there is an organisation-wide process in place to collect and store this information, e.g., an automated scanning tool or a configuration control system. Additionally, it is recommended to keep the software supplied by vendors up to date to maintain the level of support and keep systems up and running. Simultaneously, it is recommended that third-party access to software modules is carefully controlled and monitored to avoid unauthorised changes to the systems.

3.3.2 Patch Information retrieval

ISO/IEC (2013) states information on vulnerabilities should be obtained in a 'timely fashion'. It fails to give an indication of what a 'timely fashion' is, however. This standardisation body does recommend creating an organisational 'timeline' policy to react to notifications, so it is known what that 'timely fashion' means for the organisation. The decision-right on what is 'timely' seems to lay with the organisation itself. DWP (2021) gives a more specified indication: 'at least weekly'. Information sources are next to 'known, trusted third parties' and 'security sources', based on the asset inventory list (DWP, 2021; ISO/IEC, 2013). Souppaya and Scarfone (2022, p. 4) provide a concrete example of subscribing to "vulnerability feeds from software vendors, security researchers, and the National Vulnerability Database (NVD)" to keep track of new vulnerabilities. Different techniques for retrieving information on vulnerabilities are present (e.g., agent-based, agentless scanning, passive network monitoring) (Souppaya & Scarfone, 2013). It is recommended that the advantages and disadvantages of each technique are established before implementing it in the organisation.

3.3.3 Vulnerability assessment and prioritisation

Once information on a vulnerability and patch release is retrieved, there are decisions to be made by IT practitioners in risk assessment and risk prioritisation. It is recommended to assess mainly three different things (BIR, 2014; DWP, 2021; ISO/IEC, 2013; Souppaya & Scarfone, 2013):

- The security risks if the patch is not deployed
- The operational risks if the patch is deployed
- The possibility of applying other controls (i.e., not using the patch)

DWP (2021) states that the security risks should be based on the defined criticality by either a CVSS or CWE scoring calculation. Furthermore, the impact of the vulnerability should be presented in a factual and quantitative way (Hoehl, 2013). Regarding the operational risks, BIR (2014, p. 6) states that "all risks tied to the installation of that patch need to be evaluated". Additionally, business requirements need to be considered when a patch includes both security and functionality changes. If the operational risks are greater than the security risks, an organisation might decide not to fix the issue. In that case, DWP (2021) suggests reviewing the decision within no more than three months. In the meantime, other controls should be considered, e.g., turning off systems, adapting network borders, and increasing monitoring. Furthermore, a procedure should be in place when no appropriate countermeasure is available to target the security risks (ISO/IEC, 2013). For both security and operational risks, no indication of what these risks entail is given.

If there are multiple patches that need to be installed at the same time, some form of risk prioritisation is recommended. Systems at 'high risk' and those which are known to be exploited in the wild should be addressed first (ACSC, 2021; ISO/IEC, 2013). Another important aspect recommended to be used to prioritise is the location of the asset. First, internet-facing services, second, other important network devices that affect 'high-risk users', and lastly, all other systems. From a procedural side, it is recommended that once the decision is made to install a patch, pre-defined controls related to change management are followed. A recommendation is given that some form of automated ticketing system is used.

3.3.4 Testing

Testing is highly recommended to be done before deploying a patch in the 'live environment'. Here the goal is to identify ways to apply the patch, detect potential side-effects of patch installation and resolve potential conflicts that occur. A test should focus on usability, security, compatibility, effects on other systems, and user-friendliness, according to ISO/IEC (2013). Testing should be conducted on a separate system, and it is recommended that this test environment is as similar to the live conditions as possible (DWP, 2021). This can be costly and sometimes impractical, wherefore virtualisation is another recommended option (TBS, 2022). If testing is not possible due to time, costs, or resource constraints, it is recommended to re-assess the criticality based on the behaviour and expertise of other users (i.e., organisations), and a decision to delay the installation of the patch might be made (ISO/IEC, 2013). Additional statements are made on documenting the decision to apply or reject a patch, the compatibility of a patch, and whether testing is feasible or not in either a risk register or a CMDB (DWP, 2021). If a test is concluded to be satisfactory, deployment must take place.

3.3.5 Deployment

A 'timing target' needs to be established that determines when patches should be in place (Hoehl, 2013). ACSC (2021, p. 1) states, "once a patch is released by a vendor, the patch should be applied in a timeframe commensurate with an organisation's exposure to the security vulnerability and the level of cyber threat the organisation is aiming to protect themselves against". This organisation recommends different time frames depending on the level of threat (e.g., basic, moderate, advanced), whether it is a patch for an application or an operating system, and whether the particular system is 'internet-facing' or connected to the internal network.

If we take an example of a moderate threat of an internet-facing application, ACSC (2021) recommends applying the patch within two weeks, or if an exploit already exists, within 48 hours. When there is a basic threat to a network-connected system's operating system, the recommended timeframe to apply the patch is within one month. The DWP (2021) bases their recommendation on the Common Vulnerability Scoring System (CVSS) or Common Weakness Enumeration (CWE) score. If this is indicated to be 'critical', the patch must be deployed within 14 days of the release day. If the criticality of the vulnerability is indicated to be 'high', the patch must be deployed within 30 days of the release day. BIR (2014) recommends a much less specified timeframe. They state that patches for vulnerabilities with a high probability of exploitation and a high level of potential damage need to be deployed as soon as possible, preferably within one week. However, less critical patches need to be scheduled for the first upcoming maintenance moment.

Throughout all publications considered in this study, only two mention a specific timeframe to apply patches after release, the ACSC and DWP. BIR identified a more general, non-specified timeframe. All other publications do not mention any or keep themselves on 'must happen timely'. This could be the result of the complexity of the environment organisations are in; a standard timeframe seems impossible. Another thing to notice is why the different publications indicate different timeframes (e.g., days, weeks, or even months). This concludes that a unified timeframe to deploy a patch is not existent, wherefore organisations are not bound to follow any guidelines or recommendations.

There are two important aspects in patch deployment to consider; how to patch and by whom. An important aspect of decision-making in patch management is the need to find a balance between security needs, and needs for usability and availability (Souppaya & Scarfone, 2013). Successful patch management starts with a patch policy that is aligned with business objectives (Hoehl, 2013, pp. 10-11). Several aspects need to be established in this policy:

- The scope of what must be patched (e.g., data classification, asset value, location, and business purpose)
- Prioritisation and timing targets
- Roles, responsibilities, and authority associated with patching
- Procedures for obtaining exemption from patching

It is recommended to patch automatically 'wherever possible' and only manually if not (BIR, 2014; DWP, 2021; Hoehl, 2013; Mell et al., 2005). During patching, organisations should avoid potential resource overload (Souppaya & Scarfone, 2013). Patch deployment should only be done by authorised and trained administrators. End-users should not have the ability to install unauthorised patches or disable already installed patches (ISO/IEC, 2013; Souppaya & Scarfone, 2013). It is recommended to document all patches, both manual and automated, and an audit log should be kept for all procedures undertaken (BIR, 2014; ISO/IEC, 2013).

3.3.6 Post-Deployment

Monitoring is recommended to keep track of a patched system's behaviour and ensure a patch is deployed successfully. Examples of non-deployment given by Souppaya and Scarfone (2022) are the uninstallation by a user or attacker, restoring of an unpatched version from a backup, or the resetting of a system to the factory-default state. It is however recommended to have a rollback strategy (including previous versions) in place if deployment turns out to create unwanted side effects (ISO/IEC, 2013). Old versions should be archived, including corresponding relevant information. The effectiveness of patch deployment can be verified through network and host vulnerability scanning (ISO/IEC, 2013). Additionally, success and failure rates of patch deployment need to be measured to identify outliers or trace patch installation failures (TBS, 2022). Monitoring is recommended to ensure the effectiveness and efficiency of the patch itself and the patch process in general (ISO/IEC, 2013).

3.3.7 Principles and procedures of patch management

Two publications explicitly mention principles of security patching. Souppaya and Scarfone (2022) provide four general principles for organizational patch management.

- Problems will be inevitable, be prepared for them.
- Simplify decision making.
- Rely on automation.
- Start improvements now.

Souppaya and Scarfone (2022) state that to deal with the inevitable problems security patching brings to operations, an organisation's culture needs to change from fearing these, which results in delays in deployment, to become aware that these are necessary inconveniences that help to prevent crucial compromises. Additionally, the authors state that simplifying decision-making benefits organisations in dealing with balancing time, resources, expertise, and tools, resulting in a more feasible practice (Souppaya & Scarfone, 2022). This could be done by planning in advance so that decisions on how to respond can be made quicker when vulnerabilities become known. Furthermore, automation is highly recommended to keep up with security patching of the wide range of "assets, software installations, vulnerabilities, and patches" (Souppaya & Scarfone, 2022, p. 9). Lastly, the authors recommend starting directly with the implementation of needed changes as some of these might take some years to implement.

Ruppert (2008) states the '12 steps to patch management', which expresses the need for putting the focus on establishing a routine, maintaining consistency, expanding awareness, extending communication, and embracing business and IT support. Additionally, the author stresses the importance of documentation throughout the process, including the "scope, roles and responsibilities, timeline, functional guidelines, and procedures" of the process (Ruppert, 2008, p. 23). The steps are high-level advice such as 'establish importance', 'establish security organisation', 'plan for risk management', and 'monitor and review'.

3.4 Conclusion and discussion of the context and recommendations

Sub-question 1: What existing standards, frameworks, or guidelines on security patching are available for organisations?

Findings illustrate that security patch management is a complex problem where multiple stakeholders are involved with different decision-making rights and divergent interests. Different dependencies between stakeholders make it a multi-actor problem domain which cannot be solved by the actions of a single actor. Even within one organisation, different departments are making security patching a troublesome practice due to dependency on services, behaviour, and resources. Apart from threat actors, it can be said that all other stakeholders are pursuing the same incentive in the end, security. The troubling factor, however is the prioritisation of security over other values. For some stakeholders, security is most urgent, e.g., IT department, software vendors, standardisation and advisory bodies. For other stakeholders, e.g., top-level management and end-users, values of availability of services and business continuity.

The question then is to what extent guidance of existing literature is useful for organisations to cope with this high level of complexity. Essentially, the content of the publications increases the level of knowledge on security patch management and identifies what aspects are relevant for organisations to consider. These, therefore, help organisations to grasp an understanding of what security patch management entails. In this sense, the guidance succeeds in helping organisations get started with the practice of security patching. Nonetheless, there are limitations where publications fail to provide guidance on. Firstly, most recommendations are very generic and lack specified details. Formulation is done in an abstract way, e.g., "the policy should contain a few key components to be effective" (Hoehl, 2013, pp. 10-11). For one, this might be done to increase the generalizability to a wider range of organisations. However, formulating recommendations in this way will generate additional questions the publications do not provide an answer to, e.g., what is a ‘few’? And what is meant by ‘effective’?

Second, a unified indication of a timeframe to install a patch after release is not existent, and most recommendations only generally state patching needs to happen ‘in a timely fashion’. This might cause many organisations not to have a target of what to aim for, resulting in a feeling of being ‘too late’, starting on day one of patch release. The publications that do state a timeframe hold opposing views of what that timely fashion entails. It does not become clear to what extent criteria such as CVSS scores or whether a system is internet-facing influence the timeframe. The publications also fail to indicate what ‘right’ patching is and when a patch process can be identified as being successful. Undoubtedly, providing a unified guideline that suits all organisations (and its level of complexity) is beyond the bounds of possibility. That is not to say that guidance cannot help organisations indicate what, according to them, is a realistic aim to adhere to. For example, providing an overview that indicates the ranges of the timing of patch deployment for organisations based on the number of systems, the nature of systems, and the resources available, could help organisations get an idea of their current practice.

Third, many recommendations are formulated as decisions to be made by organisations rather than providing concrete guidance on how to do things, e.g., “if a vulnerability is not being exploited yet, organizations should carefully weigh the security risks of not patching with the operational risks of patching without performing thorough testing first” (Souppaya & Scarfone, 2013, p. 16). Although to some extent, it makes sense that recommendations cannot make the decisions for organisations, they fail to give organisations an understanding of how it could be done. For example, how can operational risks be compared to security risks? Are there certain criteria to take into account? In general, the publications do not provide concrete answers to these types of questions. Furthermore, the majority of recommendations are not focused on the decision-making process during patching itself. Guidance on how to organise coordination issues, stakeholder dependencies, or communication mechanisms is not established. For example, “a patch management process requires proper accountability and ownership, along with good governance and stewardship” (TBS, 2022, par. 5). Although coordination issues are indicated, it does not become clear how these should be addressed.

However, one publication succeeds in giving actionable guidance, that of Souppaya and Scarfone (2022). This publication is the odd one out, taking a different approach to organisational patching, where the focus lays more on the involvement of different stakeholders and the conflicts of interest that play a role in planning for patch deployment. Where other publications set the environment of IT practitioners to be defined and rigid, this publication addresses this gap by targeting all levels of the organisation to deepen their understanding of the role of patching in risk management. This approach shifts away from ‘the ultimate solution’ but stresses the need to make decisions about dilemmas and that trade-offs are simply inevitable. This publication is, therefore, more aligned with the modern challenges patching brings for organisations and therefore provides more valuable guidance.

4 | DECISION MAKING IN THE PATCH PROCESS

"Patching takes up much more time than in the past, it almost starts to look like our daily work"
– interviewed IT practitioner

As Chapter 3 indicated that security patching is a part of organisational risk management, in essence the process of patching is about IT practitioners making certain decisions that most suitably handle the situation they are faced with. The literature review in the previous Chapter (3) already helps to get an indication of the context organisations need to make these decisions in, and how the presence and dependencies of stakeholders influence the decisions to be made. Publications including recommendations and guidelines help indicate what elements organisations need to address but leave the fulfilment of these decisions open. This chapter therefore aims to explore how decision-making is carried out throughout the security patch process. Interviews are a suitable source of data to explore this practice.

This chapter is structured as follows; Section 4.1 will provide an explanation of the conducted interviews and how these are analysed. Section 4.2 will explore the different factors that play a role in decision-making throughout the different phases of security patching in-depth with help of quotes from interview participants. Section 4.3 will tie the different findings of section 4.2 together to summarize the aspects of decision-making on a higher level of abstraction. Furthermore, tensions and coping strategies to deal with these tensions will be identified. Finally, a conclusion will be given in section 4.4 to answer sub-question 2.

4.1 Data collection and interpretation method

For the purpose of better understanding decision-making throughout the security patching process, semi-structured interviews are held with 14 IT practitioners from seven different teams in the IT department. Where section 2.2.2 discussed why interviews are a suitable way of data collection, this section will discuss how the interviews are conducted and how the interview results will be interpreted. Table 4 provides an overview of the interviewees' team and roles. The participants are numbered for the purpose of referencing the quotes used throughout the following section (4.2).

Table 4 - Overview of interviewees

Interviewee(s)	Team	Roles	Participant
6	Systems (Linux, Windows)	System administrator, Manager	P1, P2, P3, P4, P5, P6
2	Infra	Manager, head	P7, P8
2	Quality	Manager	P9, P10
1	Workplace support	Manager, head	P11
1	Applications	Manager, head	P12
1	Data management	Manager, head	P13
1	Security	Manager	P14
Total: 14			

4.1.1 Interview process

After a short introduction of the purpose of the study and the interview, all respondents were asked to sign an informed consent form, either written or orally. This explained that data would not be shared, only in an aggregated and anonymized way. All participants gave permission to record the interview on audio. Eight interviews were held in-person, and six interviews were held online via MS Teams due to preference by the interviewee (e.g., busy schedules). All interviews were held in Dutch, wherefore a note needs to be made for the quotes used throughout this chapter. The researcher did his best to translate the statements as accurately as possible. The interviews took on average 42 minutes, with the shortest 29 minutes and the longest 53 minutes. The interview was structured in three parts. The first part consisted of general questions about the interviewee's role and responsibilities within the team, to set a context for the follow-up questions. The second part consisted of general experiences with security and patching practices. The third part followed up on the experiences in more detail. After which, the final part was open for additional comments by the interviewee. The initial list of interview questions is included in Appendix B.

4.1.2 Interview interpretation

Braun and Clarke (2006) provide an outline for *thematic analysis*, involving six steps to analyse qualitative data (Table 5). These steps are taken in this research to develop themes from underlying data.

Table 5 - Phases of thematic analysis (adopted from (Braun & Clarke, 2006, p. 87))

Phase	Description of process
1. Familiarizing yourself with your data	Transcribing data (if necessary), reading and re-reading the data, noting down initial ideas.
2. Generating initial codes	Coding interesting features of the data in a systematic fashion across the entire data set, collating data relevant to each code.
3. Searching for themes	Collating codes into potential themes, gathering all data relevant to each potential theme.
4. Reviewing themes	Checking if the themes work in relation to the coded extracts (Level 1) and the entire data set (Level 2), generating a thematic 'map' of the analysis.
5. Defining and naming themes	Ongoing analysis to refine the specifics of each theme, and the overall story the analysis tells, generating clear definitions and names for each theme.
6. Producing the report	The final opportunity for analysis. Selection of vivid, compelling extract examples, final analysis of selected extracts, relating back of the analysis to the research question and literature, producing a scholarly report of the analysis.

Two approaches were taken to analyse the interviews. Once the audio recordings were converted into transcripts, both *deductive* and *inductive analyses* were conducted, as the purpose of the interviews is two-fold. First, it aims to understand what the practice of security patch management entails, wherefore deductive coding has been used. With deductive coding, themes and concepts are already known before analysing the raw data. In this case, due to an extensive literature review in Chapter 3, basic concepts and common phases of security patch management were used. The perspective of the researcher is already shaped before the analysis in this case. Figure 14 provides an example of this process, where categories are used to combine different codes derived from raw interview data. These results are used in section 4.2 Analysis of decision-making in each phase of the patch process to group codes of similar categories together.

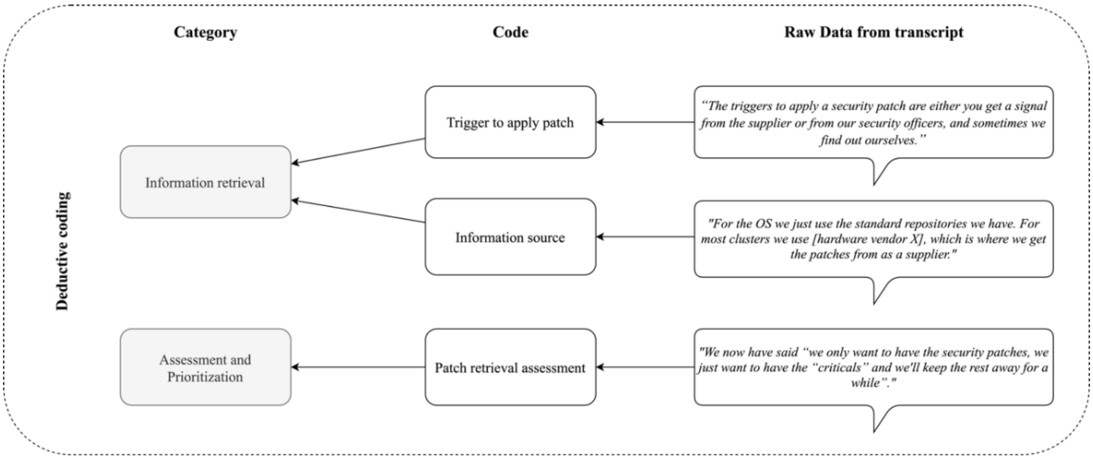
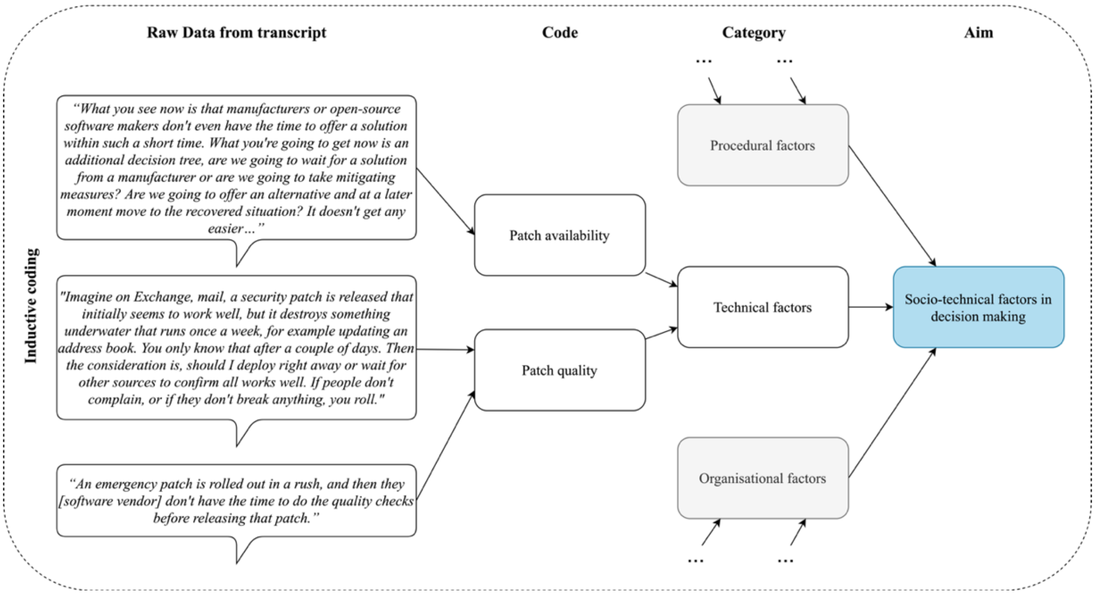

Category
Code
Raw Data from transcript
Deductive coding
Information retrieval
Assessment and Prioritization
Trigger to apply patch
Information source
Patch retrieval assessment
"The triggers to apply a security patch are either you get a signal from the supplier or from our security officers, and sometimes we find out ourselves."
"For the OS we just use the standard repositories we have. For most clusters we use [hardware vendor X], which is where we get the patches from as a supplier."
"We now have said "we only want to have the security patches, we just want to have the "criticals" and we'll keep the rest away for a while"."

Figure 14 - Deductive coding example

The second aim is to understand which decisions need to be made, why these need to be made, and how these are made by IT practitioners. The main aim is thus to explore the socio-technical factors that influence decision making. Here, because no prior knowledge of themes or categories was known, inductive coding has been used to analyse the same interview transcripts. Non-explored concepts were identified throughout the different transcripts, as the example in Figure 15 illustrates. In the different sub-sections of section 4.2, identified codes are discussed.



Raw Data from transcript
Code
Category
Aim
Inductive coding
"What you see now is that manufacturers or open-source software makers don't even have the time to offer a solution within such a short time. What you're going to get now is an additional decision tree, are we going to wait for a solution from a manufacturer or are we going to take mitigating measures? Are we going to offer an alternative and at a later moment move to the recovered situation? It doesn't get any easier…"
"Imagine on Exchange, mail, a security patch is released that initially seems to work well, but it destroys something underwater that runs once a week, for example updating an address book. You only know that after a couple of days. Then the consideration is, should I deploy right away or wait for other sources to confirm all works well. If people don't complain, or if they don't break anything, you roll."
"An emergency patch is rolled out in a rush, and then they [software vendor] don't have the time to do the quality checks before releasing that patch."
Patch availability
Patch quality
Procedural factors
Technical factors
Organisational factors
Socio-technical factors in decision making
…
…
…
…

Figure 15 - Inductive coding example

4.2 Analysis of decision-making in each phase of the patch process

Table 6 presents an overview of the identified socio-technical factors combined into challenges of three main categories; organisational, procedural, and technical. These socio-technical factors are explained in more detail in the following sections. Section 4.2.1 identifies factors that play a role of importance in the environmental context organisations are in. The factors that influence the general phases of the patch process are discussed in section 4.2.2. Organisation's procedures and guidelines are discussed in section 4.2.3. Furthermore, the difference of being in a state of emergency is discussed in section 4.2.4. In each of these sections, the corresponding factor will be indicated in-text with [F..]. Section 4.3 will provide more clarity on how the three most-right columns (i.e., type, aspect, decision) should be interpreted.

Table 6 - Challenges and socio-technical factors of security patching

Category	Challenge	Factors	Type	Aspect	Decision
Organisational	**CO1** Developments of threat environment	**[F1]** Decreasing timeframe between vulnerability awareness and exploit by threat actor	Cause	Security	Time
		[F2] Unpredictability of vulnerability exploit	Cause	Security	Time
	CO2 Behaviour and awareness of system-owners[4]	**[F3]** Lack of threat perception of system-owners	Barrier	Security	Patch
		[F4] Lack of perception that patching reduces threat of system-owners	Barrier	Security	Patch
		[F5] Lack of knowledge how to patch of system-owners	Barrier	Operational	Time
		[F6] Lack of prioritization of system-owners	Barrier	Operational	Time
	CO3 Role and capability of IT practitioners	**[F7]** Human error during patch deployment of IT practitioners	Cause	Operational	Patch
		[F8] Human error in patch assessment of IT practitioners	Cause	Applicability	Patch
		[F9] Need for balance between patching and tasks of day-to-day job of IT practitioners	Constraint	Operational	Time
	CO4 Lack of human resources	**[F10]** Lack of human capacity of IT practitioners limits the frequency of patch deployment	Constraint	Operational	Time
	CO5 Organisational structure	**[F11]** Decentralized and large organisational structure	Cause	Applicability	Patch, Time
		[F12] Need to balance security practises and organisational objectives	Constraint	Availability, Security	Time
Procedural	**CP6** Collaboration	**[F13]** Inter-dependencies of different teams (e.g., Security, Systems, Quality) within the IT department	Barrier	Operational	Time
		[F14] Dependency of external stakeholders (e.g., end-users) to determine moment of patch event for central IT services	Barrier	Availability	Time
		[F15] Dependency of external stakeholders (e.g., system-owners) to patch their own systems	Barrier	Security	Patch
		[F16] Decision-making through multiple levels of the organisation	Barrier	Operational	Time
		[F17] Need for certainty about responsibility and accountability for IT department and system-owners	Barrier	Operational	Time
	CP7 Communication	**[F18]** Inter-dependencies of different teams regarding time of patch event	Barrier	Operational	Time
		[F19] Dependency of external stakeholders (e.g., vendors, experts, media) as information source for vulnerability notification, patch criticality level, patch availability	Cause	Security	Patch, Time
	CP8 Coordination	**[F20]** Responsibilities and authority of decision-making within IT department (inter-team)	Barrier	Operational, Security	Time
		[F21] Responsibilities and authority of decision-making of IT department within organisation	Barrier	Availability, Security	Time
		[F22] Lack of centrally arranged patch information retrieval	Barrier	Operational	Time
	CP9 Procedures and guidelines	**[F23]** Lack of availability of information throughout patching process for central IT services	Cause	Operational	Time
		[F24] Lack of KPI's to indicate effectiveness of patch process	Cause	Security	Time
		[F25] Organisational restrictions of available patch moments (e.g., change weekends)	Constraint	Availability	Time
		[F26] Lack of defined process for patching of non-central IT services	Cause	Security	Patch
		[F27] Lack of clarity and standardization of emergency-response procedure	Cause	Operational	Time
Technical	**CT10** Patch quality	**[F28]** Side-effects of patch deployment can harm system's functioning	Cause	Operational	Patch
		[F29] Emergency patch often lacks proper testing by vendor due to hurry of release	Cause	Operational	Patch
	CT11 Patch availability	**[F30]** No patch available by vendor (zero-day vulnerabilities)	Cause	Operational	Patch
		[F31] No patch available by open-source software usage	Cause	Operational	Patch
	CT12 System dependencies	**[F32]** Lack of knowledge of system dependencies (e.g. layers, versions)	Cause	Applicability	Patch, Time
		[F33] Known dependencies of applications and databases of servers and networks (e.g., layers and stacks)	Barrier	Operational	Patch, Time
	CT13 Technical (hardware) resources	**[F34]** Hardware capacity limits frequency of patch deployment	Constraint	Operational	Time
		[F35] Large magnitude of patch releases	Cause	Operational	Time
	CT14 Complexity of systems	**[F36]** Large number of unique servers that all need different patches, applied manually	Cause	Operational	Time
		[F37] Large number of legacy systems that are 'unpatchable'	Cause	Applicability	Patch
		[F38] Lack of knowledge of functioning and criticality of certain systems	Cause	Applicability	Patch
	CT15 Usage of automation tools	**[F39]** Lack of integrated automation tool usage throughout phases of patch process	Cause	Operational	Time
		[F40] Need for 'human-in-the-loop' to assess applicability and relevance of patch release	Constraint	Operational	Time
	CT16 Asset overview	**[F41]** Lack of complete overview of assets	Cause	Applicability	Time
		[F42] Lack of complete information of known assets (e.g., owner, patch status)	Cause	Applicability	Time
		[F43] Dependency of departments in knowledge of change of system owners	Cause	Applicability	Time

[4] System-owners of faculty managed systems

4.2.1 Environmental context

Security patch management is a reactive practice; based on the trigger of a vulnerability, a fix needs to be installed to patch the gap. However, based on the recommendations in Chapter 3, there is a proactive step the organisation needs to take before the patching process starts. As Heiser states: “without knowing what is in place, it is impossible to know where the vulnerabilities are” (2003, p. 10). Asset and inventory management is needed to accomplish this. As it turns out, knowing all systems in the IT environment is a challenge. There are several reasons for this. First, the nature (and with that complexity) of the IT environment [F11] and how it is governed influences the level of knowledge about one’s systems. As indicated in section 2.1.2, many systems connected to the central network are systems owned by researchers at faculties. Not having a complete asset overview of system can cause a problem when it needs to be known on what systems a vulnerability is present, as P9 states [F41]: *“We supply a lot of servers that we do not use ourselves, but which go to researchers and faculties. For example, with the Log4j vulnerability, it took us a lot of time to find out which systems were affected”.* Lack of clear documentation when changes are made contributes to the trouble of knowing what systems need patches: *“You don't know which servers you have and who owns them, to be able to do a check on that server. You don't know if the documentation is still up to date”* (P10). There is a dependency on faculties to keep this information up-to-date [F43], which is experienced as troublesome. This results in many systems where it is not known who the owner is [F42], as P14 illustrates: *“It is often the case that researchers get servers, and then when they leave and don't tell anyone else, that server just stays there and runs. Then it will still be there in 10 years… there are still a lot of them running”.*

Additionally, legacy systems create a challenge to the level of awareness of what systems do [F38]: *“Every company has servers that are buzzing somewhere of which we do not know what they are doing. We don't dare turn it off. That is the reality, [our organization] is no different”* (P10). These systems are therefore in a state of being ‘unpatchable’ [F37] and can cause severe security risks. This is often the result of different strategies in the past and a classic example of path dependency, as P14 explains: *“That is because we have experienced significant growth in the past 5-10 years. It went so fast with servers that were being rolled out, then you are much more concerned with the operation of all the servers, than with how to organize and administer it”.* These examples illustrate the high dependency on the environmental context organisations are in, whether it is the current organisational structure that influences the accurateness of the asset overview or choices made in the past that influence the decisions to be made now.

4.2.2 Phases of the patching process

Decisions need to be made throughout the entire patch lifecycle, from information retrieval until post-deployment patch verification. The phases of the general patch process, as discussed in Figure 13 of section 3.3, are used to structure this section into sub-sections.

4.2.2.1 Patch information retrieval

The patching process starts with a trigger, a vulnerability notification. In this phase, IT practitioners learn of the existence of patches and acquire them from vendors of their software products (Dissanayake et al., 2022). Recognising what is potentially relevant for the organisation could be difficult [F8, F40] and remains a human task: *“In any case, the signal function, of course, if no one does anything with it, or someone makes a wrong estimate “we don't need anything with that”. It's still people's work”* (P9). There is a wide variety of information sources that IT practitioners can use to become aware of vulnerabilities. From the analysis, it is found that different information sources need to be combined in order to have an accurate overview [F19]. Table 7 provides an overview of the different sources stated by participants.

Table 7 - Sources of vulnerability notification

Sources of vulnerability notifications	**Type**
Software vendors (notifications, satellite servers)	Re-active
Advisory publications (governmental mailing lists, NVD, RSS feeds)	Re-active
GIT-repositories	Re-active/Pro-active
Media (Twitter, Tweakers, blogs, newsletters, webpages, personal networks)	Pro-active
Vulnerability scans	Pro-active

The most 'centralised' information source for different operational teams in the IT department is the security team, as P2 states: *"We have a separate security team within ICT, which receive that kind of information from all kinds of government institutions, companies, you name it"*. The security team additionally performs scans to pro-actively check for vulnerabilities: *"We scan our environment every once in a while. If vulnerabilities are revealed, we direct to the teams via ticketing what needs to be patched"* (P14). The security team therefore functions as a trigger for operational teams of the awareness of a vulnerability. However, in addition a lot of activity takes place within the operational teams. An example is the pro-active information gathering on internet forums and blogs, as P9 states: *"We have a lot of colleagues that are scouring the internet, or are members of forums, or follow tweakers. Recently something came up on tweakers, I sent it to an administrator, who would then discuss it in his team"*.

For some systems, a re-active automatised way of collecting information is through synchronising a server with the software vendor, as P1 tells: *"All we need to do is approve the patches on the WSUS[5] server. Because you automatically download them, which syncs with Microsoft twice a day"*. Similarly, P2 from another team states: *"Most of the servers we use centrally are all [company X] servers, which are linked to a 'satellite server'. You register such a server on the satellite server, and then it gets all its updates from there"*. This shows that information retrieval is done both re-actively and pro-actively and information is gathered in an ad hoc, unstructured way. The points where information is received lay spread out throughout the entire IT department, from the security team to operational IT practitioners. The lack of a centrally arranged [F22] information retrieval strategy makes it therefore more intensive to collaborate efficiently.

4.2.2.2 Vulnerability assessment and prioritisation

A risk estimation of the vulnerability determines the actions needed, as P12 states: *"An estimate is made, an analysis based on the security risk. There it is checked whether immediate action is required"*. Several elements are of importance, such as the criticality of the vulnerability and the potential operational risks it brings: *"When you look at patching, the question is always: how important is it? How quickly should it be implemented? And what is the impact of implementing or not implementing?"* (P10).

As one of the recommendations identified in Chapter 3, often the CVSS or CWE is used by vendors and organizations to rank the importance. The CVSS score, on a scale from 0 to 10, represents the severity scores of a certain vulnerability and is based on three metrics: Base, Temporal, and Environmental (NIST, 2022c). For example, one of the components of the Exploitability Metrics is the Attack Vector. The Attack Vector represents the 'level of access required to exploit a vulnerability' (Balbix, 2022, par 9). The value of this vector will be higher if an exploit can be executed remotely than if it requires physical presence (Balbix, 2022, par 9). The Impact components of the Base metric and the Requirements components of the Environmental metric are based on the CIA-principle, as discussed in section 3.1. This score determines the speed of which actions are needed, P1 argues: *"That score is from 0-10 based on a number of factors. The higher that score is, the faster you have to work. If that score is a bit lower, you might want to rest a little more"*.

[5] Windows Server Update Services

Often such a score is not available and not enough information is available to make an informed decision. In such cases, IT practitioners need to assess the potential impact themselves, sometimes in consultation with the security team and information from the software vendor: *"The threat is really the most important factor, based on information from the supplier, and whether or not in consultation with the security officer"* (P7). Another criterion taken into account in this consideration is whether it is being actively exploited, demonstrating the importance of media: *"Whether the vulnerability has already been seen in the wild, whether it is already being exploited, yes or no. Or that there is a 'Proof of concept', if it has already been described how it can be exploited, yes, then we have to get started quickly"* (P1).

Furthermore, such self-assessment is not only done because of a lack of information but also because those scores are general and not specified to the unique environment organisations have [F36]. The location of the system and whether certain functionalities are used play an important role: *"However, the score does not say everything: depending on the purpose of the system, depending on where the system is located or how it is protected"* (P14). As P7 argues this is an important element of consideration before deploying a patch: *"But we certainly won't roll out everything right away, it depends on whether we have that functionality"*.

It is eventually a consideration between security and operational and availability aspects. One important operational risk is the chance of unwanted side-effects of patch deployment [F28]. P1 provides an example of the considerations during decision-making: *"A security patch can also have side effects, it can destroy things. Then the question is should I wait? Am I going to see what other companies are doing? Will I follow Twitter? Will it outweigh the risks I'm going to take? Do I take that for granted, or do I just wait for the vendor to roll out a new patch that doesn't do this? That is always a consideration that you make"*. Here it is indicated that the knowledge of what other organisations are doing is useful to dodge side-effects that are present in the initial version of a patch.

Next to operational aspects, business requirements of top-level management and clients play an important role [F12], which are often mainly focused on availability and continuity. This can result in the decision to delay a patch to a later change moment and 'save up' patches. This brings consequences for the state of security, as P12 illustrates in an example: *"For continuity, we don't install patches unless it adds functionality or fixes a problem. If it doesn't do both, then we won't install a new version. Then it is nominated to get on the six-month list or if we have a maintenance moment and that server is down anyway. There is a risk to that, the moment that list gets too long, and you get a security issue where version 10 is at risk, and we are still on version 8, and you have to patch immediately. Then you would have to apply more patches, which means that the maintenance is even greater"*. External pressure plays a part here as well, as P13 explains: *"But we also often patch to keep our support with the suppliers. At the slightest thing that goes wrong, they say "yes, go ahead and apply those patches, and then we'll talk to you again". But that's also a risk, of course, to lose that support, apart from hacking"*. Software vendors thus have some degree of power to stimulate consumers to apply their patches.

In some cases, however, the need for decision-making is removed at all by the choice of patching everything that is available. When asked what determines what to patch and what not, P3 replied: "*Everything, they will not release it for nothing, there will be either a bug fix or a security patch"*. All patches are saved over the course of a couple of months, until the change weekend. This is done to ease the workload of needing to assess every single patch and find a balance between patching and other day-to-day tasks [F10]. In other cases, decision-making is influenced by not having patches available [F30, F31]. With very critical vulnerabilities, software vendors often do not have enough time to release a patch directly: *"With Log4j it was going on 'at the moment', but I didn't have the patch itself until a week later, and another patch a week after that"* (P4). This results in the need to make other decisions, for example instead of mitigating risks, avoiding the risk by isolating systems.

The decision of what to patch and when to patch is mainly made in collaboration with many different teams [F13, F16]. Most common system administrators and security officers: *“The people who are responsible for a service, they will check whether we are eligible for that patch, they will estimate it. Whether or not in consultation with the security team”* (P8). When a change needs to be made, the Change Advisory Board (CAP) needs to approve the decision: *“Yes, we have a division between change coordinators and the CAP. In general, a change goes through either one, or both. Depending on the severity of the change. An emergency change goes through CAP, but is communicated through Telegram, and approval is requested”* (P8). This CAP consists of representatives from all teams in the IT department, resulting in a collaborative perspective on the decision.

Based on the importance and impact, a timeframe for when to patch is indicated. This can be either in a timeframe of a couple of hours to a couple of days (i.e., an emergency patch) or in a couple of days to a couple of weeks or months (i.e., a routine patch scheduled for a change moment). As P14 indicated with an example: *“These servers all have to be patched between now and two days because the vulnerability is so big”*. The determination of a patch moment is however limited by organisational procedures [F25]. There are fixed moments where changes can be made due to restrictions that secure availability for end-users: *“What we have to do is outside office hours, we are limited by our standard service windows, and a large service window four times a year”* (P7) and *“Can we wait until the next maintenance moment? Because if we suddenly throw all those servers down and people don't expect that…”* (P1).

When multiple patches are released simultaneously, prioritization will determine which to apply first. This is an ad hoc process and is determined per occurrence: “*There is nothing on paper, that is a matter of inventorying at that time whether and what dependencies there are*” (P14). The main aspect is the dependencies between systems. If no dependencies are making certain patches ‘clash’, everything is patched at once due to resource limits: *“Everything at once. That would take way too much time, that's not possible”* (P3).

Based on the assessment of the vulnerability, several strategies can be carried out, illustrated in Figure 16 in a flow diagram. There are four possible actions to be taken when dealing with vulnerabilities:

1. **Emergency patching**: Patch directly if the severity is high and applicability is true, if a patch is directly available
2. **Routine patching**: Plan patch for a later moment if the severity is low and applicability is true
3. **Emergency workaround**: Take mitigating measures that do not involve patching if the severity is high, applicability is true, but no patch is directly available and waiting is not an option
4. **Do nothing**: Take no action if applicability is false

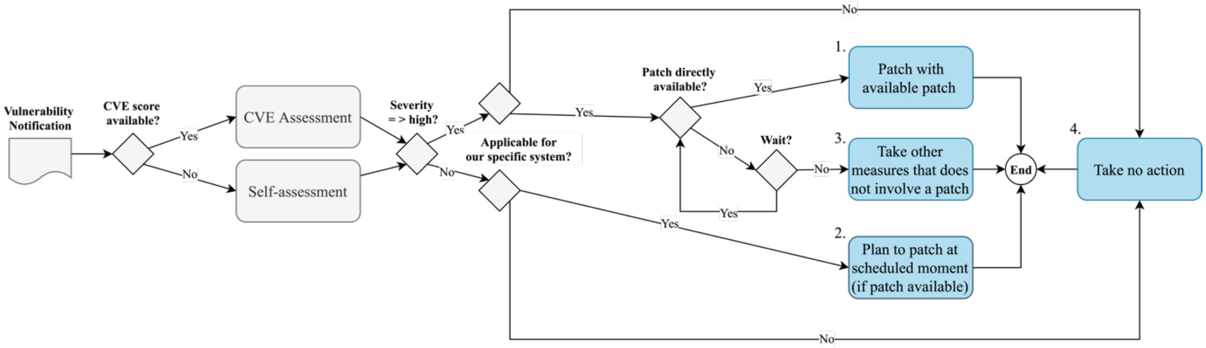

Vulnerability Notification
CVE score available?
Yes
No
CVE Assessment
Self-assessment
Severity => high?
Yes
No
Applicable for our specific system?
Yes
No
Patch directly available?
Yes
No
Wait?
Yes
No
1.
Patch with available patch
3.
Take other measures that does not involve a patch
2.
Plan to patch at scheduled moment (if patch available)
End
4.
Take no action
Yes
No
No

Figure 16 - Flow diagram of multiple scenarios

4.2.2.3 Patch testing

Once the decision is made based on security and a patch is scheduled for deployment, the patch will be tested for accuracy and stability within the context of the organisation. Most important of all, it is checked if everything is still working if the patch is deployed: *"We will look at 'are things falling over?', 'is the [exchange] still working?', 'is the exit directory still working?', 'are we suddenly encountering unpredictable things?', 'do things suddenly react differently?' or 'have we seen errors?'"* (P1). Not testing properly can have severe consequences, as P1 states: *"If it turns out that a patch is not good after all, then it is important to see what the problem is. There are often 100, 110 patches that are ready. And if that error does go through, then you have 1000 servers that have problems"*. Testing is done in a test-environment to prevent side-effects to occur in the production environment. To maximize the value of testing, the environment to test needs to be as similar as possible to the production environment: *"It's all a bit scaled-down. It is a small-scale model of what the production environment is like, but in a way that it is representative of the production environment"* (P1).

Because of time pressure, testing needs to be carried out quickly. Unfortunately, there are side effects that cannot be discovered by testing, as these develop in the after days of patch deployment, much later than the timeframe the system is tested for. This influences the decision to delay patch instalment and wait on the confirmation of other organisations. P1 explains the reasoning behind this: *"Imagine on Exchange, email, a security patch is released that initially seems to work well, but it destroys something underwater that runs once a week, for example, updating an address book. So you only know that after X number of days. Then the decision is, should I roll over right away or wait for other sources; if people don't complain, or if they don't break anything. You wait"*.

While testing is in general beneficial to minimize operational risks, it does cause a delay in the process of deployment, explained by P12: *"That is of course also a delaying factor. It takes quite some time, which is often a problem. The moment you have to roll out the patch next week, you won't have all the resources available to do that next week"*. Testing is thus costly and resource-intensive [F34], making it not feasible to test all organisation's systems due to a shortage of test equipment. This is sometimes managed by the assumption that patches are tested by the vendor before release. The reputation of the vendor plays a role in the level of trust: *"That is the advantage of a vendor like [X], before the patches are released at all, they are all thoroughly tested"* (P2).

4.2.2.4 Patch deployment

For routine patching of some systems, the organisation follows the cycle of the vendor (e.g., Microsoft). This happens every month, following a standard routine: *"The patching itself occurs every month. We have the Microsoft, of course, every second week of the month, every second Tuesday of the month, I should say more specifically"* (P1). For most systems in the organisation, routine security patching on centrally managed services happens once every three months: *"What we do every three months is what has been saved in patches over the past few months"* (P2). The vast number of patches to be deployed in a change weekend make it a big operation [F35]: *"Sometimes you have to deal with the size of the patching. If we have to do 500 switches, sometimes you have to take longer. Throughput time can be an issue"* (P7).

Interviewees, however, wish to do everything more frequently. The main incentive to do so, next to security, is to make it more manageable: *"It could be that ICT will schedule six/eight maintenance weekends per year, which would make it easier for us to patch security and releases in a shorter period of time. Ideally, we would have a weekend once a month, which is manageable in terms of work, and manageable in terms of loss of availability"* (P3). Additionally, it will benefit the awareness of end-users if there are frequent and standardised patch moments: *"The more maintenance weekends you have planned, the less you have to communicate about it, and the more standard it is for the users*" (P3).

The patch moment is bound to the availability and continuity constraints of end-users [F14]. End-users cannot be interrupted simply because a patch needs to be installed, as P4 argues: *"In 2022, you can no longer just say to your customer "I will turn off your system now because I am going to update""*. Sometimes these constraints are not a matter of days, but a matter of weeks or months: *"Suppose there is a certain research project ongoing, and they are doing a measurement and they are busy for one week or one month, then I can't do a reboot, otherwise your research results will go to waste, then they won't be happy either"* (P7). A solution is to equip systems with High-Availability (HA) [S9], meaning that if one server is out, another takes over: *"A number of services with High-Availability have been implemented, you can remove one server from them, patch it and restore it again"* (P2). However, hardware resources play a role here, which limits the timeliness of patching [F34]: *"In production, I can't suddenly shut down ten Exchange servers because the other ten servers aren't able to pull it. So it is always per two, three max"* (P1). Next to hardware resources, human resources are limiting the capacity of patching [F10]. IT practitioners are involved with many different tasks, where patching is just one of many: *"Our portfolio is quite large, if the need is not there, we don't do it. I've got my hands full already"* (P4). More manpower could increase the frequency of patching [F9], P2 argues: *"We want to take steps to do this a lot more frequently. The only problem is that we need a team of about 10/15 people to organize and arrange it all"*.

The deployment of a patch itself, once the decisions have been made, is not that troublesome: *"The patch itself isn't very exciting, there isn't that much going wrong either, that's not really a limiting factor"* (P2). Most patching is deployed automatically. What is done manually is the stopping, starting, and checking of the applications and servers: *"Patching is already 95% automated, for us, that is only a matter of getting started"* (P2). This manual work does make the duration of the patching process longer [F39].

Once systems are patched, a check is carried out to evaluate if it was successful and whether certain actions may be needed: *"…to check whether everything is still running, everything has turned up well, or do we have to manually give things a push? Did some patches go wrong?"* (P1). It can be that there are unforeseen side-effects during deployment, resulting in a patch not working. This should be solved within the same change moment: *"Sometimes you find that things that you thought worked didn't work. In a maintenance weekend we then have two days. We try to patch on Saturday, and have time to fix issues until Sunday"* (P3).

It is key to consider technical dependencies before deployment [F32, F33]. There are dependencies between different systems and their versions. Some dependencies hinder the possibility of deploying another patch, as P4 illustrates: *"Not every version of one can work with the other version of another. For example, if [X] doesn't have an update, we can't actually do the layers below it either"*. Additionally, there are dependencies between software applications of different vendors: *"It is not always the case that supplier [X] can work with the latest version of [software of supplier Y], we cannot handle that with some systems"* (P12). It can also be the case that the deployment of one patch causes another system to not function properly anymore, as P4 explained while showing the patches deployed over the previous months: *"This one we needed to downgrade, because we updated this system"*.

Apart from hindering other patches, technical dependencies can also increase the technical complexity of patching itself, making it difficult to do quickly: *"An application or service is often a combination of different systems. When rolling out such a patch, you should know very well which systems are connected to each other. The limitation is the complexity in technology, which makes it difficult to roll out patches quickly"* (P12). This forces clear communication with other teams. These inter-dependencies make coordination even more important, which can be an issue sometimes [F18]: *"Sometimes there are certain dependencies, for example SQL, which runs on a Windows server. We manage that windows server, but SQL is managed by the database management department. So if something has to be installed on it, and we have to reboot that server, then I have to consult with the people at that department"* (P1). Another factor that makes this difficult is the uniqueness of systems [F36]. This results in the need to keep patching manually: *"Everything is a bit* *of a 'special' with us. We have 1500 servers, almost none of which are the same. As a result, you will always keep the manual work"* (P2). Here the context of being an educational organisation is even more significant: *"A microscope in a laboratory that cost hundreds of thousands of euros to set up in some cases simply cannot be upgraded, because that directly affects the entire measurement setup"* (P14).

Finally, human errors can cause problems in patch deployment [F7]: *"Sometimes, there are people who also make mistakes. Errors can be made during the rollout of such a patch. When a patch says that everything went well, but was not able to do everything"* (P1). Following the procedure right can be difficult when under time pressure: *"So on the one hand, occupation, on the other hand, following the procedure of the patch. That will probably not be read in a hurry. It remains important to be secure and careful"* (P1).

4.2.2.5 post-deployment patch verification

After deployment, the patch is monitored, and it will be verified that no issues occur. The main aspects to be checked is whether the vulnerability is still present: *"You know what the patch is for; then you can check whether that incident still occurs, yes or no. Or are they still vulnerable?"* (P1), and whether everything is still working: *"We have a chain test: can I log in? Can I do everything I have to do? Only then do I move on to the next one, and if that doesn't work, switch the old one that isn't patched up again. Then do the chain test again. Hey, now it works? Then there is something wrong with that patch"* (P1). If deployment is not successful, backups can help to restore the old situation: *"You always have a backup, so that if things go wrong, you know you can go back. That is a fall-back scenario"* (P4). To deploy this verification step, tools supplied by the vendor can be used: *"In the case of Exchange, Microsoft makes a certain checker tool with which you can check whether your server is up to date, or whether components are missing"* (P1). That is however not the case for all systems, as P2 states when asked about verification: *"That is not checked. Then something would have to break before we notice it. That hardly happens"* (P2). This shows that the services vendors offer enable the ease of post-deployment verification.

4.2.3 Organisation's procedures and processes

As indicated in the earlier sections of 4.2, many choices in the process of patching are influenced by procedures and agreements with different types of stakeholders. These factors play a role throughout each of the phases of patching.

4.2.3.1 Agreements on ownership responsibilities

As indicated in section 2.1.2, there are distributed ownership and decision rights in the organisation. System-owners are responsible for patching their own systems: *"We make management agreements: what can the person do with it, what do we do, who is responsible for what. Patching is done by the users themselves"* (P8). This is very similar to how software vendors shift the patching rights to their consumers. Except, a software vendor is less dependent on the behaviour of its consumers than the IT department is on its clients and system-owners. The problem with this, is that patching of these systems could take even longer. System-owners are even more keen on keeping a system up and running as their application or research data is on it [F6, F15, F26]: *"We see that tendency taking place, that things are being pushed forward, and continue to being pushed forward. But it has to be 'ready' at some point. If there are agreements, they must be kept. If not, we should say 'we'll take action'"* (P14). This is also because of lack of threat perception and lack of perception that patching reduces that threat [F3, F4]. Even if these systems owners have the best intention to do patching right, it can be difficult to deploy patching due to lack of knowledge [F5]. It is currently difficult for the IT practitioners to reach and provide advice to all owners in time: *"Since almost every server has a different owner, you have to write to and push all owners to upgrade, update or replace their server. There is a lot of work to do"* (P2).

Relevant here is who the 'risk owner' is, or who is allowed to take a risk. The IT department feels responsible for the entire organisation. If a data breach happens through one of IT's systems, it will affect the reputation of the entire organisation. System owners of faculty-owned systems feel less pressure and are more in their own bubble. In some cases, IT would like to endow that risk in a more formal way. In this way, by formally putting the risk with someone else, it might be an external incentive for them to mitigate the risk: *"I can imagine that the system owner is not at all capable of accepting that risk. That risk can be so great that it affects a department chair, or in some cases a dean, or even the Executive Board. In that case, the Executive Board must put a scribble under it that he/she accepts the risk"* (P14).

Apart from the centrally managed systems, pre-defined rules and guidelines need to be established for the systems managed by faculties: *"Most importantly, we need to have a clear process. That that process is used for our own systems and the systems of our researchers. We have to organize and record that well"* (P9). It is not clear what the mandate of security is, and how far it reaches, when asked P14: *"Good question, we often ask what our mandate as security is… Can we at some point quarantine systems, shut them down completely, or... at any moment? It's still a grey area, we have a mandate if there is a real need to shut down systems. But that really is in an incident where we see that something is really wrong at the moment, then we can act immediately"*.

4.2.3.2 Agreements on processes and authority of decision-making

Throughout the interviews, participants mentioned the lack of pre-defined rules and guidelines as a troublesome factor in coordination. Although change processes are in place, it seems that patching is still done in an ad hoc way: *"What you often see is that you gradually start planning all kinds of consultations… what is the effect of it? What can happen? But you actually have to do that in advance"* (P7). This forms obstacles when decisions need to be made quickly, as stated by P7: *"I have to make a decision very quickly. Then at that moment, I should not have any obstacles from the change process and the dependencies with the other environments"*. The procedures that are currently in place are fragmented throughout the organisation but not centrally coordinated or documented: *"There is quite a lot, but it is fairly fragmented throughout the organization. There needs to be a little more unity. A bit of standardization is missing now"* (P9). Lack of unity makes that different teams operate on 'islands', increasing the distance between them: *"Nothing has been arranged centrally, and that is exactly what we want to set up. There must be a central patch and release management process. Now everything is just decentralized"* (P9).

The lack of well-established ownership and roles and responsibilities within the IT department is forming an obstacle [F21]: *"If we receive a signal, and it is clear to me that such a system or application must be switched off, but then I have to 'shop within the department' to see who says yes and who says no. I deal with teachers, students, business operations. And sometimes there are conflicting interests. Students want to take their exams now, but it is unsafe 'now' as well, so it has to be done 'now'. Ownership, that could really be clearer"* (P12). Decision-making rights of individuals or entire teams are not established [F20]: *"Who can decide something, who can say you are going to do this and this. Who gets to decide, that server that's infected, I'll take it off the network?"* (P10), as currently it is an informal process of getting approval: *"Don't go begging: we have a situation, can I take him off the grid? No, then you must have the authority to disconnect, a CISO, director, whatever"* (P10).

Apart from coordination and decision rights, communication can be a threshold as everyone needs to be aware of a change as the dependencies as so high: *"Communication is a threshold, and the time when we can implement the change. You have to go through the procedure very carefully, inform everyone"* (P7). Internal communication is needed to inform all team members of what will take place and when: *"Everyone should be aware of it. You have to arrange communication. You must have people within your own team ready, "when are you going to perform it, during the week or at the weekend?". You should also inform your customers of this"* (P4). External communication is critical to inform end-users and clients about patching moments: *"If there are 4 or 5 thousand people on weblogin, then I have to inform those people. I can't just pull that whole area down. I have to inform, otherwise a lot of tickets will come, or complaints"* (P1).

Pre-defined processes and procedures will help speed the process up as fewer discussions are needed to come to a decision: *"90% can be pre-arranged in a pre-defined document. If everyone signed it, accepted it, that it's clear that this is what we're going to do. Then you have fewer discussions for such moments to get to the point faster"* (P7). Procedures should clearly include the different owners of systems and other relevant information such as dependencies: *"Part of the solution is working more process-oriented. Then you can say very clearly, this process has this owner. When I turn this off, I hit that owner, then I know exactly how it is connected. We do that within ICT already, but it is not written down anywhere. What exactly do I hit when a component has to be upgraded, and the application is down?"* (P12). Simple documentation of stakeholders and who to involve for what can be beneficial to the speed of decision making, especially in states of emergency [F23]: *"A call schedule, a stakeholder overview, who am I calling for what, who is allowed to communicate, who is allowed to communicate with whom, to the outside world, internally, that sort of things. You have to think about it beforehand. If you get a 'Maastricht situation[6]', then there is panic, sure, but then you have to be able to pull it off the shelf. Even then, mistakes will be made, but at least you have a guideline"* (P10). Furthermore important is that authority in decision-making is documented. It should be clear what is prioritized over what and in which situations [F17]: *"Security comes first. But that is not marked down anywhere. I can imagine if [service X], which is currently being used extensively, if there's something with it, I'll have it turned off, but that does stir up a lot of dust. There is no guarantee that I could do that"* (P12).

4.2.4 Differences in the state of Emergency patching

As indicated in section 4.2.2, at some moments patching must be done quickly causing the IT department to be in a state of emergency. When it comes down to emergency patching, the factor of time becomes even more crucial. A difficulty here is that it can happen at any moment, as P1 states [F2]: *"I think the hardest part is the unpredictability"*. This causes the need for IT practitioners to always be available: *"For really big leaks, like Heartbleed, where you have to act immediately, then we also have to take action outside office hours. Then it depends on who is available within the team at that time"* (P3).

The frequency of emergency patch releases is: *"Very variable, sometimes three times in a week suddenly a critical security patch that we have to do something with. And sometimes it's quiet for months"* (P9). For other systems it is even less frequent: *"In addition, if there is a gap, we will do so immediately, but that is no more than once a year"* (P3). Throughout the interviews it emerged that no standardisation or emergency-response procedure determined when the state of emergency is reached [F27]. For example, no procedural choices from top-level management are made as to determine this speed of deployment: *"No, it was actually only said from the organization, if a critical patch comes, it should be installed as soon as possible"* (P1). The determination of it being an emergency patch is done ad hoc and is very dependent on the situation: *"Then we really have the colleagues here next to the desk, how are we going to handle it? Can we patch it tonight?"* (P14). When it is present, it is 'all hands on deck' to patch as soon as possible. However, as this is not documented, 'as soon as possible' is a variable concept itself: "*Do we do it urgently, within a few hours, or within a short period of time, within a week?"* (P7).

When the state of emergency is reached, other work is less prioritized in this case and is set aside: *"Things that fall over as a result, such as other work, is then put on the long track, that must then be paused"* (P1). The sense of urgency makes it differ from normal change processes. Regarding the decision-making in the process, no other decisions need to be made, the same decisions just need to be made in a shorter timeframe [F1]: *"A normal change would normally follow the change process. An emergency patch on the other hand, will cut right through it. The focus is on as fast as possible. That means that people who have to give approval*

[6] See section 2.1.2 for explanation

within the change process should not drop everything they are working on right away, but that they should get through it within 2/3 hours maximum. There's a bit of enforced priority there" (P10).

In these situations collaboration is crucial, and discussions and consultations are still needed: *"We cannot patch directly, that must always be done in consultation (management, information manager). We will have to convince them that it is very urgent, they will understand, and in the end, it may not be two hours later that it happens, but it will be that evening. As long as it is as soon as possible"* (P14). The CAP is involved in making the decision in the emergency-change process, as part of an emergency-change process: *"Yes, there is an emergency-change process, and that includes the decision by the CAP, always. But the lead time is considerably shorter. That is now set up in such a way that if a colleague makes an emergency change, it can indicate it wants to have a decision before that and that time"* (P8).

Patching itself happens mainly manual, simply because no mechanisms are in place to do it any other way, as P1 explains: *"Then they release an out-of-band patch and when that happens, then it's all hands on deck and then we actually have to patch everything by hand the same night that the patch comes out"*. That is to say, if there is a patch available [F30, F31], making the dependency on software vendors high: *"Very occasionally it happens that something comes out and that nothing has been published at [vendor X] yet. And then you have to wait"* (P3). If a patch is available, quality assurance cannot always be guaranteed as patches are developed very quickly [F29]: *"Such a patch is rolled out in a rush, and then they [vendor] don't have the time to do the quality checks for that patch"* (P1). This increases the operational risks and probability of side-effects: *"But if a leak is found that needs to be repaired quickly, they don't have time. And then you have the chance that something will break quickly"* (P1).

4.3 Overview of decision-making in the patching process

The previous section 4.2 indicated a wide range of socio-technical factors and challenges that influence how and what decisions are made. Looking at the decision-making process from a higher level of abstraction is helpful in understanding the emerging aspects that influence decisions (section 4.3.1). Furthermore, it is interesting to understand in what way factors influence the decision space of IT practitioners. Additionally, the quotes in section (4.2) imply that there are tensions between different aspects, where a balance needs to be found while making decisions. How this balance is found and how is dealt with these factors can help understand the decision-making of IT practitioners even further, as section 4.3.2 will discuss.

4.3.1 The funnel of patch decisions

Although IT practitioners need to make a multitude of decisions, the main collective decision to be made in patching is twofold; to apply a patch or not, and in what timeframe. Four aspects play a significant role of influence. First the *security aspect*; the significance of the threat and the impact of not implementing the patch. As patching is a pro-active security control, this is the first aspect that triggers the consequential decisions to be made. Secondly the *applicability aspect*; the eligibility of the vulnerability on the nature of the organisation's system. As the security aspect might trigger the assessment of a vulnerability, the applicability aspect is a threshold for the consequential decisions to be made from that point on. Thirdly the *operational aspect*; the impact of implementing the patch. This consists of the consequences for the IT department and the risks of deploying the patch itself. Fourthly the *availability aspect*; the impact and the consequences it entails for clients. Bringing these aspects together results in the open *funnel* illustrated in Figure 17.

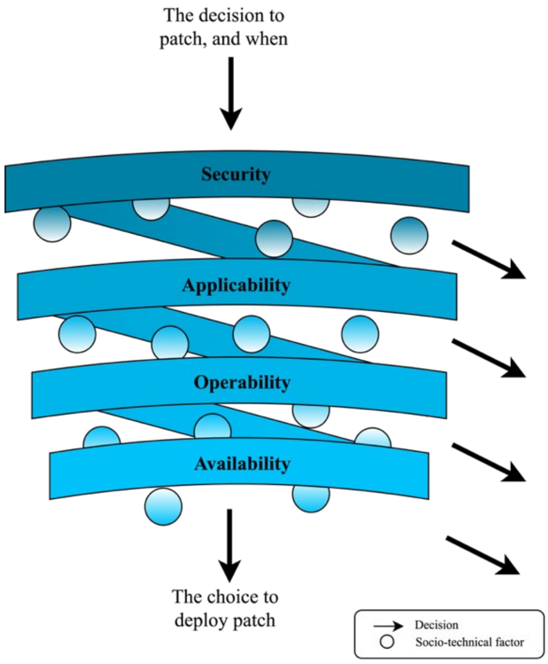


Figure 17 - The Funnel of patch decisions

The different aspects of security, applicability, operability, and availability represent the different levels of the funnel. Input of the funnel are the decisions to be made throughout the patching process, most prominently the 'decision to patch, and when'. The balls in Figure 17 represent the socio-technical factors that influence the decision space of the funnel. These are related to the four aspects, as indicated by the different shades of colours. The decision-making process is influenced by these different socio-technical factors, these are either;

- **Causes** that *shape* the decision space
- **Barriers** that *hinder* the decision space
- **Constraints** that *limit* the decision space

Socio-technical factors, therefore, impact the outcome of the decision and the time it takes to take a decision. These socio-technical factors are both on an individual level of IT practitioners and on a higher level of the organisation. Before having a result of the decision-making process (having determined a moment when to apply a patch), the decision has gone through the entire funnel. However, as illustrated, the funnel is open, wherefore the decision not to apply a patch, or postpone the decision to a later moment results in falling out of the funnel, as indicated by the arrows on the right.

Table 6 indicated all socio-technical factors and their type, the aspect it has the most influence on, and what part of the main decision it impacts. To demonstrate how to interpret this table, several examples provided here. When taking an example of [F1], it can be read as follows: The 'decreasing timeframe between vulnerability awareness and exploit by threat actor' is a cause for the 'decision of the timeframe' from the 'security aspect'. As the impact of not implementing the threat will increase, the timeframe to patch is desired to be shorter as well. Another example is [F28], the fact that 'side-effects of patch deployment can harm system's functioning' is a cause for the 'decision to patch' from the 'operational aspect'. If two aspects are indicated, it means both are equally relevant. For example [F18], the division and different levels of 'responsibilities and authority of decision-making of IT department within organisation' hinder the 'decision of the timeframe' from both the 'security aspect' and the 'operational aspect'. As not everyone has the authority to make a decision on the significance of the threat, this needs to be approved by other teams, e.g., the security team. Similarly, as not everyone has the authority to make a decision on the availability consequences for end-users, this must be approved by other teams, e.g., change coordinators, or the CAP.

4.3.2 Tensions and coping strategies

Table 6 seems to conclude that security patching is insanely hard, if not impossible, and yet, patching still takes place. The question then is, what are the coping strategies, or ‘success factors’, to make patching work? During the security patching process, the aspects of influence (i.e., security, applicability, operability, and availability) are equally important, and a satisfying balance needs to be found between them when making decisions. This section discusses the coping strategies used by IT practitioners and explains how tensions are managed successfully. Furthermore, the effect of being in a state of emergency on the tensions is discussed.

Tension 1. Security and Operability

When it comes to taking risks, there is always a certain risk acceptance level. Some people (and organisations) are just more risk-averse than others. The risks of security are in friction with the operational risks of patch deployment.

Regarding the security risks, there are certain strategies that make these more manageable. The usage of multiple channels during information retrieval (S11) creates more awareness of the security risks present. Furthermore, one of the coping strategies is the knowledge that not all vulnerabilities are exploited. This makes it feasible to wait for a moment and see what other organisations are doing (S14). Secondly, there is a certain level of risk acceptation to wait with patch deployment based on a number of characteristics of the nature of the system (e.g., is the system behind a firewall, is it a critical service, does it have a high level of dependency, is it covered with HA (High-Availability)). The strategy of taking mitigating measures (e.g., isolation) (S15) when patches are not available help to decrease the security risks posed by a vulnerability.

Regarding the operational risks, a coping strategy is to properly test a patch (S13). If errors or side-effects appear in the testing phase, changes can be made to the configurations, or the decision can be made to not install the patch but take mitigating measures. Additionally, human errors in patch assessment are being reduced by letting this be done by multiple people from different teams (S4). Additionally, the strategy of patching filtering on relevance (S16), helps manage the size of hardware resources needed.

Following this, a level of acceptance is present in the sense that ‘all organisations are struggling with this, it’s not just us’. The coping strategy here to accept the situation (S0), brings a sense of ease to the practice of security patching. The coping strategy of informal ‘decision-making’ on the go (S8) helps deal with the lack of standardized operational procedures.

Tension 2. Security and Availability

Security compromises availability in the short term. In order to patch (i.e., increase security), availability (and continuity) of systems need to be compromised due to the need to restart and update the system. This causes a short-term disruption to business continuation of end-users’ work. However, security increases availability in the long-term. When security measures are not implemented and vulnerabilities are not patched, the probability of being exploited increases. A higher probability of being exploited results in a higher probability of availability and continuity disruptions. Where the objective for security lies mainly within the IT department, the objective for availability and continuation lies mainly at top-level management.

Several ‘success factors’ are making this tension manageable. For centralized IT services, there are fixed maintenance moments weekly for small changes and quarterly for big changes (S10). In these moments, end-users are expected to be aware of the possibility that services are temporarily unavailable at these moments. With this strategy, the decision rights are drawn to the IT department. One can say that in these timeframes, security is prioritized over availability. In the timeframes outside these maintenance windows, patches are troublesome to apply. For those systems that are equipped with High-Availability (HA), active end-users are drained from one server to another, so they are patchable in sequence (S9). In this way, availability is not compromised by security. However, this is limited by hardware resources.

For non-centralized IT services, the decision on when to patch is in collaboration with the client. With the patching rights endowed on the client, so goes the power to enforce the decision of the patching moment as it is able to do with centralized services. Although the advisory role of the IT department is sufficient in most cases (S3), it can only come so far.

Tension 3. Money and Resources

In any organisation, there is a division of financial assets to be shared over different departments. More IT resources, including manpower and hardware resources, could lead to more effective and timely patching, simply because there is more capacity to do so. This tension influences the operability and availability aspect of the decision funnel. Unlike the other two tensions, this is not a balance but more a constraint. The dependency on the choices of top-level management makes that this tension is not navigated but accepted. The coping strategy here is to accept the situation as it is (S0).

Tensions in a state of emergency

The main aspect causing IT practitioners to be in a state of emergency is the risk a threat brings to the organisation. While the decision funnel itself is similar, the difference with a normal patch process is that both the impact and probability of threat occurrence are more significant. This results in the aspects and tensions having a different level of influence.

The probability of the threat is mostly influencing the security and operational aspect. When it is known that the probability is high, for example, when an exploit has been seen in the wild, security is prioritized. When taking a look at the earlier illustrated decision funnel, the funnel cannot be left based on the security aspect, as this is the trigger to get to a state of emergency. The influence of the applicability aspect remains equally important in a state of emergency, as a security threat that is not applicable has zero impact. Due to the prioritization of security, the chance of leaving the funnel at the operability or availability level is reduced in a state of emergency. The tensions are less significant due to some of the identified challenges being less troublesome due to the use of coping strategies (as indicated in Figure 18 by a smaller size of the socio-technical factors).

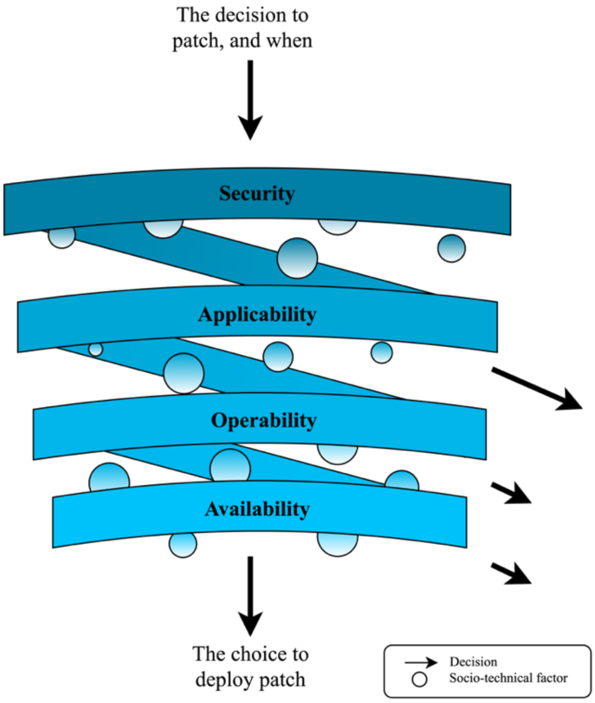


Figure 18 - The Funnel of patch decisions in a state of emergency

The coping strategy of a higher level of cooperation with an emergency patch (ES1) makes the tension between security and operability less significant, as the stakeholders with decision-power are all 'at the same side'. There is a clear company culture and awareness of the importance of patching in states of emergency (ES5). Additionally, having people on standby at any time (e.g., in weekends and off-work hours) (ES2) makes the influence of the operational aspect less vital. The tension between security and availability is reduced by the coping strategy of allocating the decision-making rights to the IT department in the state of an emergency (ES7). In rare cases, clients' systems can be cut off the network or set in quarantine until patches are installed, as a last resort. Additionally, the coping strategy of having the possibility to make changes outside the restricted timeframes agreed to if the risk is critical (ES12) reduces the tension between security and availability.

There is an organisation-wide emergency change process (ES6) that reduces the challenge of coordination as the timeframe to make a decision is forced to be shorter. Interestingly, however, is that there are no formal rules or regulations that declare the state of being in an emergency. Decision-making during emergencies is done in an ad hoc way, where the influence of other stakeholders plays an important role. Although the vendor has a certain level of authority when it comes to emphasizing the importance of a patch, the most crucial stakeholder is the media (e.g., blogs, forums, Twitter, email lists). When other organisations, security experts, or industry blogs report on the criticality of a vulnerability, the alarm bells go off. This indicates that the probability of occurrence is the main cause of being in a state of emergency. Although one could think of different reasons to be in a state of emergency, the most prominent cause relates to the security aspect. Reasoning from a business perspective, loss of productivity (e.g., slowing down of systems) or business continuity (e.g., end-users unable to access services) could be a potential reason for applying patches in an emergency state. This research did not identify any reasons not related to security, which might be caused by the framing of interview questions primarily focused on security. Table 8 presents an overview of the (emergency) coping strategies used for each of the identified challenges.

Table 8 - Overview of coping strategies

Category	Challenge	Strategies (ES = emergency strategy, S = strategy)
Organisational	CO1 – Developments of threat environment	ES1 – Higher level of cooperation with emergency patch ES2 – Having people on standby, even in weekends/ off-work hours
	CO2 – Behaviour and awareness of system-owners	S3 – Providing pressing advice and guidance on how to patch and why to patch
	CO3 – Role and capability of IT practitioners	S4 – Decision through multiple people (including different teams) ES5 – Clear company culture and awareness of importance of patching
	CO4 – Lack of Human resources	S0 – Acceptation of the situation
	CO5 – Organisational structure	S0 – Acceptation of the situation ES1 – Higher level of cooperation with emergency patch
Procedural	CP6 – Collaboration	ES6 – Organisation-wide emergency change process ES7 – Allocate decision-making rights to IT department in case of emergency S8 – Informal decision-making 'on the go' S9 – Equipping systems with High-Availability
	CP7 – Communication	S10 – Organisation-wide procedural guidelines and processes S11 – Usage of multiple channels of information (e.g., vendors, experts, media)
	CP8 – Coordination	S10 – Organisation-wide procedural guidelines and processes S8 – Informal decision-making 'on the go' S0 – Acceptation of the situation
	CP9 – Clear procedures and guidelines	S10 – Organisation-wide procedural guidelines and processes ES12 – Possibility to make changes outside restricted timeframes if risk is critical S0 – Acceptation of the situation
Technical	CT10 – Patch quality	S13 – Testing in test-environment S14 – Waiting on confirmation/response of other organisations
	CT11 - Patch availability	S15 – Taking mitigating measures (e.g., isolation, shutdown)
	CT12 – System dependencies	S0 – Acceptation of the situation
	CT13 – Technical (hardware) resources	S0 – Acceptation of the situation S16 – Patch filtering on relevance
	CT14 – Complexity of systems	S0 – Acceptation of the situation
	CT15 – Usage of automation tools	S0 – Acceptation of the situation
	CT16 - Asset overview	S0 – Acceptation of the situation

The overview indicates that there are multiple challenges where the coping strategy is to 'accept the situation' (S0), implying that part of security patching work includes passive elements. This acceptance is however not the same for each challenge, and certainly not all challenges where the only coping strategy is to accept the situation are unable to be improved. A distinction can be made between acceptance of the situation because of being highly unsolvable and acceptance of the situation because of the current organisational policies and resource limitations. Section 6.2 discusses the 'solvability' of challenges in more depth.

4.4 Conclusion and discussion of the decision-making process

Sub-question 2: What decisions and trade-offs are being made by IT practitioners in organisational security patching and what does this process look like?

This chapter aimed to explore how decision-making is carried out throughout the security patch process. The findings illustrate that security patching is a constant process of decision-making and weighting of factors, interests, and consequences. There are two main decisions to be made to which all other decisions contribute: to apply a patch and at which moment to patch. These decisions need to be made throughout the entire patch process and can be mapped as a 'funnel of decision making' that consists of four main aspects that influence the outcome of the decision: security, applicability, operability, and availability. A wide range of socio-technical factors tied to these four aspects either shape the decision space, hinder the decision space, or limit the decision space. Factors of the same nature can be grouped together, forming organisational, procedural, or technical challenges. The interplay of challenges causes tensions to occur during decision making, wherefore the decision outcome is very dependent on the situation. The main tensions are between security and operability, and security and availability. Additionally, there is a tension between money and resources. To deal with these tensions and the different socio-technical challenges, coping strategies are applied to decrease the significance of a challenge. The interplay inside the decision funnel can be summarised by the following points (illustrated in Figure 19):

1. Addressing one challenge can help to reduce the significance of another challenge.

For example, a lack of a complete asset overview and complete information of the known assets (e.g., owner, patch status) shapes the decision space of both the decision to patch and the timeframe to patch, from an applicability aspect (CT16). Here, addressing the challenge of establishing organisation-wide procedures and guidelines (CP9) can help reduce the significance of the other.

2. The existence of one challenge can cause another challenge to occur.

For example, because of the decreasing timeframe between vulnerability awareness and exploitation by the threat actor (CO1), vendors often release emergency patches without proper testing due to being in a hurry to fix a vulnerability. This causes the challenge of lack of patch quality (CT10) to occur.

3. As a coping strategy can reduce the significance of one challenge, it can simultaneously increase the need to address other challenges.

For example, the coping strategy (S4) of letting patch assessment be done by multiple people from different teams reduces the likelihood of a wrong assessment, therefore reducing the impact of the challenge of potential human errors (CO3). This, however, increases the involvement of multiple people, which increases the need for coordination (CP8).

4. As a coping strategy can reduce the significance of one challenge, it can be in conflict with another challenge.

For example, the coping strategy to wait on confirmation from other organisations (S14) helps reduce the risk of the challenge of potential side-effects (CT10), but at the same time clashes with the factor of the decreasing timeframe between vulnerability awareness and exploitation by the threat actor (CO1).

5. Challenges influence other challenges from the same category, but mostly from other categories.

It is common for challenges to influence other challenges in the same category as these are closely related. For example, the challenge of collaboration (CP6) impacts the challenges of communication (CP9). Because of the collaborative inter-dependencies, communication regarding the time of the patch event is needed. However, the socio-technical interplay really becomes visible through the interactions of challenges in different categories. For example, the decentralised organisational structure (CO5) impacts the significance of the challenge of collaboration (CP6). The synthesis of Chapter 6 will go into further detail about the significance of challenges and which of these socio-technical factors should be targeted first.

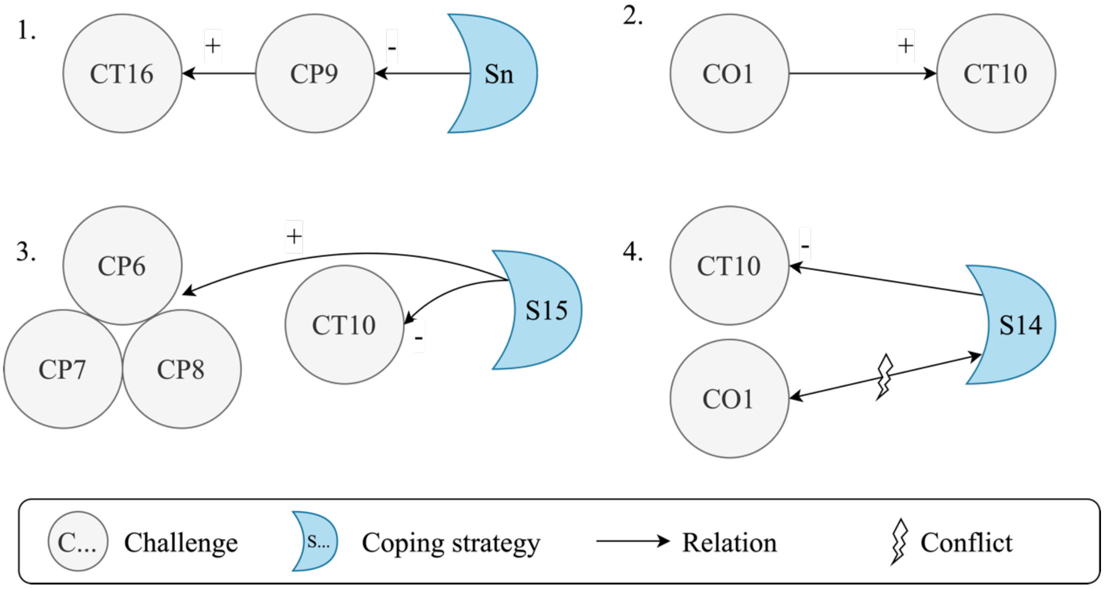


Figure 19 - Overview of interactions between challenges and coping strategies

5 | PATCHING ACTIVITY

"Eventually, security will be almost completely metrics-driven. A reliance on metrics is, after all, the mark of a mature corporate function" – Berinato, Daintry, Scalet, Wailgum, and Wheatley (2004)

Chapter 4 identified the decision-making process IT practitioners go through with security patching, where multiple challenges and socio-technical factors influence aspects of the decisions to be made. The goal of this chapter is to explore the possibilities of getting a deeper understanding of the challenges and effectiveness of the patch process with the use of quantitative data. Where the qualitative data of interviews in Chapter 4 is useful to explore the interplay of factors in decision-making, quantitative data could be used to provide more in-depth insights into the significance of these factors. The possibility of using measurements as a source for decision-making is explored by assessing the state of measurement practices of the case study organisation.

This chapter is structured in the following way. First, section 5.1 discusses the purpose of measurements and metrics analysis regarding security patching. Here the relevant publications of Chapter 3 will be used to understand the usefulness of quantitative data. Additionally, here the objective of the further analysis will be discussed. Second, section 5.2 will demonstrate the quantitative data analysis conducted with the case study organisation to assess the usefulness of having some data. Thirdly, section 5.3 discusses the difficulties in measurement practices and the limitations to come to meaningful measurements. Finally, section 5.4 formulates a conclusion and provides an answer to sub-question 3, discussing the current measurement quality of the case study organization.

5.1 Purpose of measurements and metrics analysis

The Capability Maturity Model Integration (CMMI) "helps organizations streamline process improvement, encouraging a productive, efficient culture that decreases risks in software, product, and service development" (White, 2021, par. 1). This model, therefore, suits the objective of understanding the maturity state of an organisation regarding security patch management. It consists of five maturity levels, from level 1, the initial level, to level 5, the optimizing level (Elina, 2021). The Initial level 1 is where there is no documented and adhered to process. The Managed level 2 is where there are some processes established but not executed in a standard manner (Elina, 2021). The Defined level 3 is where there are defined process inputs and outputs, metrics are established, the level of risk is understood, and processes are audited (Elina, 2021). Going further up the ladder, the Quantitatively Managed level 4 is where decisions are based on quantifiable data and key performance indicators (KPIs) and critical success factors (CFSs) are established (Elina, 2021). The most mature level is 5, Optimizing, here decisions are highly data-driven and automation and integration of non-IT processes is included (Elina, 2021).

The case study organisation is currently at the maturity level of between 1 and 2, as there are some processes established, for example on incidents and changes, but no standardized documentation on patch releasement and deployment. The existing processes are furthermore only loosely managed. The aim of the organisation is to reach at least maturity level 3, and perhaps level 4. The goal of this chapter is to investigate the possibilities for this organisation to move to level 3 regarding its security patch management process and explore what is needed to even move up to level 4. To understand how to go to level 3, where the usage of measurements and metrics play a significant role in decision making, it helps to assess current literature on the usage of metrics in security patch management. Jaquith (2007) in his book 'Security Metrics' argues metrics can help organisations in several ways: To understand security risks, spot emerging problems, understand weaknesses in their security infrastructures, measure the performance of countermeasure processes, and recommend technology and process improvements (p. 40). Several publications included in the literature review in Chapter 3 mention 'metric(s)', 'measure', or 'KPI(s)' (Hoehl, 2013; Mell et al., 2005; Souppaya & Scarfone, 2013; Souppaya & Scarfone, 2022; TBS, 2022). The next section (5.1.1) will discuss what these publications recommend regarding the use of metrics for security patch management. Following, section 5.1.2 will discuss the scope and objective of the analysis in the remaining of this chapter.

5.1.1 Guidance on the use of metrics

Metrics are useful to determine the effectiveness and the efficiency of patch management in an organisation and can reveal the vulnerability condition of the IT environment (Hoehl, 2013; Souppaya & Scarfone, 2013). Examples of such metrics are the percentage of patches deployed within the suggested deployment schedule or the minimum, average, and maximum time of applying patches to X% of hosts (Souppaya & Scarfone, 2013; TBS, 2022) (see Appendix D for a full list of exemplary metrics). Metrics can additionally be used to get insight into risks, for example, on the number of planned patches not in place or the number of vulnerabilities per host (Mell et al., 2005). As this helps to get an understanding of the level of security, it can also be used as a way of complying to organisational policies (Hoehl, 2013). It can help illustrate the adequacy and progress of the security controls and organisation's procedures (Mell et al., 2005; Souppaya & Scarfone, 2013). Examples of these types of metrics are the response time for risk assessment, the number of 'change management' violations, or the coverage of patch management technologies for the organisation's desktops and laptops (Hoehl, 2013; Souppaya & Scarfone, 2013). These metrics allow to demonstrate in what way pre-defined targets are met. Furthermore, the usage of metrics enables the justification of security investments to upper-management. Here cost-metrics could be useful, for example, the costs of tools and services, costs of system administrator support, or costs of program failures (Hoehl, 2013). Reporting on metrics should be done frequently to manifest the safeguarding of credible and sustained patching (Hoehl, 2013). It additionally enables to establish a baseline of performance, according to TBS (2022). Performance targets should be established and communicated to relevant parties (i.e., system-owners and security officers) (Mell et al., 2005). These targets should be realistic, and if achieved, carefully adjusted to avoid the overwhelmingness of IT practitioners (Mell et al., 2005). Table 9 sums up the three main utilities of metrics in the context of security patching.

Table 9 - Utilities of metrics

Utilities of metrics
Way to measure the effectiveness and efficiency of the patch process
Way of compliance to organisational guidelines or agreements
Way of justification of needs for security investments to upper management

In addition to providing examples of why metrics are helpful, the publications also provide some guidance on considerations to take into account when implementing the usage of metrics. Implementing metrics is an iterative process that is dependent on the level of security maturity of organisations (Hoehl, 2013). For example, organisations with a low maturity level should focus on metrics that indicate the susceptibility of attack, organisations with a more mature level should focus more on their response time to vulnerabilities, and those with a very mature level should focus on the optimization of costs (Mell et al., 2005). Metrics should be measured over time to indicate this matureness of security patching. For instance, scans need to be carried out frequently, as only doing this on a monthly basis would result in the number of identified vulnerabilities being inaccurate (Souppaya & Scarfone, 2022). Mell et al. (2005) argue that there could be limitations to the tools used for metric measurements as false positives or false negatives could occur. Host-based scanners and network-based scanners have had the most issues, historically, as these aim to provide an 'alert' but do not indicate an actual vulnerability (Mell et al., 2005). Organisations should take advantage of the information sources already available to them, i.e., the low-hanging fruits of existing inventories of software and hardware (Souppaya & Scarfone, 2022). However, 'overly simplistic' metrics, e.g., only the number of vulnerabilities, will not be actionable. Souppaya and Scarfone (2022, p. 16) illustrate this by an example: "If you were told that 10 % of your assets were not being patched, what does that actually mean in terms of your organization's risk? What is the relative importance of each of those assets?". Adding the relative importance of an asset and the vulnerability can provide a more insightful overview already. This is in line with the challenge of identifying meaningful and actionable metrics explained by Souppaya and Scarfone (2022). This is also because multiple different stakeholders involved with an organisation's procedures and guidelines (e.g., the Board of Directors, the Chief Information Security Officer, the security team, and operational IT practitioners) all prefer a different level of abstraction of metrics.

5.1.2 Meaningfulness of measurements

The previous section concludes that metrics are highly useful for organisations to reach a more mature security patch process. While most of these metrics require organisations to have high-quality measurements, it is stated that the type of metrics should be tailored to the existing maturity level. This allows organisations with a low maturity level to still be able to retrieve valuable information by existing measurements. As identified in Chapter 4, the case study organisation does not make use of KPIs [F24] to keep track of the effectiveness of patching. This sparked the interest in exploring the state of their existing measurement practices and in what way these are able to provide meaningful information. Important to distinguish is the difference between metrics and measurements. Measurements are the source of data to give meaning to a metric. Figure 20 illustrates this process of getting from raw data to developing insights. Where the step of going from data to information is done by measurements, going from information to insights is by using metrics. These insights are used to inform business decisions.

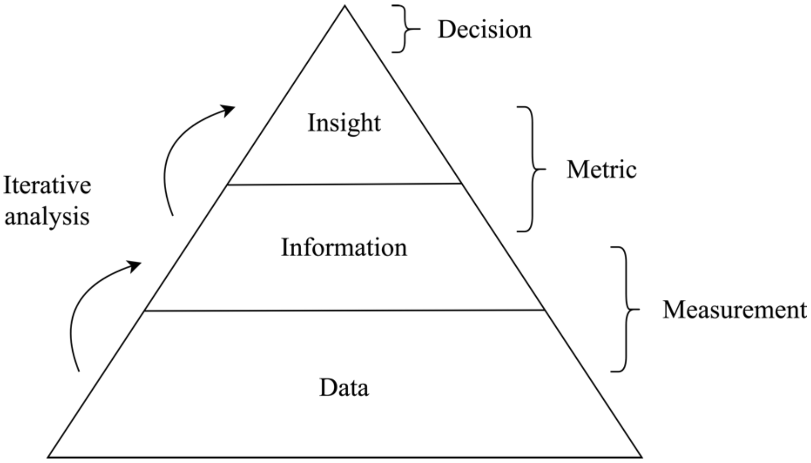


Figure 20 - Data, information, insight (based on Figure 5-1 from (Jaquith, 2007, p. 134))

Raw data used in this research will be collected by collaborating with two teams within the IT department, Team A and Team B. Team A is responsible for the maintenance and operations of applications and services on around 600 Windows servers, whereas Team B is responsible for similar activities of Linux systems. This distinction is made based on the different maturity levels of both teams when it comes to security patching. The discussions with IT practitioners of both teams were of exploratory nature and were focused on getting an understanding of their current way of using measures, the possibilities of using measures, and the potential difficulties they were facing. The topics of discussion thus were centred around three main subjects:

- Exploring measures: *What is currently being measured?*
- Exploring ways of measures: *How is it currently being measured?*
- Exploring the purpose of measures: *How is what is measured used?*

The meetings indicated two states of measurement: having *some* measures and having *no* measures. The following sections will discuss these two states, where section 5.2 will explore the possibilities of having some measures, section 5.3 will explore the difficulties of getting useful measures.

5.2 Exploring the possibilities of having *some* quantitative data measures

This section explores the potential of using quantitative data to provide meaningful information on the aspects of the decision funnel (see section 4.3.1) IT practitioners go through. Discussions with Team A revealed some data is available on patch activity due to tools provided by the software vendor. These tools are used to receive and approve patches from the vendor, and to keep track of the patch status of systems. The following section (5.2.1) will discuss the aim of analysing this data to assess the meaningfulness of the information it provides.

5.2.1 Objective of data analysis

The meaningfulness of data is dependent on what is defined to be 'meaningful'. For the purpose of this study, it is meaningful to understand the decision-making that influences the patch activity and in what context the organisation needs to make decisions. This will allow the case study organisation to understand how the measurements in place are able to increase the knowledge of the functioning of their security patch process. In this way, quantitative measurements might be able to get a better understanding of how socio-technical factors or challenges influence the current patching process. Challenges in Chapter 4 indicated that the occurrence of vulnerabilities is unpredictable (CO1) and information gathering in the case study organization is done in an ad hoc way (CP8), wherefore it is interesting to explore if quantitative data is able to provide a better understanding of the potential discrepancies in the flow of patch releases, patch assessments, and patch deployments. As earlier illustrated in Chapter 3, Figure 21 shows the common phases of the patch process. The aim here is to explore relevant data on phases one, two, and four (indicated in blue).

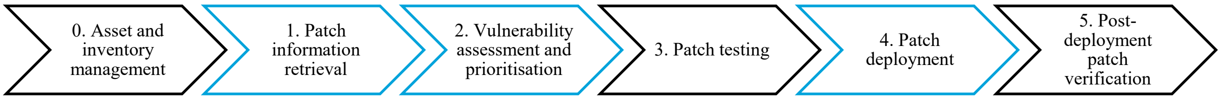

0. Asset and inventory management
1. Patch information retrieval
2. Vulnerability assessment and prioritisation
3. Patch testing
4. Patch deployment
5. Post-deployment patch verification

Figure 21 - Phases of a common patch process

Hypothesising this flow of patch releases, patch assessments, and patch deployments leads to Figure 22, which illustrates the different sizes of the number of patches in each phase. Information on patches is released by software vendors daily; however, not all patches of this set are received by the IT department. As seen in section 4.2 of Chapter 4, there are multiple information sources to become aware of the existence of a patch. Of the patches that the IT department is aware of, a part of these might not be relevant from a security or applicability aspect. This results in two groups of patches, those the IT department is aware of but which are not explicitly evaluated due to being not relevant, and those the IT department is aware of which are explicitly evaluated due to being relevant enough. The latter is then evaluated on whether to patch and, if so, when and how. Here, some patches are assessed to be not relevant after all and will not be deployed.

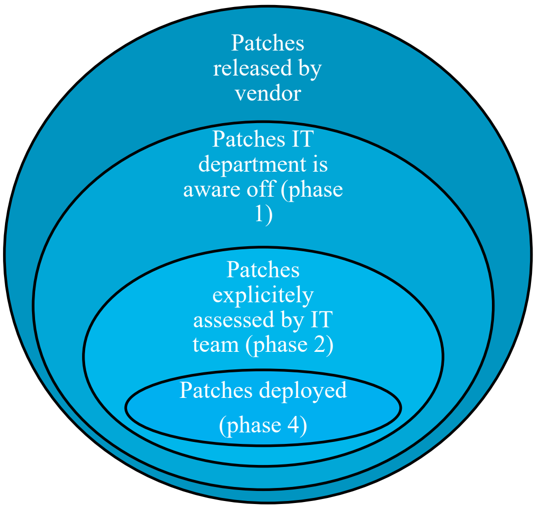

Patches released by vendor
Patches IT department is aware off (phase 1)
Patches explicitely assessed by IT team (phase 2)
Patches deployed (phase 4)

Figure 22 - Hypothesis of patch flow

As discussed in section 5.1, the relevance of the type of measure is dependent on the level of security maturity of organisations (Hoehl, 2013). As the case study organization has a low security maturity level, the focus lies on more simple measures that could indicate the susceptibility of attack and the response time to vulnerabilities. The measurements that will be explored in this case study are focused on volume, frequency, and where data availability allows, timing (see Table 10). Due to the availability of information, the scope of data collection ranges from July 2021 until July 2022 for section 5.2.2 and section 5.2.3 and ranges from the 20th of June until the 20th of July for section 5.2.4. Several data sources are used, including management summaries of Microsoft's tools, Windows Server Update Services (WSUS) update scripts, Microsoft Security Update Release Notifications, and a connections manager used that allows an overview of the different servers.

Table 10 - Quantitative measures analysed

Section	Steps in funnel		Type	Measurement	Unit
5.2.2	**Patches released by vendors**		Volume	Number of security patches released by vendor per criticality score (low, moderate-important, critical)	[#patches]
			Frequency	Number of security patches per month	[#patches / month]
5.2.3	**Patches the IT department is aware of**	**Patches explicitly evaluated**	Volume	Number of (security) patches evaluated	[#patches]
			Frequency	Number of (security) patches per month	[#patches / month]
		Patches not explicitly evaluated	Volume	Number of (security) patches not evaluated	[#patches]
			Frequency	Number of (security) patches not evaluated per month	[#patches / month]
	Patches that are deployed by the IT department		Volume	Number of (security) patches approved	[#patches]
			Frequency	Number of (security) patches approved per month	[#patches / month]
5.2.4			Timing	Throughput time of patches (patch delay) over a month	[# unpatched systems / day]
			Coverage	Percentage of systems with planned patches not patched yet	[% #unpatched systems / #total systems]

5.2.2 Setting the scope: Vulnerability and patch release by vendor Microsoft

The number of vulnerabilities and released patches of all Microsoft products is publicly accessible[7]. Here, per month can be determined how many vulnerabilities were identified and how many security patches were deployed to fix these vulnerabilities (see Figure 23). These include three different kinds of patches: Security Updates, Monthly Rollups, and Security Only. Where Security Only and Security Updates contain patches for solely new vulnerabilities, the Monthly Rollups include all security patches from the previous months as well. This strategy of Microsoft helps customers to manage their security patch management, as it takes away the hassle of having to rely on previous versions of software to install certain patches. With Monthly Rollups, these prior patches are included in the same package as the new security patches, reducing the challenge of system dependencies (CT12). Figure 23 illustrates that the number of patches released is greater than the number of vulnerabilities present for each month (except July/22). This indicates that the relationship between vulnerabilities and patches is not 1-on-1, as more patches are needed to fix a lower number of vulnerabilities. Here the number of patch releases differs strongly per month, indicating the unpredictability factor (CO1) and the factor of a large magnitude of patch releases (CT13) identified as challenges in Chapter 4. Furthermore, the division of low impact, moderate to important impact, and critical impact of patches differs greatly per month. Comparing January 2022 with May 2022, in both months there were roughly 250 patches released. However, in January none of these had a low impact and 18.8% of these patches were critical, where in May 41.7% of patches had a low impact and 0.4% of patches were critical. The nature of a patch therefore differs throughout time. This does not provide any *new knowledge* as these numbers are published by Microsoft already, but it does provide a good understanding of the context of security patching.

[7] https://msrc.microsoft.com/update-guide/deployments & https://msrc.microsoft.com/update-guide/vulnerability

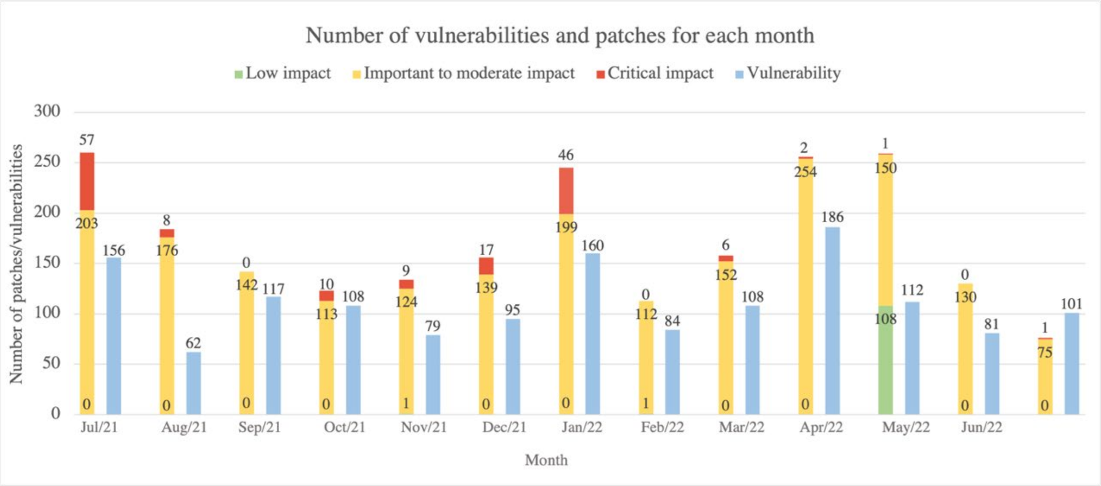


Figure 23 - Number of vulnerabilities and patches per month for all Microsoft products

5.2.3 Analysing assessment behaviour: Patches received and approved by the IT team

Comparing this public data of the patches released by Microsoft and the number of patches received and approved by the IT team, several observations can be made (see Figure 24). First, the number of patches released is for all months much greater than the number of patches received by the IT department. Furthermore, for the majority of months (7/12), the number of security patches released is greater than the number of patches received by the IT department. P5 explains the coping strategy of patch filtering on relevance (S16) to reduce the number of patches to be assessed by the IT department. Not all Microsoft products are used in the organisation, wherefore some patches are not applicable. The difference between the number of patches released and the number of patches received could therefore be explained by the applicability aspect and the operational aspect of the decision funnel (see section 4.3.1). Using this coping strategy reduces some operational challenges, for example the need to balance time between patching and tasks of day-to-day job (CO3).

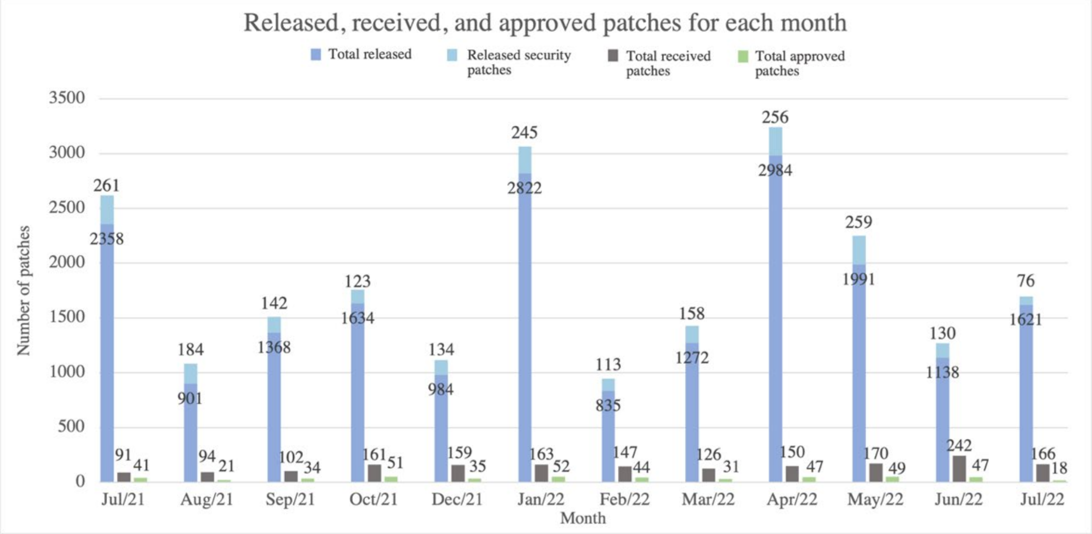


Figure 24 - Patches released, received, and approved per month

Secondly, the number of patches received is greater than the number of patches approved for each month. The difference between the number of patches released and the number of patches received could therefore similarly be explained by the applicability aspect of the decision funnel (see section 4.3.1). P5 argues that although the first filtering step (between released and received patches) is beneficial, there are always patches that are still not relevant for the specific IT environment of the organisation, for example when a certain functionality is not used.

5.2.4 Analysing patching behaviour: Patch status for Windows servers over time

Microsoft releases its routine patches in a bundle on the second Tuesday of each month. When exploring the status of servers before and after Patch Tuesday of July 2022, a few things can be observed (see Figure 25):

- In the days leading up to Patch Tuesday (until the 5th of July), the majority of servers are patched and the number of servers in need of an update slowly decreases.
- On the days closest to Patch Tuesday (from 6th to 8th of July), the number of servers in need of an update increases, slowly.
- The day after Patch Tuesday (13th of July), the distribution between patched and unpatched systems is mirrored in comparison with the days before, with the majority of systems unpatched.
- On the second day after Patch Tuesday (14th of July), the distribution between patched and unpatched systems is roughly equal, with around half of the servers patched and half of the servers unpatched.
- On the first and second day after Patch Tuesday (13th and 14th of July), the number of servers with install errors and the number of servers needing a reboot is highest compared to all other days.
- In the days that follow (from the 15th of July and onwards), the distribution is similar to that of around the same date in the month before (20th of June).

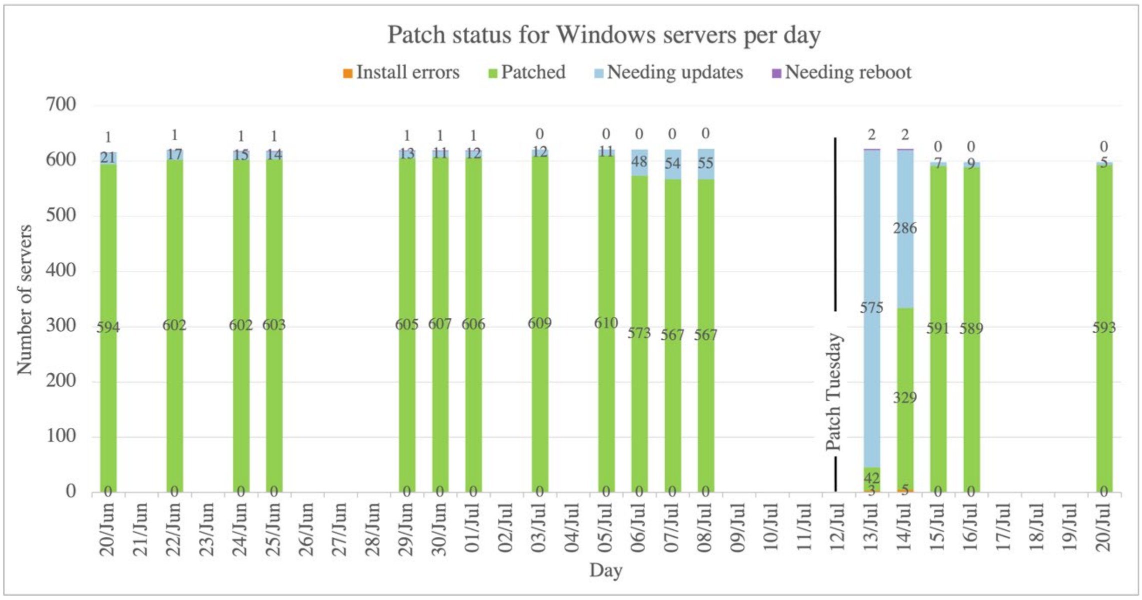


Figure 25 - Patch status for Windows servers over time

These observations give more meaning when compared to the aspects of decision-making and the socio-technical challenges identified in Chapter 4. The aspects of the decision funnel most present in this dataset are operability and availability. Even in the last days of the month of June, there are still servers in need of an update. P5 explains that these are the faculty-managed servers, those where the system-owners are responsible for patching. This data thus supports the existence of the challenges of the behaviour and awareness of system-owners (CO2), collaboration (CP6), coordination (CP8), and procedures and guidelines (CP9). The data demonstrates that the availability aspect weights heavier than the security aspect for some system-owners (and with that their systems), resulting in the lack of patching certain servers. This indicates the lack of agreements on ownership responsibilities discussed in section 4.2.3.1, where the distributed decision rights allow for this situation to occur.

Assessing the decision-making of IT practitioners in Team A, a first hint of the operability aspect is represented in the number of unpatched servers on the days closest to Patch Tuesday (from 6th to 8th of July). According to P5, Microsoft often releases certain patches for MS Office several days before Patch Tuesday. These patches are saved up until after Patch Tuesday to keep it more manageable from an operational perspective. What indicates this even better is the observation of the second day after Patch Tuesday (14th of

July), where the distribution between patched and unpatched systems is roughly equal, with around half of the servers patched and half of the servers unpatched. This illustrates how is dealt with the challenge of limited resources (CT13) and the availability coping strategy of always having some systems available for end-users (S9). Once again here, the quantitative data provides better insight into the significance of earlier identified challenges.

The coping strategy of waiting for the confirmation of other organisations (S14) and testing in a test environment (S13) to deal with the challenge of patch quality (CT10) and the possibility of side effects is represented in the numbers of 13th of July, the day after Patch Tuesday. Instead of deploying all approved patches directly, the IT practitioners first wait a day, in which they can also test patches in the test environment, as indicated by the high number of unpatched systems and the number of servers with install errors non-zero. Here, the quantitative data provides insight into the application of coping strategies.

5.2.5 Limitations of measurements

The data collected provides insights into the monthly cycle of Microsoft patches and how the aspects of applicability, operability, and availability are represented in the patching activity of IT practitioners. Such data and analysis can be useful for IT practitioners to get an initial overview of how patching is conducted and where improvements can be made (e.g., faculty-managed servers). There are, however, several limitations to the usefulness of this data analysis and the current measurements in place.

- *The current measurements are highly dependent on the tools made available by the software vendor*

A practical example of this limitation is the data analysis in section 5.2.3 where the initial aim was to compare security patches only. Throughout following iterations of gathering data, it turned out this was not possible due to the difficulty of retrieving that information in the used tool as it was not able to provide the distinction of what patches had which level of criticality. The examples of this section (5.2) demonstrate that even when some data is available, the usefulness is still limited by not having precise enough data.

- *The current measurements do not allow the distinction between routine patches and emergency patches*

As identified in section 4.2, there are two main strategies; emergency patching and routine patching. A limitation of these measurements is that it only covers routine patches as data is collected monthly and daily. When it comes to emergency patches, hourly data is able to provide more accurate insights. Of the number of patches approved and deployed, the current way of measurement cannot determine which of these were critical. The meaningfulness of data regarding the emergency patching process is thus limited.

- *The current measurements do not provide information on the relative importance of the security aspect*

Thirdly, no conclusions can be made about the security aspect, as no data on for example criticality of vulnerabilities and patches is included. For the results of section 5.2.2 and section 5.2.3, only the total number of patches is included, where no differentiation is made between the levels of criticality. This is due to the strategy of Team A to apply all patches that involve a certain level of security fix. This division of patches released, received, and approved is therefore not relevant to determine the security state of the organisation. The number of unpatched servers in section 5.2.4 does furthermore not indicate whether these contain critical patches. As the earlier discussed example of Souppaya and Scarfone (2022) in section 5.1.1, these types of data sources do not consider the relative importance of the systems in the analysis. For example, when the results of section 5.2.4 indicate that 13 servers still need patches, it does not inform whether these are critical patches, nor does it include the number of patches. Information on the security risks is therefore not provided by the current measurements.

- *The current measurements do not allow consistent measures over time due to servers' lifecycles*

Lastly, the number of servers in section 5.2.4 is not constant over time. This is caused by the lifecycle of servers and the daily addition and removal of servers in the IT environment. These are servers replaced by newer ones, or those that are only used temporarily for certain research calculations. For the current maturity level of the organisation, this does not play a significant role of importance. However, when a higher maturity level is reached (level 4), the process needs measurements that allow the evaluation of the effectiveness or efficiency over time.

5.3 Exploring the limitations to come to meaningful measurements

Where Team A had some measures available provided by the software vendor, Team B did not have any measurements; or at least no measurements that are able to provide meaningful information. Investigations with Team B of the IT department show that the usage and collection of quantitative data is not always that effortless, resulting in having no useful quantitative data measures related to security patching. This section aims to better understand what makes this difficult with the help of The Johari Window. Chua et al. (2019) use The Johari Window (see Figure 26) to discover the emergence of unintended consequences of cybersecurity measures.

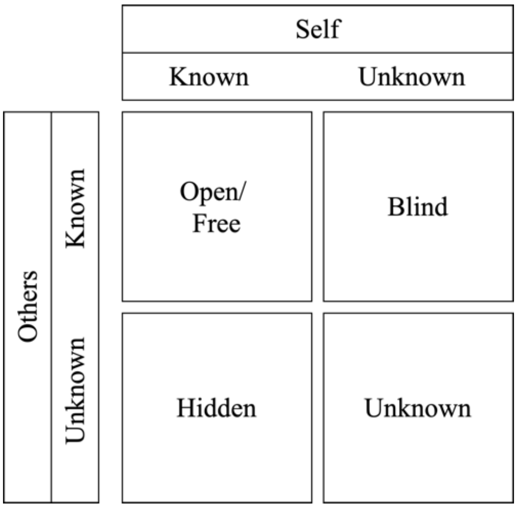


Figure 26 - Johari Window (adopted from (Chua et al., 2019, p. 2))

This window is developed by psychologists Joseph Luft and Harrington Ingham (Massie & Morris, 2011) to help people better understand one's relationship with themselves and others. In the research by Chua et al., it serves as "a means to consider the limitations to knowledge of risks (and in turn, countermeasures) between one entity and others in the ecosystem" (2019, p. 2). As the authors argue, actions taken based on incomplete knowledge could be better informed by gathering information from other stakeholders in the environment. Discussions on the availability of measures with the case study organization indicate the presence of discrepancies throughout the different stakeholder groups in the IT department. The Johari Window is in this study a means to consider the limitation of knowledge on measurement practices on the meaningfulness of decision making. The *Open area* consists of information known to the self and known to others. Reasoning from a decision-making point of view, this is the optimal area as information is known to all stakeholders involved. This leads to the best possible informed decision-making process. The following sections discuss what currently hinders this, summarized in Table 11.

5.3.1 The Blind area

The *Blind area* consists of information known to others, but unknown to the self. Observations with the IT practitioners of Team B indicate that there is no objective truth present in the statistics indicated by their log systems. It is difficult to retrieve information on patches as many systems are uniquely designed, and technical dependencies are present (CT12, CT14). The systems in place to collect patch information need to be able to indicate the technical dependencies between systems. If this is not the case, the numbers in place are not an accurate representation of the actual state of patches, and with that, the level of security. For example, some log systems indicate the number of patches deployed and the patch status of systems. Currently, for data on the number of deployed patches, each individual server can be consulted to give an overview of the number of patches on that particular server. However, here it is not indicated whether they were security patches or general patches. In this way, evaluation of patches by their level of criticality becomes impossible, making it unknown if critical patches are installed in a timely manner. The information is thus unknown to IT practitioners, however, known to IT systems in some form.

5.3.2 The Hidden area

The *Hidden area* consists of information known to the self, but unknown to others. Here there is information asymmetry between operational teams and top-level management, and between operational teams and other operational teams. Data on patching activity is scattered throughout the different teams in the IT department. Where each individual team quite clearly knows what is being measured, and perhaps more importantly, what is not measured, other operational teams and top-level management are not aware of this. This results in different levels of knowledge on what aspects of the patching activity are registered, and in what way. This amplifies the coordination issue (CP7, CP8) identified in section 4.2, as there is a discrepancy between what coordinators and higher-level IT practitioners think is available and what operational IT practitioners know is available. Furthermore, there is information asymmetry between IT practitioners and system-owners. Where measurements (see section 5.2) provide knowledge on the patch status of systems to IT practitioners and with that information on the potential risks the entire organisation faces, system-owners are unaware of these risks.

5.3.3 The Unknown area

The *Unknown area* consists of information not known to the self and not known to others, thus not known to anyone in the organisation. As no organisational goal is determined, security patching is currently no more than a shot in the dark hoping to fix just those vulnerabilities that will not be exploited by threat actors in time. Furthermore, say a breach takes place but no organisational objective determined the level of patching needed, then it can be a tough conversation to determine to what extent IT practitioners are really to be accounted for. Often ‘patch everything as fast as possible’ is the main aim to deal with the challenging developments of the threat environment where the timeframe between vulnerability awareness and exploitation decreases rapidly. As P3 states: *“The goal is just to be up to date, but you are only up to date on the day you do it”*. However, the organisational, procedural, and technical challenges identified in Chapter 4 illustrate that this is an unrealistic aim as the large magnitude of patch releases outplays the capacity and capability of IT resources. One can argue that aiming to patch 100% of system at all times does not contribute to the stimulant of effective patching, as it is known that that percentage is not reached anyway. Additionally, the lack of clear guidance and commands from upper management and the non-existence of company culture create an unknown unknown, as it is not known to any stakeholder what is expected as effective patching. Lastly, a lack of information on assets and inventories causes limitations to the level of knowledge of what systems a vulnerability is present (CT16).

Table 11 - Summary of limiting elements of measurement

Blind area	Hidden area	Unknown area
• Lack of information on IT environment for IT practitioners	• Information asymmetries between operational teams and top-level management • Information asymmetries between operational teams and other operational teams • Information asymmetries between operational teams and system-owners	• Lack of information on what effective security is • Lack of information on assets and inventories

5.4 Conclusion and discussion of current measurement quality

Sub-question 3: What is the quality of organisations' logs to determine patch activities?

The goal of this chapter was to investigate the possibilities for this organisation to move to level 3 and explore what is needed to even move up to level 4. Findings of section 5.1 show the usage of metrics is highly recommended for a multiplicity of utilities and is suitable for measuring the effectiveness or efficiency of the patch process.

Results of section 5.2 show that having some measurements available can help to understand the significance of socio-technical factors and challenges identified in the qualitative data of Chapter 4. Firstly, the results provide insight into the context of security patching, e.g., the scope of patch releases. Current measurements are helpful to better understand certain coping strategies applied to deal with this context, e.g., patch filtering. Even more useful, data can provide more insight into the challenges in the decision-making process related to the applicability, operability, and availability. Quantitative data is useful to verify the existence of challenges, while simultaneously, challenges are a way to explain patterns in this data to explain the ineffectiveness or inefficiency. Important to note is that quantitative data cannot be used to form causal relations between socio-technical factors and metrics. Although there might be a relationship between a certain socio-technical factor and a metric, it cannot be said with certainty that a one-on-one relationship is existent. For example, the presence of unpatched systems throughout a month can be caused by the behaviour of system-owners and ineffective collaboration between them and the IT department; however this can only be assumed. Furthermore, quantitative metrics are limited in the way that they cannot address all socio-technical factors and challenges. It can be concluded that the quality is good to some extent, as some insights can be gathered on patching activity and the challenges that influence this. These findings thus allow the case study organisation to understand how the measurements in place are able to increase the knowledge on the functioning of their security patch process, which is a good start at the current stage of having maturity level 1. This allows to understand where improvements need to be made in order to reach level 3.

However, certainly steps need to be taken to improve the current state of measurements to reach maturity level 3 or level 4. The identified limitations in section 5.2.5 and the discrepancies and absence of needed information identified in section 5.3 make the current measurement quality insufficient. The results of section 5.3 indicate that there is hidden information resulting in information asymmetries between different stakeholders within the organisation, there is blinded information not being able to be retrieved by IT practitioners, and there is unknown information due to unawareness of what effective patching entails and lack of information on assets and inventories. Mapping the limitations to data quality dimensions helps to understand where improvements need to be made. The Universities and Colleges Information Systems Association (UCISA) developed a toolkit for (educational) organisations that provides advice and instructions on key aspects of information security management systems (ISMS) (UCISA, n.d.). One of the chapters of this toolkit advises organisations to assess their existing or proposed measurement against eight measurement quality dimensions. The choice for using this source and these dimensions is based on the scope and level of abstraction of the assessment purpose of this research. The questions help to determine the quality of measurements and assess whether these are informative (see Appendix E for the full list). Table 12 provides a summary of how the current state of the quality of information limits the possibilities for measuring the effectiveness of security patching.

Table 12 - Evaluating the quality of measurement

Metric	Explanation
Reliable?	No objective truth is present in the way patch activity is currently gathered and stored
Sustainable?	Due to the high level of technical dependencies between systems collecting measurements is cost-intensive and time-intensive
Measurable?	Certain measurements can be identified that seem relevant to measure the effectiveness and/or efficiency of patching, however, are limited in their way of providing meaningful information
Objective?	No objective truth is present in the way patch activity is currently gathered and stored
Scoped?	Due to the high level of technical dependencies between systems collecting measurements, interpretation is challenging and context (i.e., system) specific
Instrumentable?	The environments of different IT teams are designed differently resulting in limited scalability and portability
Transparent?	There is a gap between a potential specified state (“should be”) and the real operational state (“as is”)
Progressive?	No information assurance due to lack of targets, lack of ability to compare to previous measurements, and lack of linkage to specific business goal

6 | SYNTHESIS AND RECOMMENDATIONS

"60% of data breaches are caused by a failure to patch. If you correct that, you've eliminated 60% of breaches. And I didn't even have to say AI or Blockchain! See how that works?"
– Ricardo Lafosse, CISO (SecureWorld, 2019)

The goal of this chapter is to synthesize and reflect on the qualitative results of Chapter 4 and the quantitative results of Chapter 5 to determine what this entails. This is step 4 of the convergent design of the mixed methods approach discussed in section 2.1.1. This aims to provide a better understanding of the interplay of socio-technical challenges and the usage of quantitative measurements. It further aims to explore what the findings reveal about the case study organisation, to recommend how improvements could be made to make their security patch process more effective.

The synthesis in section 6.1 will connect the different elements of this research by discussing several concluding statements, answering sub-question 4. Section 6.2 aims to reflect on this synthesis to indicate the significance of the findings and whether a distinction could be made on the importance of the different challenges and socio-technical factors. This will lead to practical recommendations for the case study organization in section 6.3.

6.1 Synthesis of qualitative and quantitative findings

Synthesising both qualitative findings of Chapter 4 and quantitative findings of Chapter 5 helps to achieve the benefits of mixed methods research. The remaining of this section will discuss several concluding statements that synthesize the results and demonstrates how qualitative and quantitative components relate, to provide an answer to sub-question 4.

Sub-question 4: How does measurement of patching activity relate to the socio-technical factors causing the tensions and dependencies found in the patching decisions?

- *CH4: Socio-technical factors influence the effectiveness of organisational security patching*

The qualitative results of Chapter 4 help to get a broader understanding of organisational security patching. The observed socio-technical factors that form organisational, procedural, and technical challenges are causing the phases of the security patch process to be less effective than potentially possible. During the decision-making process IT practitioners go through to determine what and when to patch, socio-technical related to the aspects of security, applicability, operability, and availability influence the outcome of the decision. A socio-technical factor is either a cause that shapes the decision space, a barrier that hinders the decision space, or a constraint that limits the decision space.

- *CH4: The interplay between socio-technical factors and challenges results in different ways of influence*

The different factors do not exist in a vacuum as these interact and influence one another. Section 4.4 demonstrates this interplay in more detail, where multiple types of interactions are present. The existence of one challenge can cause the occurrence of another challenge. Furthermore, measures and coping strategies to target challenges are able to reduce the significance of challenges that socio-technical factors bring. A coping strategy to target one challenge could, however, lead to a conflict with another challenge or increase the need to target another challenge.

- *CH5: Quantitative data can, to some extent, provide insight into the significance of challenges and socio-technical factors and, with that, the effectiveness of the patch process*

As the results in section 5.2 show, quantitative data is able to provide a deeper understanding of socio-technical factors and the significance of these to some extent. Quantitative data is useful to verify the existence of socio-technical factors and challenges, while simultaneously, challenges are a way to explain patterns in this data to explain the ineffectiveness or inefficiency. However, quantitative data is limited in the way that it cannot explain causal relationships nor is able to address all identified socio-technical factors. The quantitative results furthermore help to understand the relation between the quality of quantitative data and the socio-technical challenges.

- *CH5: The presence of socio-technical factors limits the quality of quantitative data and, with that, the meaningfulness of current measurements*

The usefulness of measurements is highly dependent on the maturity level of the organisation. Getting high-quality, meaningful data turns out to be a challenge for the organisation due to the presence of socio-technical factors. Apart from the existing limitations of present data, there are limitations that make using quantitative data troublesome. Lack of information and information asymmetries between different teams in the organisation contribute to the difficulty, limiting the knowledge of measurement practices and, with that, the meaningfulness of decision making.

- *CH6: High-quality quantitative data can provide help in decision-making of organisational measures*

Having insight into both the effectiveness of the patch process and the challenges and socio-technical factors could help IT practitioners in the formulation or compliance of organisational policies and procedures (as indicated by literature in section 5.1). This linkage between challenges and socio-technical factors in Chapter 4 and the possibilities of quantitative data in Chapter 5 can help to ease decision-making during the security patch process if pre-established choices are made for certain circumstances. Figure 27 visualises these statements between the different components of this research.

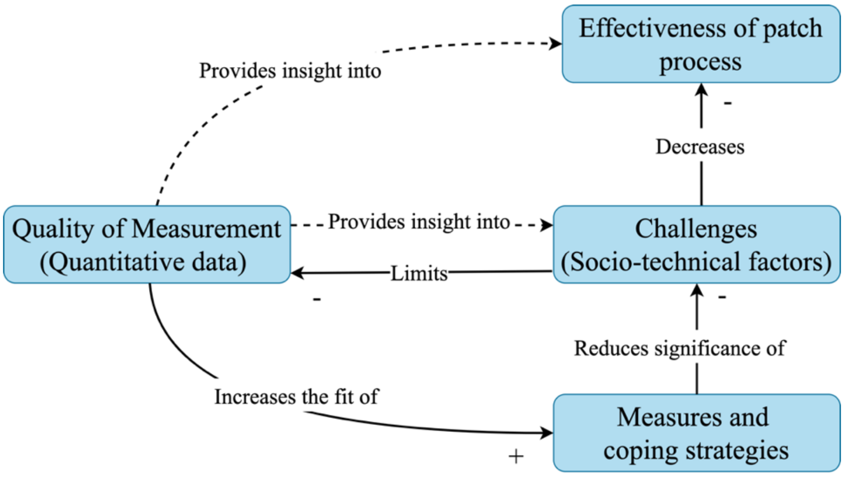

Effectiveness of patch process
Provides insight into
-
Decreases
Quality of Measurement (Quantitative data)
Provides insight into
Challenges (Socio-technical factors)
Limits
-
-
Reduces significance of
Increases the fit of
Measures and coping strategies
+

Figure 27 - Synthesis of qualitative and quantitative findings

The convergence and interrelation of qualitative and quantitative components indicate a vicious cycle. Socio-technical challenges impact the effectiveness of patching, e.g., the higher the significance of challenges, the more ineffective the patch process. Challenges can, fortunately, be reduced by having the right measures and coping strategies in place, e.g., policies and procedures. The fit of these measures and coping strategies can be increased by insights measurements provide on the effectiveness of the patch process and the socio-technical challenges. The better the quality of measurements and quantitative data, the better the fit of measures. However, the quality of quantitative data is limited by the existence and significance of socio-technical factors. This indicates a reinforcing feedback loop between challenges, measures, and the quality of quantitative data.

6.2 Reflection of findings synthesis

This section aims to reflect on the synthesis of results. Figure 27 indicates that if organisations have multiple challenges to tackle and at the same time have a low quality of quantitative data, it would be nearly impossible to get out of this vicious cycle, causing the practice of security patching to remain ineffective. Fortunately, there is some nuance to be added to this for a multiplicity of reasons. This section reflects on the nature of the challenges and forms a bridge to the recommendations in section 6.3.

- *CH6: Not all factors have the same level of significance*

In previous chapters, all challenges are assessed as being equally important and therefore equally contributing to the problem of ineffective patching. Lumping all socio-technical factors together would suggest that targeting a random one would have an equal impact on the effectiveness. This is not the case; moreover, there is a significant distinction to be seen when relating the different challenges and their interactions together. Appendix C discusses the significance of each of the identified challenges in detail. The following Figure 28 presents an indication of the significance of each challenge on the decision to patch and when. The arrows on the left indicate the underlying relationships between different challenges.

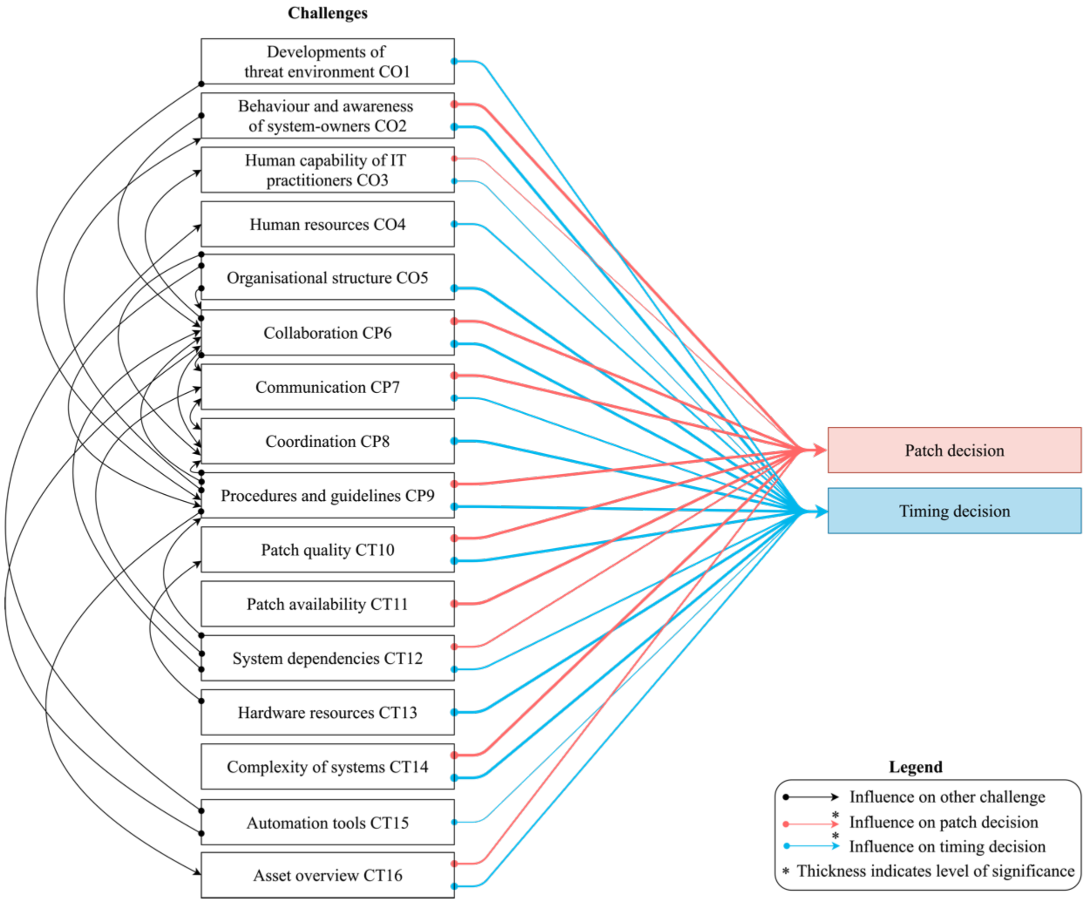

Challenges
Developments of threat environment CO1
Behaviour and awareness of system-owners CO2
Human capability of IT practitioners CO3
Human resources CO4
Organisational structure CO5
Collaboration CP6
Communication CP7
Coordination CP8
Procedures and guidelines CP9
Patch quality CT10
Patch availability CT11
System dependencies CT12
Hardware resources CT13
Complexity of systems CT14
Automation tools CT15
Asset overview CT16
Patch decision
Timing decision
Legend
Influence on other challenge
Influence on patch decision *
Influence on timing decision *
* Thickness indicates level of significance

Figure 28 - Interrelations between challenges and corresponding influence on decisions

This indicates that the majority of these socio-technical factors influence the decision on the timeliness of the patch moment. Additionally, this indicates that the challenge of collaboration (CP6), coordination (CP8), and procedures and guidelines (CP9) are most significant. This is based on the assumption that the number of interrelations of a challenge with other challenges could indicate the significance of the challenge itself. This is not yet based on quantitative data or calculations, therefore there lies an opportunity to quantify the relationships based on their numbers for further research, as will be discussed in section 7.2. Although this might give a rough understanding of the significance of challenges, which could be helpful for organisations wanting to implement certain measures, exemptions are certainly at order. For example, the technical challenge of the complexity of systems (CT14) does not have any interrelations with other challenges, but nevertheless has a high level of significance for the patch and timing decision (see Appendix C). Despite the exemptions, would it make sense for organisations to target the challenges with a high density of interrelations first? Yes, it could be argued that this is a good starting point. However, based on the socio-technical complexity of the problem, the following statement needs to be considered.

- *CH6: Not all challenges are to be solved as easily as others*

Where certain challenges could be fixed relatively easily, others certainly require a higher level of effort. There are roughly three levels of the ease of fixing: highly easy, moderately to not easy, and to a certain extent, but not in its entirety. This categorisation is based on the hypothesised interpretation criteria of the researcher. Challenges that could be fixed '*highly easy*' are those where implementations from within the IT department could already have a high level of influence. Examples of these are internal communication (CP7) and coordination (CP8), organisational procedures and guidelines (CP9), and the challenge of the usage of internal automation tools (CT15). Challenges that fall into the category of being fixed '*moderately easy*' to '*not easy*' are those where a dependency on other stakeholders and their behaviour remains to be present. Examples of these are the behaviour and awareness of system-owners (CO2), patch quality (CT10) and availability (CT11), and the challenges the organisational structure (CO5) brings. Lastly, there are challenges that could only be fixed '*to a certain extent, but not in its entirety*', where the nature of the challenge does not allow it to be fixed. Examples of challenges in this category are the developments of the threat environment (CO1) or the complexity of systems (CT14) like legacy systems. The overview in Table 13 combines both the level of significance and the solvability of the challenges.

Table 13 - Solvability and significance of identified challenges

Solvability	Challenge	Level of Significance
High	CO3 Role and capability of IT practitioners	Low
	CO4 Human resources	Moderate
	CP7 Communication	High
	CP8 Coordination	High
	CP9 Procedures and guidelines	High
	CT15 Automation tools	Low
Moderate to Low	CO2 Behaviour and awareness of system-owners	High
	CO5 Organisational structure	High
	CP6 Collaboration	High
	CT10 Patch quality	High
	CT11 Patch availability	High
	CT13 Hardware resources	High
	CT16 Asset overview	Moderate
Low to impossible	CO1 Developments of threat environment	Moderate
	CT12 System dependencies	Moderate
	CT14 Complexity of systems	High

6.3 Practical recommendations for case study organisation

Based on the synthesis and reflection of the previous two sections of this chapter, recommendations can be formulated for the case study organisation. Under the assumption that it would be most feasible and beneficial to target those challenges that both have a high solvability and a high level of significance, three main challenges meet this criterium: Communication (CP3), Coordination (CP8), and Procedures and Guidelines (CP9). Targeting these challenges first will positively impact the effectiveness of the current security patching process, and as concluded from the interrelations of challenges in section 4.4, these will likely positively influence other challenges simultaneously. As these challenges are intertwined, there is a bundle of recommendations to target these all together. Starting with the essential one, determining an objective.

- *Fix the gap between wanting to do 'better', and knowing what 'better' entails*

A security patch process and the measures to deal with the faced challenges can only really be effective when it is known what effective means. While this is not a new recommendation (see section 5.1.1), it does reveal that security patching is still practised in an ad hoc way, without a certain goal. While this is mainly a governance problem, behaviour and awareness are certainly at the centre of attention. Even if IT practitioners' intentions and efforts are genuine, the lack of a clear organisation culture that prioritizes patching contributes to ineffective patching. Addressing this coordination issue between top-level management and operational IT practitioners should involve the documentation of clear and suitable targets. As this research demonstrated, the current measurement practices are insufficient to be meaningful. Determining what effective patching entails should be established with due consideration of the identified socio-technical factors. This should be based on the maturity level of different teams. Although organisation-wide targets help to increase the security patching culture, the determination of a target should be tailored to individual teams or groups of teams. This includes the integration of the complexity of interdependencies with different external stakeholders. The dependency of the software vendors' release cycle influences the current way security patching is organised. The dependency on external system-owners brings difficulties that are only relevant for certain teams. Furthermore, designing targets should be tuned to the measurement possibilities and challenges of different teams. Where some teams benefit from software vendors' tools and support, other teams are self-assigned to keep track of patching activity.

- *Formalize agreements on ownership accountability and responsibility to deal with the decentralised nature of the organisation*

Where any organisation deals with dependencies on the behaviour of end-users, in decentralised organisations where certain systems are owned by other parties, this dependency is more significant. Especially when patching rights are endowed to other parties, negligence of formal agreements of patch activity causes great security risks. The tension between security and availability is more steered towards availability as, in general, external parties are less concerned with security. A way to limit this dependency is to withdraw patching rights to the central IT department. This however amplifies other challenges related to the operational aspect of security patching, e.g., resource availability and communication. Another way is to formalise agreements on ownership accountability, and responsibility. These agreements could entail pre-defined targets and the consequences of not reaching these. Consequences could entail the termination of usage of the organisation's network through isolation of the system until patched. Furthermore, formalising the security risks that come with ineffective patching could bring an external stimulant to increase the feeling of responsibility. Apart from external agreements on accountability and responsibility, internal agreements within the organisation are needed to decrease the significance of socio-technical challenges. The current practice is based on informal interactions throughout the process. Determination of formal roles, responsibilities, and authority of individuals and entire teams associated with each activity in the process to regulate who has the decision-power in what circumstances, will improve the effectiveness of decision-making. Key here is universal clarity of and adherence to these agreements. This will additionally ease communication by increasing the availability of information and level of knowledge on who to involve in what stage of the process.

- *Define processes and strategies for different circumstances based on the decision funnel, including the aspects of security, applicability, operability, and availability*

As the cliché entails 'a goal without a plan is just a wish', internal procedures and policies need to be documented to determine what actions to take throughout the patch process. Specifically important here is to determine when the state of emergency is met, as currently this is not established, resulting in a higher significance of challenges. The different aspects of security, applicability, operability, and availability involved in decision-making should be acknowledged and explicitly considered to determine what actions are needed. Here, choices need to be pre-defined of when one aspect is prioritized over another. For example, in a state of emergency, security should be prioritized over availability, allowing IT practitioners to take quick action. This would not only make security patching more effective, but it will also allow for more consistency in dealing with patches.

- *Actively explore and document current limitations and organisation-wide opportunities to use measurements to keep track of the effectiveness*

The findings of Chapter 5 indicated the current state of measurements in the organisation and indicated changes need to be made in order to reach a higher maturity level of security patching. It is therefore recommended that for each team, it is clearly mapped what current measurements exist, what the faced problems are, and what data they think is useful to improve their patch process. Furthermore, it is highly recommended to map the existing assets and inventories in the IT environment to understand where potential data sources lay. Here it is important to collectively determine the relative importance of systems and the services they provide. This helps ease decision-making when prioritization is needed. It is recommended to investigate the possibilities of tool usage that can identify technical dependencies between systems.

7 | DISCUSSION

"Patching keeps being a difficult subject, often because theory does not match with what we experience in practice" – Interviewed IT practitioner

Several aspects reported in this research may be seen as 'common sense' to IT practitioners or researchers. However, the findings certainly contribute to a better understanding of security patch management from a socio-technical perspective. This chapter aims to reflect on the relevance of this research and the contributions made.

Section 7.1 discusses the societal relevance and contribution, both for the case study organisation and for other organisations dealing with security patching, thus answering sub-question 5. A reflection on the scientific relevance and contributions is given in section 7.2, whereafter the limitations of this research and the recommendations for further research are discussed in section 7.3. Finally, the findings will be related to the CoSEM study program in section 7.4.

7.1 Reflection on societal relevance and contributions

7.1.1 Reflection on the case study organisation

Reflecting the findings on the objective of the case study organisation is able to provide insight into the value of this research. For this, written feedback and three evaluative interviews are held with IT practitioners of different layers of the organisation, both from operating teams and higher-level management. For this, a summary of the findings was presented to the participants. The goal of this validation was two-fold: it aimed to explore opinions and reactions on the findings and reflect on the goal of the IT department regarding patching to bring context to the findings.

Sub-question 5: How can the findings be used in practice, and how viable are the results to speed up an effective patching process?

There are several novel contributions to the case study organisations. As the quote at the beginning of this chapter states: "*Patching keeps being a difficult subject, often because theory does not match with what we experience in practice*", the findings in this research narrow this gap. First, the findings demonstrate the influence of socio-technical factors on the existing security patch process. This will contribute to a combined understanding of the challenges faced in the organisation throughout the entire IT department as this research includes perspectives of employees from different levels. Furthermore, the findings help understand where improvements can be made to achieve a more effective security patch process. The identification of tensions between different stakeholders helps to be aware of the dilemmas faced by IT practitioners. The *decision funnel* (section 4.3.1) is suitable when establishing processes and strategies as it informs the main aspects of the decisions to be made. Additionally, the results show the current state of measurement practices. This enhances the understanding of where improvements need to be made in order to reach a higher level of maturity in security patch management. The results additionally show the interplay between different types of challenges, both socially and technically. It helps to reason about improvements from a more governance perspective instead of purely technical, as the nature of the challenges are socially and organisationally embedded.

The evaluation brings some nuance to the findings and helps better understand the root causes of ineffective security patching: real change starts with a *shift in mindset*. The nature of the organisation makes that education and research are always the first priority. The organisational culture is established with this in mind, including the culture within the IT department. Currently, decisions by IT practitioners are generally being made to adhere to the values of clients. The lack of clear requirements and boundaries of the IT department results in decisions not being in balance with security. Where the IT department needs to be clear to end-users and system-owners, top-level management needs to set its requirements and boundaries to operational teams.

Where the recommendations in Chapter 6 are shaped from a governance perspective, the practical usefulness of these requires suitable behaviour of IT practitioners and system-owners. The functioning of the establishment of an organisational objective of effective patching and the definition of corresponding processes and strategies is limited if these are not adhered to by the people they are designed for. Behaviour and awareness of external system-owners are difficult to address, as a paradigm shift is needed in the values of security and availability. As recommended in this research, formalizing agreements on ownership accountability and responsibility is an important way of imposing this. The evaluative discussions furthermore reveal that changing the behaviour and way of thinking of IT practitioners might be even more troublesome. Although IT practitioners have the awareness and understanding of what the risks entail, the urgency to change is lacking. Literature shows that real change sometimes only allows to take place due to the occurrence of an incident. A scientific example is a study by Mayer, Zou, Schaub, and Aviv (2021), showing the changes in the behaviour of participants after the occurrence of a data breach. An industry example is the measure Maastricht University took after its data breach by implementing a Security Operation Center to monitor traffic day and night (Maastricht University, 2021). Where the findings in this research illustrate that this sense of urgency is present during emergency patches, this is not present otherwise. While this makes sense as risks are, in general lower, the example of Maastricht University (section 2.1.2) reveals that vulnerabilities which are not widely known or identified to be critical can be used to exploit. This evaluation thus reveals that human behaviour is the root cause of a technology-embedded problem.

7.1.2 Reflection on the generalizability to other organisations

Reflecting on the implications of this research to other organisations helps to understand the generalizability of the findings. First, the identified challenges of this research are not unique to this organisation, as many organisations are established in the same way. Especially those organisations where IT was added in later than the operating of the organisation. The results are thus able to provide organisations with similar maturity levels a better understanding of the challenges and decision-making process of security patching. Furthermore, as this research explicitly included contextual factors of the decentralised organisational structure, the research findings are to be used by other organisations with a similar structure where decision-power lays both at a centrally organised IT department as well as decentralised by individuals or groups of individuals. No conclusions can be made of the generalisability of findings related to the size of an organisation’s IT environment or the number of connected end-users.

The findings also indicate that challenges and problems will be inevitable, and no optimal solution is able to solve everything. The significance of these problems will, however, certainly decrease with a better understanding of the aspects that are concerned with decision-making. The results could increase the awareness and understanding of IT practitioners or IT managers to encourage evaluating their own security patching process. Although guidance ‘can only come so far’ due to the nature of this problem, it is able to improve the understanding to ease decision-making and conduct patching in a consistent way.

Findings furthermore show that the maturity level of an organisation plays an important role in the opportunities for measurement of security patching. Section 5.2 indicated the high dependency on the tools provided by software vendors for organisations with a low maturity level. Fiebig et al. (2021) discussed the reliance of universities on cloud infrastructures, where the authors see an "increase in Microsoft based mail hosting between late 2018 and early 2020" (p. 6) and "several universities already considered a migration to Microsoft products for their main domains" (p. 6) of universities in the Netherlands. Furthermore, an overview of universities' use of cloud providers (Amazon, Google, Microsoft) and email providers from January 2015 to June 2021 reveals the big share of Microsoft in universities in multiple countries. This implies that universities using Microsoft products benefit from the investigations in this research, for example from section 5.2.

Lastly, reflecting on the recommendations in this research and those provided by standardization and advisory bodies (see Chapter 3) indicates the need to evaluate security patching practices from a more socio-technical governance perspective, rather than focusing on technical aspects only. Technical solutions are not in order without considering behavioural and organisational components.

7.2 Reflection on scientific relevance and contributions

Two other studies (Dissanayake et al., 2022; Dissanayake et al., 2021) look at security patching from a socio-technical perspective. These are already the 'odd ones out' in a field where many studies on security patching look with a more detailed focus on a specific aspect of the process of deploying patches. For example, Nappa, Johnson, Bilge, Caballero, and Dumitras (2015) investigate patch deployment in client-side vulnerabilities and show that the patching rate is affected by user-specific and application-specific factors. Additionally, Li et al. (2019) investigate the stages of patch deployment and challenges for IT practitioners, but fail to include the role of coordination. This research therefore contributes to the understanding of security patching from a wider, more integrated socio-technical perspective.

Where Dissanayake et al. (2021) conceptualize socio-technical factors in causes, constraints, breakdowns, and mechanisms, they all tie it to the role of coordination. Tiefenau et al. (2020) explored the update behaviour of system administrators and identified obstacles. The novelty of the results in this research lay in the structuring of socio-technical factors to not only the role of coordination and behaviour, but to the decision-making process IT practitioners go through every time they patch. The decision funnel in Figure 17 is a new concept that builds on the overlapping tensions in security patching. This helps to examine how to simplify decision-making, as recommended by Li et al. (2019). The results of this study go further by linking the socio-technical factors to four aspects that influence the decision-making of security patching. The identification of tensions is not per se novel; for example Potter and Nieh (2005) introduce a new system "that enables unscheduled operating system updates while preserving application service availability" (p. 1). However, here the tension between security and availability is addressed by the introduction of a new system, but the decision-making aspect is not included.

Furthermore, the mapping and distinction of socio-technical factors on either the decision to deploy a patch or not, and the timeframe it is deployed in are new to the field of security patch management. For example, Dissanayake et al. (2022) focus their socio-technical factors on the ineffectiveness of security patching but do not make a distinction on whether it influences the timing of patch deployment or the decision to patch at all. Additionally, the abstraction of these socio-technical factors as either causes, barriers, or constraints of the decision space contributes to the understanding of their way of influence. This research contributes to understanding how challenges of different socio-technical categories interact, and what the effect of certain coping strategies entails. This research also contributes by revealing a distinction in decision-making between routine patching and emergency patching, as socio-technical factors are perceived differently in a normal state and in a state of emergency (section 4.2.4). The results identify the potential for organisations to act more effectively if in a state of emergency. Here, especially the organizational and procedural challenges such as communication and collaboration are less significant.

One of the recommendations for further research from Dissanayake et al. (2021) was to explore how organisational culture affects the role of coordination. This study included data from the entire IT department and illustrated the difference between maturity levels of different teams of the same organisation. The non-existence of clear organisation-wide policies results in different patch approaches between different teams. It was identified that IT teams are living on their own islands and are not aware of how security patching is carried out outside one's own team. Furthermore, this research explicitly takes into account the educational context where IT is more decentralised than in other organisations. A novelty is the insight into what the effect of endowing patching decision rights is on the effectiveness, and furthermore, the overall state of an organisation's security. Where Dissanayake et al. (2021) argue for the need for collaboration in an organisation where patching is carried out mainly centrally, this study adds to this aspect by providing more knowledge on the significance of the role of good coordination. This study also identified that collaboration in decentralised organisations can be non-existent. This illustrates that the dependency of other stakeholders is more significant in decentralised environments, therefore increasing the ineffectiveness of organisation-wide patch practices.

Lastly, the mixed methods approach explores and combines qualitative data from interviews with IT practitioners and quantitative data of patching activity logs; thus, using two different research approaches in a complementary way contributes to developing a better understanding of security patch management. As the synthesis in Chapter 6 reveals, security patch management is a suitable research domain for both qualitative and quantitative exploration. The qualitative data has contributed to exploring the socio-technical challenges present in the decision-making process of IT practitioners. Where Tiefenau et al. (2020) and Li et al. (2019) also use a mixed methods approach by combining qualitative interview data and quantitative survey data, this research differs in the type of quantitative data used. The investigation of quantitative data in this research has contributed to indicating the significance of several of these challenges, and furthermore indicated what is needed to reach a more effective process.

7.3 Limitations and recommendations for further research

The results illustrate that security patching is a *multi-stakeholder problem*, with high dependencies on external stakeholders as end-users and system-owners. The exploration in this research is conducted with IT practitioners of the central IT department as the objective was to better understand their behaviour and decision-making process. The behaviour and decision-making process of these external stakeholders is in this research assumed as being rigid, where in each situation these people express similar behaviour. It is recommended to explore this behaviour in more depth to understand the way of reasoning and prioritization of values. For example, what is needed for system-owners to become aware of the security risks their behaviour brings to the entire organisation? Furthermore, end-user investigation can help determine the perception of availability and patch moment trade-offs. For example, what elements influence the perception of end-users regarding organisational availability disruptions when compared to the risks of getting their personal data breached?

This research included data from one organisation, therefore being a *single case study*. Section 7.1.2 discussed the implications of this limitation, as no conclusions can be made of the generalisability of findings related to the size of an organisation's IT environment or the number of connected end-users. It is therefore recommended to do a *multi-case study* investigating multiple organisations of varying sizes and distinctive sectors. By comparing small and large organizations from both the public and private sector, a comparative understanding of security patching will be reached. The socio-technical challenges identified in this research could be used as a conceptual framework to assess the existence of these or to compare these to new challenges to be identified.

This research used a *mixed methods research* approach which allowed to use both qualitative and quantitative data to explore security patch management. The quantitative findings in this research did however not contribute to the comparison of the potential discrepancy between what IT practitioners think their security patch process entails and what the actual patching activity of IT practitioners entails. Where this research demonstrates the usefulness of qualitative data in understanding patching, it lacks the usage of quantitative data to understand the patching process itself. As stated in section 1.3, the case study selection was based on it being a large organisation where it was assumed it would have a mature patch policy in place, increasing the availability of quantitative data. However, the case study organisation had a low maturity level where measurements are not actively used to base decisions on. This research, therefore, disproves the earlier made assumption that large organisations necessarily have a mature patch process in place. This research is limited in providing insight into measurable patching activity. It is recommended to investigate an organisation with a higher level of maturity that actively collects and uses measurements to base decisions on. This will allow optimizing the advantages mixed methods research brings. This could additionally limit the existing limitation of the nature of qualitative interviews. Participants are potentially biased in the way they provide answers by presenting themselves in a certain way or by giving socially acceptable answers through pressure from higher-level management or other teams in the organisation. Comparison of quantitative patching activity would allow validation of these findings. This could be in the form of a *longitudinal case study*, to create a more extensive and in-depth understanding of patterns and fluctuations of data throughout time.

It is furthermore recommended to investigate the possibilities of quantification of the importance of each of the socio-technical challenges identified in this explorative study. As this research is solely based on the assumption that the number of interrelations a challenge has with other challenges determines the significance of challenges, this can only be assumed. As the synthesis in Chapter 6 already indicated an exemption, for example the complexity of systems, this is thus limited in its substantiation. Quantifying the identified challenges to their significance can help organisations to more effectively target the most important challenges. This will also allow organisations to make use of quantifiable elements to make decisions on business implementations. For example, could it be quantifiable that the number of hardware resources increases the frequency of patching by IT practitioners by X times?

Lastly, this research indicated the existence of the difference between emergency patching and routine patching. However, it did not become clear what criteria altered the state to be in an emergency as the interviews were the only source of data here. It is therefore recommended to do a field exploration of IT practitioners handling an emergency patch ‘in the moment’. What makes for example that the identified tensions in this research are really less significant? Do other factors play a role here, apart from the identified coping strategies? It is acknowledged that this might be difficult to establish as emergency patches are unpredictable. However, more insight into this distinction is helpful to potentially use similar results in routine patching practices.

7.4 Reflection on the link to the CoSEM Study Programme

Reflecting back on the initial link to the CoSEM study program (as discussed in section 1.3.3) indicates that the findings of this research demonstrate the multi-disciplinary complexity of security patch management. Chapter 3 illustrates the high dependency on behaviour and interactions with multiple internal and external stakeholders, one of the key characteristics of the problems dealt with in the program of Complex Systems Engineering and Management (CoSEM). Furthermore, the multitude of both social and technical challenges in Chapter 4 demonstrates the embedded nature of the problem of security patching. The socio-technical factors result in the existence of tensions in decision-making, resulting in the inability to easily come to an optimal solution. As proposed in Chapter 1, tackling these socio-technical problems requires technical, institutional, economic, and social knowledge of the functioning of this system. This research integrates these components to help both the industry and the scientific field to have a better understanding of security patch management. Chapter 5 furthermore elaborates on how quantitative data is able to help organisational process management strategies. Chapter 6 lastly synthesises the different findings in a systematic way, helping to better understand the interrelations of qualitative and quantitative data. In conclusion, this research shows that security patch management is a real governance problem requiring coordination and cooperation of multiple stakeholders in a dynamic environment.

8 | CONCLUSION

"In the world of IT there is always something going wrong at the last minute, that's guaranteed. That risk is always there, it is never zero, you can't convince me that either" – Interviewed IT practitioner

This chapter concludes the findings in the previous chapters to answer the main research question. Aiming to develop a deeper understanding of security patching behaviour in organisations that cause current practices to be ineffective, thus increasing security risks, this explorative study posed the following research question: *What socio-technical factors influence the effectiveness and timeliness of the security patching process in organisations?*

Organisations increasingly depend on software to process data, optimise workflows, or provide services to employees. As IT is often a facilitating part of organisations' core businesses, third-party software is regularly used to bring in expertise and keep costs and other resources low. Although third-party software makes it easier for organisations to manage their IT systems, it is not the silver bullet to carefree operations. Within software products, defects, flaws, or glitches can be present that create opportunities for threat actors to exploit. Vulnerability exploits could lead to asset and financial losses, productivity losses, reputation damage, or legal liabilities (Seemma et al., 2018). Using software thus brings risks to the operations of an organisation, making the practice of dealing with software part of risk management. A well-recognised and effective risk management strategy to mitigate software vulnerabilities and secure digital assets is software security patching, the practice of installing fixes to security vulnerabilities in software products and systems deployed in an organisation's IT environment (Dissanayake et al., 2022). This practice is, however, not as straightforward due to different interactions and dependencies influencing how IT practitioners make decisions. While some of these are technical in nature concerning IT systems, others are more social in nature concerning the objectives and behaviour of other stakeholders. These socio-technical influences result in organisations struggling to achieve this risk mitigation practice in a timely manner. The best practice to 'patch early and often' (Dissanayake et al., 2022) is often not satisfied, resulting in vulnerabilities in software being exposed for much longer than desired.

- *Sub-question 1: What existing standards, frameworks, or guidelines on security patching are available for organisations?*

Some publications are of explanatory nature and are designed to help organisations understand the basics of security patch management and stress the need for designing this the right way (CIS, 2009; DWP, 2021; Mell & Tracy, 2002; Ruppert, 2008; Souppaya & Scarfone, 2013; TBS, 2022). Other publications are of advisory nature and are designed as a reference and guidance document for organisations to develop their own organisational policies and implement information security controls (ACSC, 2021; BIR, 2014; Hoehl, 2013; ISO/IEC, 2013; Mell et al., 2005). The content of the publications increases the level of knowledge on security patch management and identifies what aspects are relevant for organisations to consider. These, therefore, help organisations to grasp an understanding of what security patch management entails, and thus succeed in assisting organisations to get started with the practice of security patching.

Nonetheless, the publications are limited in their ability to provide guidance on how to deal with the complexity of security patch management. Most recommendations are very generic and lack specified details, leaving the reader with a sense of incompleteness. This is further manifested in the lack of a unified indication of a timeframe to install a patch after release, as most recommendations only generally state patching needs to happen 'in a timely fashion'. Simultaneously the publications that do state a timeframe hold opposing views of what that timely fashion entails. Furthermore, many recommendations are formulated as decisions to be made by organisations rather than providing concrete guidance on how to do so. The socio-technical complexity of security patching is occasionally identified; however, it is not linked to the decision-making process of IT practitioners. Guidance on how to organise coordination issues, stakeholder dependencies, or communication mechanisms is not established. Fortunately, the most recent publication by Souppaya and Scarfone (2022) is more aligned with the modern challenges patching brings for organisations and therefore provides more valuable guidance. Here, the focus lays more on the involvement of different stakeholders and the conflicts of interest that play a role in planning for patch deployment, stressing the inevitable need to make decisions about dilemmas and trade-offs.

- *Sub-question 2: What decisions and trade-offs are being made by IT practitioners in organisational security patching, and what does this process look like?*

Security patching is a constant process of decision-making, weighing off factors, interests, and consequences in each phase. The main decision to be made in patching is twofold; to apply a patch or not, and in what timeframe. These decisions need to be made throughout the entire patch process, from information retrieval until patch deployment. This study maps this process as a *funnel of decision-making* that consists of four main aspects that play a significant role of influence on the outcome of this decision: security, applicability, operability, and availability. First the security aspect, the significance of the threat and the impact of not implementing the patch. This is the most significant consideration for the choice to install a patch. However, the applicability aspect, the eligibility of the vulnerability on the nature of the organisation's system, influences this decision. The severity of the vulnerability can be critical; if it is not applicable to the unique IT systems, the patch would not be installed. Third the operational aspect, the impact of implementing the patch. This consists of the consequences for the IT department and the risks of deploying the patch itself. Fourth the availability aspect, the impact patching has on clients, and the consequences it entails.

Within this *funnel of decision-making*, the presence of socio-technical factors is causing the decision-making process to be ineffective. This study identified that socio-technical factors are either causes that *shape* the decision space, barriers that *hinder* the decision space, or constraints that *limit* the decision space. Identifying emerging categories of these factors indicate the nature of these factors to be organisational, procedural, and technical. An example of a cause is the potential occurrence of side-effects during patch deployment that can harm a system's functioning, shaping the decision space from an operational aspect by for example delaying the decision to see if other organisations are experiencing problems. An example of a constraint is the organisational restrictions that limit the decision space of when to patch to the available change moments. An example of a barrier is the dependency of external system-owners to patch their own systems, hindering the decisions space of patch deployment. These socio-technical factors thus influence the decision-making process in their own way, but all contribute to the ineffectiveness of patch management.

The presence and interplay of these factors cause *tensions* to occur between the different aspects, wherefore IT practitioners make trade-offs in their decision, thus making the decision outcome very dependent on the situation. There is a tension between security and operability, as the risks of security are in friction with the operational risks of patch deployment. The causes for this are primarily technical, where side-effects, technical dependencies, hardware resources, or 'unpatchable' legacy systems play a role. A second tension is between security and availability, influencing the timeframe of patching. The causes for this are primarily procedural, where fixed maintenance moments, decision-making through multiple levels, and distributed decision rights play a significant role of importance. This tension is present due to the different objectives of IT practitioners and end-users. Where IT practitioners prioritize security, end-users prioritize availability. A third tension is between money and resources. Unlike the other two tensions, this is not a trade-off but a constraint. Similar to the second tension, this tension is present due to a difference in the prioritization of values, this time between IT practitioners and top-level management.

These tensions and socio-technical factors bring challenges to the practice of security patching. To deal with these, coping strategies are applied to decrease the significance of challenges and make patching more manageable. Relating back to the earlier stated examples, a coping strategy to reduce the risks of side effects during patch deployment is to test patches in a test-environment before deploying them in the production environment. This study revealed furthermore that certain coping strategies are only used when being in a state of *emergency*. The key aspect causing IT practitioners to be in a state of emergency is the security risks a vulnerability brings to the organisation. When it is known that the probability is high, for example when an exploit has been seen in the wild, security is prioritized. The decision funnel in a state of emergency does not differ from the normal decision funnel; decisions just have to be made quicker. Although the socio-technical factors and tensions are still present, they are less significant due to the objectives of both internal and external stakeholders being more aligned. The coping strategy of a higher level of cooperation with an emergency patch makes the tension between security and operability, and between security and availability less significant, as the stakeholders with decision-power are all ‘at the same side’.

Diving deeper into the interplay between the socio-technical factors reveals the complex nature of security patch management. This study indicated the following interrelations between challenges and coping strategies:

- Addressing one challenge can help to reduce the significance of another challenge.
- The existence of one challenge can cause another challenge to occur.
- As a coping strategy can reduce the significance of one challenge, it can be in conflict with another challenge.
- As a coping strategy can reduce the significance of one challenge, it can simultaneously increase the need to address other challenges.

Decision-making is thus influenced by tension and the presence of socio-technical factors related to different aspects. While coping strategies have the ability to reduce the significance of factors, coping strategies have externalities to other challenges. Where the previous findings are based on qualitative data through interviews with IT practitioners, this study additionally explored how measurements are able to provide additional insight into the significance of the effectiveness of patching and the challenges that influence this.

- *Sub-question 3: What is the quality of organisations’ logs to determine patch activities?*

Security metrics are a proven way to understand security risks; to spot emerging problems, to understand weaknesses in security infrastructures, measure the performance of countermeasure processes, and recommend technology and process improvements (Jaquith, 2007). An increased level of knowledge about organisations’ operations can help decision-making in the patch process. In order to come to metrics, measurements need to be of high quality to bring raw quantitative data to useful information. The case study in this research reveals that having *some* measurement in place helps to understand the existence of challenges and reveals the working of coping strategies.

Comparing the differences between the number of patches released by Microsoft to the number of patches received and approved by IT practitioners helps understand the context to make decisions in. The lower number of patches received illustrates the coping strategy of patch filtering to deal with the applicability aspect and the operational aspect of patching, e.g., the challenge of needing to balance patching and tasks of the day-to-day job of IT practitioners. Results of the analysis of patch behaviour of Windows servers over time illustrate the dependency of external system-owners to patch their own systems. The consistent presence of unpatched servers throughout all months demonstrates that the availability aspect weighs heavier than the security aspect for some system-owners. Furthermore, quantitative data results reveal the patch strategy of how is dealt with the challenge of limited resources and the availability coping strategy of always having some systems available for end-users, as numbers indicate patch deployment is performed in two sequential groups of systems.

An assessment of the quality of these measurements reveals a high dependency on tools provided by the software vendor to keep track of patching practices. These bring opportunities due to their ease of use and high compatibility with the IT systems. On the other hand, these bring limitations to the usefulness due to a lack of information on certain aspects. For example, provided tools were not able to give the distinction of deployed patches and their level of criticality, resulting in a lack of information about the security aspect. IT practitioners working on patches of software vendors that do not facilitate tools reveal a second discovery in this study: the struggle of establishing meaningful measurements.

Measurements are meaningful when able to provide new information that helps decision-making in the patch process. The usage of The Johari Window (Massie & Morris, 2011) as a means to consider the limitation of knowledge on measurement practices on the meaningfulness of decision-making reveals three main matters of attention. There is a *Blind area* of information which is unknown to IT practitioners but known to IT systems. Observations indicated that there is no objective truth present in the statistics indicated by their log systems. It is difficult to retrieve information on patches as many systems are uniquely designed, and technical dependencies are present. There is a *Hidden area* of information which is known to some, but unknown to others. Here there is information asymmetry between operational teams and top-level management, and between operational teams and other operational teams. Data on patching activity is scattered throughout the different teams in the IT department. Where each individual team quite clearly knows what is being measured, and more importantly, what is not measured, other operational teams and top-level management are not aware of this. This results in different levels of knowledge on what aspects of the patching activity are registered, and in what way. There is an *Unknown area* consisting of information not known to anyone in the organisation. Due to a lack of a target determining what effective security patching entails, the current practice of security patching is trying to fix just the right vulnerabilities in time.

- *Sub-question 4: How does measurement of patching activity relate to the socio-technical factors causing the tensions and dependencies found in the patching decisions?*

Synthesising both qualitative findings of sub-question 2 and quantitative findings of sub-question 3 helps to understand how the findings of this study are able to help the organisation in establishing more effective decision making. The convergence and interrelation of qualitative and quantitative components indicate a potential vicious cycle. Socio-technical challenges impact the effectiveness of patching, e.g., the higher the significance of challenges, the more ineffective the patch process. Challenges can, fortunately, be reduced by having the right measures and coping strategies in place, e.g., policies and procedures. The fit of these measures and coping strategies can be increased by insights quantitative data provides on the effectiveness of the patch process and the socio-technical challenges. The better the quality of quantitative data, the better the fit of measures. However, the quality of quantitative data is limited by the existence and significance of socio-technical factors. This indicates a reinforcing feedback loop between challenges, measures, and the quality of quantitative data. In this research, all challenges are assessed as being equally important and therefore equally contributing to the problem of ineffective patching. However, lumping all socio-technical factors together would suggest that targeting any random one would have an equal impact on the effectiveness. This is not the case; moreover, there is a significant distinction to be seen when relating the different challenges and their interactions together.

- *Sub-question 5: How can the findings be used in practice, and how viable are the results to speed up an effective patching process?*

Analysis of the findings hypothesises that not all socio-technical factors have the same level of significance, nor are all challenges as easily solved as others. The majority of these socio-technical factors influence the decision on the timeliness of the patch moment, thus the effectiveness of the patch process. Based on the assumption that the number of interrelations of a challenge with other challenges could indicate the significance of the challenge itself; the challenge of collaboration, coordination, and procedures and guidelines are most significant.

Assessment of the solvability of socio-technical challenges identified three levels. Challenges that could be fixed '*highly easy*' are those where implementations from within the IT department could already have a high level of influence. Examples of these are internal communication and coordination, organisational procedures and guidelines, and the challenge of the usage of internal automation tools. Challenges that fall into the category of being fixed '*moderately easy*' to '*not easy*' are those where a dependency on other stakeholders and their behaviour remains to be present. Examples of these are the behaviour and awareness of external system-owners, patch quality and availability, and the challenges the decentralised organisational structure bring. Lastly, there are challenges that could only be fixed '*to a certain extent, but not in entirety*', where the nature of the challenge does not allow it to be fixed. Examples of challenges in this category are the developments of the threat environment or the complexity of systems like legacy systems.

Under the assumption that it would be most feasible and beneficial to target those challenges that both have a high solvability and a high level of significance, three main challenges meet this criterium: communication, coordination, and procedures and guidelines. Targeting these challenges first will positively impact the effectiveness of the current security patching process, and as concluded from the interrelations of challenges in section 4.4, these will likely positively influence other challenges simultaneously. As these challenges are intertwined, there is a bundle of recommendations to target these all together:

- Fix the gap between wanting to do 'better', and knowing what 'better' entails
- Formalize agreements on ownership accountability and responsibility to deal with the decentralised nature of the organisation
- Define processes and strategies for different circumstances based on the decision funnel, including the aspects of security, applicability, operability, and availability
- Actively explore and document current limitations and organisation-wide opportunities to use measurements to keep track of the effectiveness

Evaluative discussions with IT practitioners on the recommendations reveal that the root cause of ineffective security patching is the need for a *shift in mindset*. The practical usefulness of these requires appropriate behaviour of IT practitioners and external system-owners. The functioning of the establishment of an organisational objective of effective patching and the definition of corresponding processes and strategies is limited if these are not adhered to by the people they are designed for.

REFERENCES

ACSC. (2021). *Assessing Security Vulnerabilities and Applying Patches*. Retrieved from https://www.cyber.gov.au/acsc/view-all-content/publications/assessing-security-vulnerabilities-and-applying-patches

Adams, W. (2015). Conducting semi-structured interviews. i KE Newcomer, HP Hatry, & JS Wholey, Handbook of practical program evaluation (ss. 492-505). In: Jossey-Bass a Wiley Imprint.

Al Maskari, S., Saini, D. K., Raut, S. Y., & Hadimani, L. A. (2011). *Security and vulnerability issues in university networks.* Paper presented at the Proceedings of the World Congress on Engineering.

Al-Alawi, A. I., Mehrotra, A. A., & Al-Bassam, S. A. (2020). Cybersecurity: Cybercrime Prevention in Higher Learning Institutions. In *Implementing Computational Intelligence Techniques for Security Systems Design* (pp. 255-274): IGI Global.

Alexander, R. D., & Panguluri, S. (2017). Cybersecurity terminology and frameworks. In *Cyber-Physical Security* (pp. 19-47): Springer.

Andrew, C. (2005). The five Ps of patch management. *computers & security, 5*(24), 362-363. doi:10.1016/j.cose.2005.06.005

August, T., Dao, D., & Kim, K. (2019). Market segmentation and software security: Pricing patching rights. *Management Science, 65*(10), 4575-4597. doi:https://doi.org/10.1287/mnsc.2018.3153

Automox, & AimPoint Group. (2020). *2020 Cyber Hygiene Report: What you need to know now - Lessons learned from a survey of the state of endpoint patching and hardening*. Retrieved from https://patch.automox.com/rs/923-VQX-349/images/Automox_2020_Cyber_Hygiene_Report-What_You_Need_to_Know_Now.pdf

Balbix. (2022). What are CVSS Scores. Retrieved from https://www.balbix.com/insights/understanding-cvss-scores/

Berinato, S., Daintry, D., Scalet, S., Wailgum, T., & Wheatley, M. (2004). *The ABC's of New Security Leadership*. Retrieved from CSO Online: http://www.csoonline.com/fundamentals/abc_leadership.html

BIR. (2014). Patch Management - Guideline 12.6. Retrieved from https://www.cip-overheid.nl/media/1163/bid-operationale-producten-bir-008-patch-management-10.pdf

Birch, S. S. (2015). IBM's CEO on hackers: "Cyber crime is the greatest threat to every company in the world". Retrieved from https://www.ibm.com/blogs/nordic-msp/ibms-ceo-on-hackers-cyber-crime-is-the-greatest-threat-to-every-company-in-the-world/

Boehm, J., Merrath, P., Poppensieker, T., Riemenschnitter, R., & Stäle, T. (2018). *The holistic approach to managing cyber risk proceeds from a top-management*

overview of the enterprise and its multilayered risk landscape. Retrieved from https://www.mckinsey.com/business-functions/risk-and-resilience/our-insights/cyber-risk-measurement-and-the-holistic-cybersecurity-approach

Brandman, G. (2005). Patching the Enterprise: Organizations of all sizes are spending considerable efforts on getting patch management right - their businesses depend on it. *Queue, 3*(2), 32–39. doi:https://doi.org/10.1145/1053331.1053344

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology, 3*(2), 77-101. doi:10.1191/1478088706qp063oa

Bresnick, P. (2021). 4 Reasons Cyber Criminals Are Targeting Higher Education: Part 1. Retrieved from https://www.fierceeducation.com/best-practices/4-reasons-cyber-criminals-are-targeting-higher-education-part-1

Cáceres, J., Vaquero, L. M., Rodero-Merino, L., Polo, A., & Hierro, J. J. (2010). Service scalability over the cloud. In *Handbook of Cloud Computing* (pp. 357-377): Springer.

Cavusoglu, H., Cavusoglu, H., & Zhang, J. (2006). *Economics of Security Patch Management.* Paper presented at the WEIS.

Chan, J. (2004). Essentials of Patch Management Policy and Practice. Retrieved from http://www.patchmanagement.org/

Chua, Y. T., Parkin, S., Edwards, M., Oliveira, D., Schiffner, S., Tyson, G., & Hutchings, A. (2019). *Identifying unintended harms of cybersecurity countermeasures.* Paper presented at the 2019 APWG Symposium on Electronic Crime Research (eCrime).

CIS. (2009). Twenty Critical Controls for Effective Cyber Defense: Consensus Audit Guidelines. Retrieved from https://csis-website-prod.s3.amazonaws.com/s3fspublic/legacy_files/files/publication/Twenty_Critical_Controls_for_Effective_Cyber_Defense_CAG.pdf

CIS. (2022). About Us. Retrieved from https://www.cisecurity.org/about-us

Creswell, J. W., & Clark, V. L. P. (2017). *Designing and conducting mixed methods research*: Sage publications.

Crowe, S., Cresswell, K., Robertson, A., Huby, G., Avery, A., & Sheikh, A. (2011). The case study approach. *BMC medical research methodology, 11*(1), 1-9. doi:https://doi.org/10.1186/1471-2288-11-100

de Bruijne, M., van Eeten, M., Gañán, C. H., & Pieters, W. (2017). *Towards a new cyber threat actor typology*. Retrieved from http://hdl.handle.net/20.500.12832/2299

Dearnley, C. (2005). A reflection on the use of semi-structured interviews. *Nurse researcher, 13*(1). doi:10.7748/nr2005.07.13.1.19.c5997

Dijkstra, M., & van Dantzig, M. (2020). *Spoedondersteuning Project Fontana*. Retrieved from https://www.rijksoverheid.nl/documenten/rapporten/2020/02/05/reactie-universiteit-maastricht-op-rapport-fox-it

Dissanayake, N., Jayatilaka, A., Zahedi, M., & Babar, M. A. (2022). Software security patch management-A systematic literature review of challenges, approaches, tools and practices. *Information and Software Technology, 144*, 106771. doi:https://doi.org/10.1016/j.infsof.2021.106771

Dissanayake, N., Zahedi, M., Jayatilaka, A., & Babar, M. A. (2021). *A grounded theory of the role of coordination in software security patch management.* Paper presented at the Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.

DWP. (2021). *Security Standard – Security Patching (SS-033)*. Retrieved from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/957088/dwp-ss033-security-standard-security-patching-v1.3.pdf

Elina, D. (2021). IT Maturity Model – Why Should a Company Undertake ITIL Assessment Process. Retrieved from https://www.itil-docs.com/blogs/asset-management/it-maturity-model-why-should-a-company-undertake-itil-assessment-process

Embroker. (2022). 2022 Must-Know Cyber Attack Statistics and Trends. Retrieved from https://www.embroker.com/blog/cyber-attack-statistics/

ENISA, & CERT-EU. (2022). Boosting your Organisation's Cyber Resilience - Joint Publication. Retrieved from https://www.enisa.europa.eu/publications/boosting-your-organisations-cyber-resilience

Eriksen-Jensen, M. (2013). Holding back the tidal wave of cybercrime. *Computer Fraud & Security, 2013*(3), 10-16. doi:https://doi.org/10.1016/S1361-3723(13)70028-9

Fernandez, O., & Azorin, J. (2011). *The use of mixed methods research in the field of behavioral sciences* (Vol. 45).

Fiebig, T., Gürses, S., Gañán, C. H., Kotkamp, E., Kuipers, F., Lindorfer, M., . . . Sari, T. (2021). Heads in the clouds: measuring the implications of universities migrating to public clouds. *arXiv preprint arXiv:2104.09462*.

Flick, U., Von Kardorff, E., & Steinke, I. (2004). What is qualitative research? An introduction to the field. *A companion to qualitative research*, 3-11.

Gerace, T., & Mouton, J. (2004). *The challenges and successes of implementing an enterprise patch management solution.* Paper presented at the Proceedings of the 32nd Annual ACM SIGUCCS conference on User services.

Gianini, G., Cremonini, M., Rainini, A., Cota, G. L., & Fossi, L. G. (2015). *A game theoretic approach to vulnerability patching.* Paper presented at the 2015 International Conference on Information and Communication Technology Research (Ictrc).

Hafeez, U. U., Karve, A., Dumba, B., Gandhi, A., & Zeng, S. (2019). *Towards Automated Patch Management in a Hybrid Cloud.* Paper presented at the International Conference on Service-Oriented Computing.

Halcomb, E. J. (2019). Mixed methods research: The issues beyond combining methods.

Heiser, J. (2003). The perils of security patch management. *Network Security, 7*, 9-12. doi:10.1016/S1353-4858(03)00709-8

Hoehl, M. (2013). Framework for building a comprehensive enterprise security patch management program. *STI Graduate Student Research SANS.*

IBM. (2021). Cost of a Data Breach Report 2021. Retrieved from https://www.ibm.com/downloads/cas/OJDVQGRY

IBM. (2022). *X-Force Threat Intelligence Index 2022.* Retrieved from https://www.ibm.com/downloads/cas/ADLMYLAZ

ISO/IEC. (2013). Information technology — Security techniques — Code of practice for information security controls. Retrieved from https://www.iso.org/standard/54533.html

ISO/IEC. (2018). ISO/IEC 27000:2018 - Information technology — Security techniques — Information security management systems — Overview and vocabulary. Retrieved from https://akela.mendelu.cz/~lidak/IPI/ISO_IEC_27000_2018.pdf

Jaquith, A. (2007). *Security metrics: replacing fear, uncertainty, and doubt*: Pearson Education.

Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of mixed methods research, 1*(2), 112-133. doi:https://doi.org/10.1177/1558689806298224

Li, F., Rogers, L., Mathur, A., Malkin, N., & Chetty, M. (2019). *Keepers of the machines: Examining how system administrators manage software updates for multiple machines.* Paper presented at the Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019).

Maastricht University. (2021). Op weg naar een veilige cyberwereld. Retrieved from https://www.maastrichtuniversity.nl/nl/nieuws/op-weg-naar-een-veilige-cyberwereld

Massie, M. J., & Morris, A. T. (2011). *Risk acceptance personality paradigm: How we view what we don't know we don't know.* Paper presented at the AIAA Infotech@ Aerospace Conference.

Mayer, P., Zou, Y., Schaub, F., & Aviv, A. J. (2021). *" Now I'm a bit {angry:}" Individuals' Awareness, Perception, and Responses to Data Breaches that Affected Them.* Paper presented at the 30th USENIX Security Symposium (USENIX Security 21).

Mell, P., Bergeron, T., & Henning, D. (2005). Creating a Patch and Vulnerability Management Program. Retrieved from https://csrc.nist.rip/library/alt-SP800-40v2.pdf

Mell, P., & Tracy, M. C. (2002). *Procedures for Handling Security Patches: Recommendations of the National Institute of Standards and Technology*. Retrieved from

Microsoft. (n.d.). Security Update Guide. Retrieved from https://msrc.microsoft.com/update-guide

Morgan, S. (2020, 18-11-2020). Cybercrime To Cost The World $10.5 Trillion Annually By 2025. *Cybersecurity Ventures.* Retrieved from https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/

Nappa, A., Johnson, R., Bilge, L., Caballero, J., & Dumitras, T. (2015). *The attack of the clones: A study of the impact of shared code on vulnerability patching.* Paper presented at the 2015 IEEE symposium on security and privacy.

NCSC. (n.d.). Wettelijke taak. Retrieved from https://www.ncsc.nl/over-ncsc/wettelijke-taak

NIST. (2022a). Cybersecurity Framework. Retrieved from https://www.nist.gov/cyberframework

NIST. (2022b). Glossary Retrieved from https://csrc.nist.gov/glossary/term/system#:~:text=An%20information%20system%20is%20a,dissemination%2C%20or%20disposition%20of%20information

NIST. (2022c). NATIONAL VULNERABILITY DATABASE. Retrieved from https://nvd.nist.gov/vuln-metrics/cvss#:~:text=The%20Common%20Vulnerability%20Scoring%20System,Base%2C%20Temporal%2C%20and%20Environmental.

Ponemon Institute. (2020). Costs and Consequences of Gaps in Vulnerability Response. Retrieved from https://www.servicenow.com/thank-you/lpayr/ponemon-vulnerability-survey.html?elqUniqueCampainId=ecb36d6f-1217-43e3-a717-117db34d018e

Poppensieker, T., & Riemenschnitter, R. (2018). *A new posture for cybersecurity in a networked world.* Retrieved from https://www.mckinsey.com/business-functions/risk-and-resilience/our-insights/a-new-posture-for-cybersecurity-in-a-networked-world

Potter, S., & Nieh, J. (2005). *Reducing downtime due to system maintenance and upgrades.* Paper presented at the Proc. of the 19th USENIX Systems Administration Conf.

Reynolds, G. (2014). *Ethics in information technology*: Cengage learning.

Ruppert, B. (2008). *Patch Management*. Retrieved from https://www.sans.org/white-papers/2064/

SANS. (2022). About SANS Institute. Retrieved from https://www.sans.org/about/?msc=main-nav

Scholz, S., Hagen, W., & Lee, C. (2020). The Increasing Threat of Ransomware in Higher Education. *Cybersecurity and Privacy.* Retrieved from https://er.educause.edu/articles/2021/6/the-increasing-threat-of-ransomware-in-higher-education

Schouten, D., & Bomers, L. (2021, 21-07-2021). De nachtmerrie van ransomware: hoe hackers afkopen voor de Universiteit van Maastricht in 2019 de enige oplossing was. *EenVandaag*. Retrieved from https://eenvandaag.avrotros.nl/item/de-nachtmerrie-van-ransomware-hoe-hackers-afkopen-voor-de-universiteit-van-maastricht-in-2019-de-enige-oplossing-was/

SecureWorld. (2019). 8 Great Cybersecurity Quotes from SecureWorld Chicago. Retrieved from https://www.secureworld.io/industry-news/8-great-cybersecurity-quotes

Seemma, P., Nandhini, S., & Sowmiya, M. (2018). Overview of cyber security. *International Journal of Advanced Research in Computer and Communication Engineering, 7*(11), 125-128. doi:10.17148/IJARCCE.2018.71127

Sihvonen, H.-M., & Jäntti, M. (2010). *Improving release and patch management processes: An empirical case study on process challenges.* Paper presented at the 2010 Fifth International Conference on Software Engineering Advances.

Singh, U. K., Joshi, C., & Gaud, N. (2016). Measurement of security dangers in university network. *International Journal of Computer Applications, 155*(1), 6-10.

Souppaya, M., & Scarfone, K. (2013). Guide to enterprise patch management technologies. *NIST Special Publication, 800*, 40.

Souppaya, M., & Scarfone, K. (2022). *SP 800-40 Rev. 4 - Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology*. Retrieved from https://csrc.nist.gov/publications/detail/sp/800-40/rev-4/final

Suby, M. P. (2018). Missing Critical Patches: A Cybersecurity Epidemic

Endpoint Security Hygiene Practices Fail to Keep Up with IT Priorities. *Frost & Sullivan.* Retrieved from https://www.ncsi.com/wp-content/uploads/2021/01/Whitepaper-Missing-Critical-Patches-A-Cybersecurity-Epidemic.pdf

TBS. (2022). Patch Management Guidance. Retrieved from https://www.canada.ca/en/government/system/digital-government/online-security-privacy/patch-management-guidance.html#shr-pg0

Tiefenau, C., Häring, M., Krombholz, K., & Von Zezschwitz, E. (2020). *Security, availability, and multiple information sources: Exploring update behavior of system administrators.* Paper presented at the Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020).

Tracy, M. C., Jansen, W., Scarfone, K. A., & Butterfield, J. (2007). Guidelines on electronic mail security. *NIST Special Publication*.

UCISA. (n.d.). *UCISA Information Security Management Toolkit*. Retrieved from https://www.ucisa.ac.uk/ismt

Voas, J. M. (2020). The" Patching" Mentality. *Computer, 53*(7), 12-13. doi:10.1109/MC.2020.2991277

Von Solms, R., & Van Niekerk, J. (2013). From information security to cyber security. *computers & security, 38*, 97-102. doi:https://doi.org/10.1016/j.cose.2013.04.004

Walkowski, M., Krakowiak, M., Oko, J., & Sujecki, S. (2020). *Distributed analysis tool for vulnerability prioritization in corporate networks.* Paper presented at the 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM).

Watson, D. L., & Jones, A. (2013). *Digital forensics processing and procedures: Meeting the requirements of ISO 17020, ISO 17025, ISO 27001 and best practice requirements*: Newnes.

Werlinger, R., Hawkey, K., & Beznosov, K. (2008). *Human, Organizational and Technological Challenges of Implementing Information Security in Organizations.* Paper presented at the Proceedings of the Second International Symposium on Human Aspects of Information Security & Assurance (HAISA 2008).

White, S. K. (2021). What is CMMI? A model for optimizing development processes. Retrieved from https://www.cio.com/article/274530/process-improvement-capability-maturity-model-integration-cmmi-definition-and-solutions.html

Whitman, M. E., & Herbert, J. (2009). Mattord Principles of information security. *Cengage Learning EMEA, 598.*

Willis, V. (2020). What is the Best Vulnerability and Patch Management Process? Retrieved from https://www.automox.com/blog/vulnerability-patch-management-process

Wolford, B. (2022). What is GDPR, the EU's new data protection law? Retrieved from https://gdpr.eu/what-is-gdpr/

APPENDICES

Appendix A – Overview of recommendations

This appendix provides an **overview of all relevant statements** in the publications in Table A.14.

Table A.14 - Overview of publications

	Date	Body	Author(s)	Publication title
Standardisation	**2002**	NIST Special Publication (SP) 800-40	(Mell & Tracy, 2002)	Procedures for Handling Security Patches: Recommendations of the National Institute of Standards and Technology
	2005	NIST Special Publication (SP) 800-40 Version 2	(Mell et al., 2005)	Creating a Patch and Vulnerability Management Program.
	2013	NIST Special Publication (SP) 800-40 Revision 3	(Souppaya & Scarfone, 2013)	Guide to Enterprise Patch Management Technologies.
	2022	NIST Special Publication (SP) 800-40 Revision 4	(Souppaya & Scarfone, 2022)	Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology
	2013	ISO27002:2013, 2013	(ISO/IEC, 2013)	Information technology — Security techniques — Code of practice for information security controls
Advisory	**2008**	SANS Whitepaper	(Ruppert, 2008)	Patch Management
	2013	SANS Whitepaper	(Hoehl, 2013)	Framework for building a Comprehensive Enterprise Security Patch Management Program
	2009	CIS 20	(CIS, 2009)	Twenty Critical Controls for Effective Cyber Defense: Consensus Audit Guidelines
Governmental	**2014**	Baseline Informatiebeveiliging Rijksdienst (BIR)	(BIR, 2014)	Patch Management - Guideline 12.6
	2021	UK's Department for Work & Pension (DWP)	(DWP, 2021)	Security Standard – Security Patching (SS-033)
	2021	Australian Cyber Security Centre (ACSC)	(ACSC, 2021)	Assessing Security Vulnerabilities and Applying Patches
	2022	Treasury Board of Canada Secretariat	(TBS, 2022)	Patch Management Guidance

A.1 Policy, process, and coordination

Table A.15 - Overview of process and coordination issues

	Guideline	Publication
Policy	For all organizations, the first step to a successful patch management program is a patch policy. • The policy must align with business objectives and provides the authority to advance security patching. • The policy should contain a few key components to be effective. • The scope of what must be patched must be clearly described. Scope can be determined by data classification, asset value, location, and business purpose. • Establishing prioritization and timing targets are vital for determining when security updates are to be in place. • Procedures for obtaining exemption and who can authorize the exemption (and ultimately accept risk) must be clear. • Some form of risk register should be referenced to track the exemptions including authorization and expiration. Ideally, the policy references risk management policies and practices.	(Hoehl, 2013, pp. 10-11)
	There are many reasons for patch management program failure. These include: • No Corporate policy requiring patching • No clear understanding of roles and responsibilities associated with patching • Wrong expectations of scope • Poor software lifecycle management (EOL software not removed, multiple releases of same software installed with different versions, etc.) • Attempting to use one solution for all needs • No release or change management maturity • No tools or automation to support process in a repeatable manner • No computer build standard or accreditation for new computers	(Hoehl, 2013, p. 16)
	Organizations should balance their security needs with their needs for usability and availability.	(Souppaya & Scarfone, 2013)
	Organizations should have an explicit and documented patching and vulnerability policy as well as a systematic, accountable, and documented set of processes and procedures for handling patches. The patching and vulnerability policy should specify what techniques an organization will use to monitor for new patches and vulnerabilities and which personnel will be responsible for such monitoring. An organization's patching process should define a method for deciding which systems get patched and which patches get installed first. It should also include a methodology for testing and safely installing patches.	(Mell & Tracy, 2002)
	To develop a successful patch-management strategy, it is best to form a committee involving Change Management, I.T. Operations, Business and I.T. Directors, Information Security, and Audit.	(Ruppert, 2008, p. 22)
	The document must include scope, roles and responsibilities, timeline, functional guidelines, and procedures. The scope is used to outline what systems are addressed with patching. It should include reference to servers and desktops and flavors of operating systems. The scope can also be used to define a high-level timeline of patching efforts such as monthly or quarterly. The roles and responsibilities should specify actual groups or persons required to perform a function. This is required to ensure accountability and provides reference for individuals not directly involved with the patching efforts. The timeline should include events before, during and after patching.	(Ruppert, 2008, p. 23)
Process	[A process is designed to control technical vulnerabilities; which includes periodic penetration testing, risk analysis of vulnerabilities and patching.]	(BIR, 2014) 12.6
	In addition, there are administrative activities occurring throughout the software vulnerability management life cycle, such as updating documentation, audit logging, and generating actionable insights and reports as part of enterprise change management. Having robust change management policies and processes in place is a fundamental part of software vulnerability management.	(Souppaya & Scarfone, 2022)

Accountability	It is important to define roles and responsibilities. These responsibilities should include a patching coordinator, patching administrator, system or application support, systems monitor, and patching auditor.	(Ruppert, 2008, p. 16)
	A patch management process requires proper accountability and ownership, along with good governance and stewardship.	(TBS, 2022)
	Create a patch and vulnerability group (PVG) to facilitate the identification and distribution of patches within the organization	(Hoehl, 2013, p. 10)
	Roles and responsibilities associated with technical vulnerability management, including vulnerability monitoring, vulnerability risk assessment, patching, asset tracking and any coordination responsibilities required should be defined and established	(ISO/IEC, 2013) section 12.6.1
Communication	• Communication within the organization before, during, and after patching is vital. • ISO 27002 Section 16.2.1.a advises, “the organization should define and establish the roles and responsibilities associated with technical vulnerability management, including vulnerability monitoring, vulnerability risk assessment, patching, asset tracking and any coordination responsibilities required” (ISO, 2013). • Consider the use of a RACI chart to clarify roles and responsibilities. This helps prevent communication breakdown because of gaps or uncertainty with patch management duties.	(Hoehl, 2013, p. 16)
Standardisation	Consider using standardized configurations for IT resources.	(Hoehl, 2013, p. 8)
Measures	Organizations should implement and use appropriate measures for their enterprise patch management technologies and processes.	(Souppaya & Scarfone, 2013) Section 5
Principles	The 12 steps to patch management: 1. **Establish Importance** - how does patch management relate to our business objectives? What impact could a compromise of our information systems have on our enterprise? 2. **Define Scope** – identifying critical information assets/systems that need to be patched; cover systems and supporting infrastructure that are connected with the business objectives. 3. **High Level Policy** – define high level security objectives and develop a policy specific to patch management. This will help demonstrate management’s commitment and provide reference to patch management standards. 4. **Establish Security Organization** – develop a patch management committee that will own the policy and be responsible for implementing patch management. 5. **Identify & Classify** – identify all assets that impinge on the Information Security Management System (ISMS) and classify them from highest to lowest priority. 6. **Identify & Classify Risks** – perform a risk analysis of your company’s assets identified in step 5. 7. **Plan for Risk Management** – prepare for a risk treatment plan based on threats and vulnerabilities. Prioritize risks from high to low. 8. **Implement Risk Mitigation Strategy** – implement a patch management strategy based on the plan outlined from step 7. 9. **Statement of Applicability** – provide a list of every control related to patch management and compare to the ISMS suggested controls; provide gap analysis. 10. **Training & Security Awareness** – provide patch management security training for management, staff, and maintenance groups. 11. **Monitor & Review** – utilize patch management tools to capture activity logs and audit the logs to ensure compliance 12. **Maintain & Improve** – hold routine match management reviews with the committee; improve processes and correct outstanding issues.	(Ruppert, 2008, pp. 9-10)

	The recommendations support the following principles, which organizations should strive to adopt in their enterprise patch management practices: • **Problems are inevitable; be prepared for them.** Risk responses, including patching, will never be perfect. Some may inadvertently cause operational problems, for example, but most will not. To improve enterprise patch management, organizations need to change their culture so that instead of fearing problems and thus delaying risk responses, personnel are prepared to address problems when they occur. The organization needs to become more resilient, and everyone in the organization needs to understand that problems caused by patching are a necessary inconvenience that helps prevent major compromises. • **Simplify decision making.** Conducting a risk assessment of each new vulnerability in order to plan the optimal risk response for it is simply not feasible. Organizations do not have the time, resources, expertise, or tools to do so. Planning needs to be done in advance so that when a new vulnerability becomes known, a decision can quickly be made about how to respond to it. • **Rely on automation.** There is no way that an organization can keep up with patching without automation because of the sheer number of assets, software installations, vulnerabilities, and patches. Automation is also needed for emergency situations, like patching a severe vulnerability that attackers are actively exploiting. Having automation in place gives an organization agility and scalability when it comes to its risk responses. • **Start improvements now.** Some of the changes that an organization may need to make might take years to put in place, but that does not mean that other practices cannot be improved in the meantime.	(Souppaya & Scarfone, 2022)
	Organizations should strive to decrease the number of vulnerabilities introduced into their environments. This shrinks the attack surface and can lower the amount of patching that organizations need to do. Possible methods for decreasing the number of vulnerabilities include: • Harden software, such as enforcing the principles of least privilege and least functionality (e.g., deactivating or uninstalling software services, features, and other components that are not needed). • Acquire software that is likely to have fewer vulnerabilities over time compared to other software. • Work with software development partners that are likely to introduce fewer vulnerabilities into software over time, taking into consideration factors such as how rigorous their secure software development practices are, how quickly they address issues and release patches, how often problems are associated with their patches, and how transparent they are in their security-related communications. • Use managed services instead of software when feasible. • Select stacks or platforms that are likely to have fewer vulnerabilities over time compared to other stacks or platforms (e.g., running software within a small container instead of a larger operating system).	(Souppaya & Scarfone, 2022)
	Organizations should define a maintenance plan for each maintenance group for each applicable risk response scenario. A maintenance plan defines the actions to be taken when a scenario occurs for a maintenance group, including the timeframes for beginning and ending each action, along with any other pertinent information. Along with the maintenance plans, organizations should define a risk assessment process for determining which plan should be used at any given time and for deciding when to switch from one plan to another as the understanding of risk changes.	(Souppaya & Scarfone, 2022)
	Organizations should consider deploying applications in ways that make patching less likely to disrupt operations. One example is to run applications on stacks or platforms where patching is a fundamental part of the deployed technology and is less likely to disrupt operations (e.g., modernizing and running software within cloud-based containers instead of on-premises server operating systems). Another example is to take advantage of existing toolchains that already build applications with updated components and test them before production release.	(Souppaya & Scarfone, 2022)
	The focus of patch-management should be to establish routine, maintain consistency, expand awareness, extend communication, and embrace business and I.T. support.	(Ruppert, 2008, p. 36)

Scenarios	**Organizations should define the software vulnerability risk response scenarios they need to be prepared to handle.** Examples of such scenarios include: • **Routine patching.** This is the standard procedure for patches that are on a regular release cycle and have not been elevated to emergency status. Most patching falls under this scenario. However, because routine patching does not have the urgency of emergency scenarios, and routine patch installation can interrupt operations (e.g., device reboots), it is often postponed and neglected. This provides many additional windows of opportunity for attackers. Delaying routine patching also makes emergency patching more difficult, time-consuming, and disruptive because of the need to first install previous patches that new patches depend upon. • **Emergency patching.** This is the procedure to address patching emergencies in a crisis situation, such as a severe vulnerability or a vulnerability being actively exploited. If one or more of the organization's vulnerable assets have already been compromised, emergency patching may be part of incident response efforts. Emergency patching needs to be handled as efficiently as possible to prevent the imminent exploitation of vulnerable assets. • **Emergency mitigation.** This is the emergency procedure in a crisis situation, like those described above for the emergency patching scenario, to temporarily mitigate vulnerabilities before a patch is available. The mitigation can vary and may or may not need to be rolled back afterward. Emergency mitigations are sometimes needed because of issues with a patch. For example, a patch might be flawed and not actually correct a vulnerability, or a patch might inadvertently disrupt the operation of other software or systems. A patch could even be compromised. • **Unpatchable assets.** This is the implementation of isolation or other methods to mitigate the risk of systems that cannot be easily patched. This is typically required if routine patching is not able to accommodate these systems within a reasonable time frame. Examples of why an asset may be unpatchable include the vendor not providing patches (e.g., asset is at end-of-life, asset does not support updates) or an asset needing to run uninterrupted for an extended period of time because it provides mission-critical functions. Unpatchable assets need to be included in risk response planning because a new vulnerability in an asset might necessitate a change in the methods needed to mitigate its risk.	(Souppaya & Scarfone, 2022)
	Organizations should plan to implement multiple types of long-term risk mitigation methods besides patching to protect vulnerable assets. There should be an approved set of methods for each maintenance group, and these methods should have been reviewed and analyzed in advance by security architects/engineers to determine their adequacy in mitigating risk.	(Souppaya & Scarfone, 2022)
	Organizations should plan to implement multiple types of mitigations to protect vulnerable unpatchable assets. In addition to using long-term risk mitigation methods for unpatchable assets, organizations should also implement mitigations as needed to prevent exploitation of specific vulnerabilities that the long-term risk mitigation methods don't adequately address.	(Souppaya & Scarfone, 2022)
	Organizations should plan on periodically reevaluating their alternatives to patching. There are two main aspects to this. One is conducting a risk assessment to see if the alternatives to patching are still sufficiently effective at mitigating risk. The other is conducting a cost-benefit analysis to see if the assets provide sufficient value to the organization compared with the additional costs of mitigating, transferring, or accepting the risk of unpatchable assets.	(Souppaya & Scarfone, 2022)
	Organizations should closely track and monitor all exceptions to maintenance plans. As explained in Section 3.4, maintenance groups should be defined to minimize assets considered "exceptions." However, having some exceptions is inevitable. All exceptions to maintenance plans should be reviewed regularly to determine if the maintenance plan can be implemented now. Assets with similar long-term exceptions might need to be moved to a separate maintenance group with its own maintenance plan.	(Souppaya & Scarfone, 2022)

A.2 Asset Management

Table A.16 - Overview of asset management aspects

		Guideline	Publication
Inventory	Information type(s)	Section 3.4.1: The inventory of the patchable software (applications and operating systems) installed on each host should include not only which software is currently installed on each host, but also what version of each piece of software is installed.	(Souppaya & Scarfone, 2013)
		A current and complete inventory of assets (see Clause 8) is a prerequisite for effective technical vulnerability management. Specific information needed to support technical vulnerability management includes the software vendor, version numbers, current state of deployment (e.g. what software is installed on what systems) and the person(s) within the organization responsible for the software.	(ISO/IEC, 2013) section 12.6.1
		on having a current and complete inventory of the patchable software (applications and operating systems) installed on each host. This inventory should include not only which software is currently installed on each host, but also what version of each piece of software is installed. Without this information, the correct patches cannot be identified, acquired, and installed. This inventory information is also necessary for identifying older versions of installed software so that they can be brought up to date. A major benefit of updating older versions is that it reduces the number of software versions that need to be patched and have their patches tested.	(Souppaya & Scarfone, 2013, p. 6)
		It is sometimes helpful to add metadata to the inventory database that cannot be harvested directly from the asset. Information such as geographic location, data classification, and redundancy is valuable when performing initial risk assessment. Details about compensating controls (e.g., Host Intrusion Prevention) might also be valuable when stored within the asset inventory. For example, the risk assessment for a single patch might be different for a desktop PC used for word processing located in the Corporate Offices behind a firewall and IPS as compared to a laptop containing confidential information connected directly to the Internet.	(Souppaya & Scarfone, 2013, p. 6)
		A record of all assets MUST be maintained along with their patch status, history, and next review date where appropriate e.g. in a Configuration Management Database (CMDB), AWS Inventory Manager or equivalent. For automated patching via upgrades, a record of the reason for rejecting a product update must be maintained and reviewed on a regular basis, along with a risk assessment.	(DWP, 2021) Section 10.3.5
		Inventory the organization's IT resources to identify the hardware equipment, operating systems, and software applications that are used within the organization.	(Mell et al., 2005)
		Organizations should approach patching from a per-asset perspective. Software inventories should include information on each computing asset's technical characteristics and mission/business characteristics. Making decisions for risk responses and their prioritization should not be based solely on which software and software versions are in use. Each asset has technical and mission/business characteristics that should be taken into consideration because they provide context for the vulnerable software running on that asset.	(Souppaya & Scarfone, 2022)
		The characteristics that an organization should inventory will vary, but the following are examples of possible characteristics to track: • The asset's platform type (e.g., IT, OT, IoT, mobile, cloud, VM) • The party who administrates the asset (e.g., IT department, third party, end user, vendor/manufacturer, shared responsibility model) • The applications, services, or other mechanisms used to manage the asset (e.g., endpoint management software, virtual machine manager, container management software) • The asset's network connectivity in terms of protocols, frequency/duration, and bandwidth • The technical security controls already in place to safeguard the asset • The asset's primary user(s) or interconnected services and their privileges	(Souppaya & Scarfone, 2022)
		Examples of mission/business characteristics that an organization should track include: • The asset's role and importance to the organization, which are contextual and may be hard to define or determine	(Souppaya & Scarfone, 2022)

		• Laws, regulations, or policies that specify how soon a new vulnerability in the asset must be addressed • Contractual restrictions on patching (e.g., a highly regulated asset can only be patched by its manufacturer after testing and certification) • Mission/business restrictions on risk responses for that asset (e.g., an asset can only be rebooted during a monthly maintenance outage)	
	Process	Organizations should establish and constantly maintain up-to-date software inventories for their physical and virtual computing assets, including OT, IoT, and container assets. This information could be in a single enterprise asset inventory, or it could be split among multiple resources. While a comprehensive inventory of all assets is ideal, it may be impossible to achieve, given the highly dynamic nature of assets and software. A realistic goal is to maintain a close-to-comprehensive inventory by relying on automation to constantly discover new assets and collect up-to-date information on all assets.	(Souppaya & Scarfone, 2022)
		Without constant updates, inventories will quickly become outdated and provide increasingly inaccurate and incomplete information for patching efforts. At one time, when assets and software were mostly static and were located within static logical and physical perimeters, it was generally considered acceptable to update inventories on a monthly or quarterly basis by performing a vulnerability scan. That model should no longer be used.	(Souppaya & Scarfone, 2022)
		Constantly updating inventories for all of the technologies and environments in use today requires a combination of automation techniques and tools. Organizations should leverage inventory capabilities built into platforms and assets whenever feasible. For example, APIs built into a cloud-based platform may enable continuous updates of inventory information for the software on that platform, as well as other platform characteristics helpful for patch management purposes. Vulnerability scans and passive network monitoring on local networks can still contribute to asset inventories, especially in terms of asset discovery. If vulnerability scans are to be used for software inventories, they will need sufficient access to the assets (i.e., authenticated scanning) in order to detect changes to their software and other technical characteristics.	(Souppaya & Scarfone, 2022)
		A configuration control system should be used to keep control of all implemented software as well as the system documentation.	(ISO/IEC, 2013) section 12.5.1
		Asset Inventory Management is another essential prerequisite for patch and vulnerability management. • Some form of organization-wide automated scanning is necessary to gather information about the installed program and binary files (e.g., for Microsoft Windows this includes .exe, .dll. and .ocx. files). • Several commercial and open source products provide this function including Microsoft System Center, IBM Tivoli, Secunia CSI, and OCS Inventory NG.	(Hoehl, 2013, p. 11) (ISO/IEC, 2013) section 12.6.1
		As the automated tools accumulate information about vulnerabilities, essentially a "recipe box" is being created to successfully and substantially exploit the organization's technology assets.	(SANS, Hoehl, 2013, p. 13)
		Section 3.3: Organizations should carefully consider all alternative host architectures in use for the enterprise when designing enterprise patch management policies and solutions	(Souppaya & Scarfone, 2013)
		Automated scanning MUST be deployed to report patch status on a regular basis, to correlate current patch status against vulnerabilities.	(DWP, 2021) Section 10.6.2
		Know when new software vulnerabilities affect your organization's assets, including applications, operating systems, and firmware.	(Souppaya & Scarfone, 2022)
	Strategy	Organizations should use the software inventories, technical and business/mission characteristics, and risk response scenarios to assign each asset to a maintenance group. A *maintenance group* is a set of assets with similar characteristics that generally have the same software maintenance needs for each risk response scenario. Maintenance needs include not only patching (e.g., patch schedule, patch testing needs, outage restrictions, level of impact if vulnerable software is compromised) but also any other appropriate forms of mitigation and risk response, such as temporary mitigations used	(Souppaya & Scarfone, 2022)

		when patches are not yet available. Organizations should define their maintenance groups at whatever they decide the best level of granularity is, then periodically reassess their maintenance group definitions and adjust them as needed.	
		Here are a few simplified examples of possible maintenance groups: • Mobile workforce laptops for standard end users • On-premises datacenter (including servers, network equipment, storage, etc.) • Legacy OT assets • Smartphones for the mobile workforce • On-premises servers for automated software testing • Containers with customer-facing applications in the public cloud Maintenance groups can also be defined based on other characteristics, like personnel roles (e.g., software developer workstations, system administrator workstations) or asset importance (e.g., low-impact IoT consumer assets, OT and IoT assets with life-safety impact).	(Souppaya & Scarfone, 2022)
Support	Software version	Vendor supplied software used in operational systems should be maintained at a level supported by the supplier. Over time, software vendors will cease to support older versions of software. The organization should consider the risks of relying on unsupported software.	(ISO/IEC, 2013) section 12.5.1
	Access control	Physical or logical access should only be given to suppliers for support purposes when necessary and with management approval. The supplier's activities should be monitored.	(ISO/IEC, 2013) section 12.5.1
		Computer software may rely on externally supplied software and modules, which should be monitored and controlled to avoid unauthorized changes, which could introduce security weaknesses.	(ISO/IEC, 2013) section 12.5.1
	Procurement	Organizations should take software maintenance into consideration when procuring software**.** Software maintenance is one factor of many that organizations should consider. It is beyond the scope of this publication to provide methodologies for estimating software maintenance costs or factoring software maintenance into procurement decisions.	(Souppaya & Scarfone, 2022)

A.3 Timeframes

Table A.17 - Overview of timeframe indications

	Guideline	Publication
Timeframe	Once a patch is released by a vendor, the patch should be applied in a timeframe commensurate with an organisation's exposure to the security vulnerability and the level of cyber threat the organisation is aiming to protect themselves against.	(ACSC, 2021)
	The following are recommended timeframes for applying patches for applications: To mitigate basic cyber threats: • internet-facing services: within two weeks, or within 48 hours if an exploit exists • commonly-targeted applications: within one month To mitigate moderate cyber threats: • internet-facing services: within two weeks, or within 48 hours if an exploit exists • commonly-targeted applications: within two weeks • other applications: within one month To mitigate advanced cyber threats: • internet-facing services: within two weeks, or within 48 hours if an exploit exists • commonly-targeted applications: within two weeks, or within 48 hours if an exploit exists • other applications: within one month.	(ACSC, 2021)
	The following are recommended timeframes for applying patches for operating systems: To mitigate basic cyber threats: • internet-facing services: within two weeks, or within 48 hours if an exploit exists • workstations, servers, network devices and other network-connected devices: within one month To mitigate moderate cyber threats: • internet-facing services: within two weeks, or within 48 hours if an exploit exists • workstations, servers, network devices and other network-connected devices: within two weeks To mitigate advanced cyber threats: • internet-facing services: within two weeks, or within 48 hours if an exploit exists • workstations, servers, network devices and other network-connected devices: within two weeks, or within 48 hours if an exploit exists.	(ACSC, 2021)
	[Patches for vulnerabilities with a high probability of exploit and a high level of potential damage need to be deployed as soon as possible, preferably within one week. Less critical patches need to be planned for the first upcoming maintenance window.]	(BIR, 2014) 12.6
	Approved patches MUST be applied across the enterprise in a timeframe based on their criticality (defined in the risk assessment).	(DWP, 2021) Section 10.4.5.
	Where a security patch fixes a vulnerability that the CVSS or CWE score defines as 'critical', applications, systems and devices must be patched within 14 days of an update being released. Where a security patch fixes a vulnerability that the CVSS or CWE score defines as 'high', applications, systems and devices must be patched within 30 days of an update being released.	(DWP, 2021) Section 10.2.3
	In exceptional circumstances, a patch may need to be implemented faster than those requirements outlined at para 10.2.3. That advice will be based on an assessment of the threat and the vulnerability in question. In such circumstances, the response MUST be treated under SS014 - Security Incident Management as an Emergency Patch.	(DWP, 2021) Section 10.2.5

	Organizations should offer flexibility with how soon routine patches are to be installed, while also forcing installation after a grace period has ended. A routine patch does not necessitate immediate installation, but at some point, patches must be installed to reduce the risk for the entire environment. Forcing installation can be direct, like triggering patch execution, or indirect, like preventing network access for unpatched assets until they are patched.	(Souppaya & Scarfone, 2022)
Emergency patch	Organizations should consider using the same general approach for emergency patching as for routine patching, except with a highly accelerated schedule	(Souppaya & Scarfone, 2022)
	Organizations should plan for the quick implementation of multiple types of emergency mitigations to protect vulnerable assets. Mitigations may require deactivating system functionality or isolating an asset from other assets and having automated mechanisms to apply these changes. Without the processes, procedures, and tools in place to implement mitigations, too much time may be lost, and vulnerable assets may be compromised.	(Souppaya & Scarfone, 2022)
	Organizations should plan to replace emergency mitigations with permanent fixes. Once a permanent fix, such as a patch, is available, the patch will need to be deployed and the mitigation removed. Schedules should be set and enforced for both patch deployment and mitigation removal.	(Souppaya & Scarfone, 2022)

A.4 Patch Information retrieval

Table A.18 - Overview of information retrieval aspects

	Guideline	Publication
Information retrieval technique	Section 4.1: Organizations should carefully consider the advantages and disadvantages of each technique for identifying missing patches (e.g., agent-based, agentless scanning, passive network monitoring) when selecting enterprise patch management technologies.	(Souppaya & Scarfone, 2013)
Timeliness of information retrieval	Information about technical vulnerabilities of information systems being used should be obtained in a timely fashion, the organization's exposure to such vulnerabilities evaluated and appropriate measures taken to address the associated risk.	(ISO/IEC, 2013) section 12.6.1
	A timeline should be defined to react to notifications of potentially relevant technical vulnerabilities.	(ISO/IEC, 2013) section 12.6.1
	Threat intelligence feeds that detail system vulnerabilities MUST be collected at least weekly and reviewed from known, trusted third parties. These MUST be analysed and processed by a dedicated team and distributed to a relevant Triage Team.	(DWP, 2021) Section 10.2.1
	Distribute vulnerability and remediation information to local administrators.	(Mell et al., 2005)
Information source	Information resources that will be used to identify relevant technical vulnerabilities and to maintain awareness about them should be identified for software and other technology (based on the asset inventory list, see 8.1.1); these information resources should be updated based on changes in the inventory or when other new or useful resources are found.	(ISO/IEC, 2013) section 12.6.1
	Monitor security sources for vulnerability announcements, patch and non-patch methods of remediation, and emerging threats that match up with the software within the system inventory of the PVG.	(Mell et al., 2005)
	For example, your organization might subscribe to vulnerability feeds from software vendors, security researchers, and the National Vulnerability Database (NVD).	(Souppaya & Scarfone, 2022)
Information integrity	If a patch is obtained via manual download, the source and integrity of the package must be confirmed prior to its deployment.	(TBS, 2022)
	Validate the patch. A patch's authenticity and integrity should be confirmed, preferably by automated means, before the patch is tested or installed. The patch could have been acquired from a rogue source or tampered with in transit or after acquisition.	(Souppaya & Scarfone, 2022)

A.5 Vulnerability assessment, and prioritisation

Table A.19 - Overview of assessment and prioritization aspects

	Guideline	Publication
Risk assessment	Section 3.1: If a vulnerability is not being exploited yet, organizations should carefully weigh the security risks of not patching with the operational risks of patching without performing thorough testing first.	(Souppaya & Scarfone, 2013)
	Once a potential technical vulnerability has been identified, the organization should identify the associated risks and the actions to be taken; such action could involve patching of vulnerable systems or applying other controls;	(ISO/IEC, 2013) section 12.6.1
	If a patch is available from a legitimate source, the risks associated with installing the patch should be assessed (the risks posed by the vulnerability should be compared with the risk of installing the patch);	(ISO/IEC, 2013) section 12.6.1
	In case a patch is available, all risks tied to the installation of that patch need to be evaluated (the risk of the vulnerability needs to be compared to the risk of installing the patch)	(BIR, 2014) 12.6
	Any decision to upgrade to a new release should take into account the business requirements for the change and the security of the release, e.g. the introduction of new information security functionality or the number and severity of information security problems affecting this version. Software patches should be applied when they can help to remove or reduce information security weaknesses (see 12.6).	(ISO/IEC, 2013) section 12.5.1
	Patches MUST have a defined criticality using the current CVSS or, where applicable, CWE scoring calculation.	(DWP, 2021) Section 10.2.2
	Lastly, reporting on the current risk condition can be the biggest challenge. Presenting the impact of a vulnerability in factual, quantitative terms is an effective approach to compel management for action.	(Hoehl, 2013, p. 4)
	Several helpful methodologies for risk assessment exist including Factor Analysis of Information Risk (FAIR, 2007), NIST SP800-30 Guide for Conducting Risk Assessments (NIST, 2012), and OCTAVE (SEI, 2001).	(Hoehl, 2013, p. 4)
	Where appropriate, accountable parties MUST patch systems and end points based on their criticality.	(DWP, 2021) Section 10.5.3.
	Patching should be conducted as standard but where a risk to service delivery is identified, a risk assessment is required that considers the risk of: • Not deploying the patch • The risk of implementing the patch (i.e. destabilising a system or business process). • The availability or lack of compensating security controls that may impact the CVSS score. This must be delivered through a 'vulnerability triage group', consisting of staff with knowledge of cyber security risk, business risk and IT estate management. Where a decision is made **not** to fix the issue but to acknowledge it, a timeframe for reviewing this decision needs to be made, which should be no more than 3 months. This decision MUST be made by a suitable responsible person within the accountable business area.	(DWP, 2021) Section 10.3.2
	Where applicable, threat Intelligence functions may support the risk owner by providing advice on what mitigating actions can be taken to minimise the threat from zero-day exploits that do not have a patch available.	(DWP, 2021) Section 10.2.5
Risk prioritization	Prioritize the patch. A patch may be a higher priority to deploy than others because its deployment would reduce cybersecurity risk more than other patches would. Another patch may be a lower priority because it addresses a low-risk vulnerability on a small number of low-importance assets.	(Souppaya & Scarfone, 2022)
	Systems at high risk should be addressed first;	(ISO/IEC, 2013) section 12.6.1
	A single critical patch with no known attempts to exploit might be of lower risk than 10 less severe patches with known "exploits in the wild". Location of the asset also has an influence on risk condition. A laptop directly connected to the Internet might be of greater immediate risk than a server with the same vulnerability located within a firewall segmentation and intrusion prevention system.	(Hoehl, 2013, p. 4)

	In situations where resources are constrained, organisations are encouraged to prioritise the deployment of patches. For example, patches should first be applied for all internet-facing services. This should then be followed by important network devices, servers and workstations of high-risk users (e.g. senior managers and their staff; system administrators; and staff members from human resources, sales, marketing, finance and legal areas). Finally, all other network devices, servers and workstations should be patched.	(ACSC, 2021)
	Prioritize the order in which the organization addresses the remediation of vulnerabilities, based on analysis of risks to systems.	(Mell et al., 2005)
Processes and procedures	depending on how urgently a technical vulnerability needs to be addressed, the action taken should be carried out according to the controls related to change management (see 12.1.2) or by following information security incident response procedures (see 16.1.5);	(ISO/IEC, 2013) section 12.6.1
	an effective technical vulnerability management process should be aligned with incident management activities, to communicate data on vulnerabilities to the incident response function and provide technical procedures to be carried out should an incident occur;	(ISO/IEC, 2013) section 12.6.1
	Section 3.4.4: Organizations should use other methods of confirming installation, such as a vulnerability scanner that is independent from the patch management system.	(Souppaya & Scarfone, 2013)
	Section 3.4.5: Organizations using application whitelisting technologies should ensure that they are configured to avoid problems with updates.	(Souppaya & Scarfone, 2013)
	Effective organizations link their vulnerability scanners with problem-ticketing systems that automatically monitor and report progress on fixing problems and that make visible unmitigated critical vulnerabilities to higher levels of management to ensure the problems are solved.	(CIS, 2009)
	[It is recommended that earlier scan reports are stored to help identify differences. Minimally, this needs to be done for the most important systems. All changes in systems (open network ports, added services) need to be investigated.]	(BIR, 2014) 12.6
	Section 3.1: Organizations should carefully consider the relevant issues related to timing, prioritization, and testing when planning and executing their enterprise patch management processes.	(Souppaya & Scarfone, 2013)
	Define a procedure to address the situation where a vulnerability has been identified but there is no suitable countermeasure. In this situation, the organization should evaluate risks relating to the known vulnerability and define appropriate detective and corrective actions.	(ISO/IEC, 2013) section 12.6.1
	Plan the risk response. This involves assessing the risk the vulnerability poses to your organization, choosing which form of risk response (or combination of forms) to use, and deciding how to implement the risk response. For example, you might determine that risk is elevated because the vulnerability is present in many organization assets and is being exploited in the wild, then choose mitigation as the risk response and mitigate the vulnerability by upgrading the vulnerable software and altering the software's configuration settings.	(Souppaya & Scarfone, 2022)
	Prepare the risk response. This encompasses any preparatory activities, such as acquiring, validating, and testing patches for the vulnerable software; deploying additional security controls to safeguard the vulnerable software; or acquiring a replacement for a legacy asset that cannot be patched. It might also include scheduling the risk response and coordinating deployment plans with enterprise change management, business units, and others.	(Souppaya & Scarfone, 2022)

A.6 Testing

Table A.20 - Overview of testing aspects

	Guideline	Publication
Risk analysis of patch deployment	Section 3.2: Organizations should identify all the ways in which patches could be applied and act to resolve any conflicts among patch application methods.	(Souppaya & Scarfone, 2013)
	All systems requiring vendor, or other authorised patches MUST be assessed	(DWP, 2021) Section 10.3.1
	Section 3.4.3: Organizations should be capable of detecting side effects, such as changes to security configuration settings, caused by patch installation.	(Souppaya & Scarfone, 2013)
	Patches should be tested and evaluated before they are installed to ensure they are effective and do not result in side effects that cannot be tolerated; if no patch is available, other controls should be considered, such as: 1) turning off services or capabilities related to the vulnerability; 2) adapting or adding access controls, e.g. firewalls, at network borders (see 13.1); 3) increased monitoring to detect actual attacks; 4) raising awareness of the vulnerability;	(ISO/IEC, 2013) section 12.6.1
	Vendors are often under significant pressure to release patches as soon as possible. Therefore, there is a possibility that a patch does not address the problem adequately and has negative side effects. Also, in some cases, uninstalling a patch cannot be easily achieved once the patch has been applied.	(ISO/IEC, 2013) section 12.6.1
	Test the patch. A patch may be tested before deployment. This is intended to reduce operational risk by identifying problems with a patch before placing it into production. Testing may be performed manually or through automated methods.	(Souppaya & Scarfone, 2022)
Testing facets	Applications and operating system software should only be implemented after extensive and successful testing; the tests should cover usability, security, effects on other systems and user-friendliness and should be carried out on separate systems (see 12.1.4); it should be ensured that all corresponding program source libraries have been updated	(ISO/IEC, 2013) section 12.5.1
Test environment	All patches MUST be tested in a suitable environment (that meets live conditions) prior to being applied to the enterprise. This is also applicable to immutable infrastructure, which goes through a development environment and Continuous Integration pipelines.	(DWP, 2021) Section 10.4.1.
Approach	In some cases, the security updates do not include a back-out or uninstall option. Proper testing and phased implementation are the best methods for early detection of problems introduced by the new code.	(Hoehl, 2013, p. 8)
	The patch becomes approved once testing has been concluded satisfactorily.	(DWP, 2021) Section 10.4.3.
	Delivery of the approved patch across the estate MUST be in stages to reduce impact. The 'Blue/Green' deployment model can also be utilised to achieve this.	(DWP, 2021) Section 10.4.4.
	If adequate testing of the patches is not possible, e.g. because of costs or lack of resources, a delay in patching can be considered to evaluate the associated risks, based on the experience reported by other users. The use of ISO/IEC 27031[14] can be beneficial.	(ISO/IEC, 2013) section 12.6.1
	Conduct the testing of patches and non-patch remediation methods on IT devices that use standardized configurations.	(Mell et al., 2005)
Documentation	A record of the decision to apply or reject manual patches, MUST be documented within the Risk Register defined by the Risk Assessment process. For automated patching via upgrades, a record of the reason for rejecting a product update must be maintained and reviewed on a regular basis, along with a risk assessment.	(DWP, 2021) Section 10.3.3
	Accountable parties MUST test the patch to check for compatibility, and create a back-up or restore point. This detail must be documented on the Risk Register and CMDB.	(DWP, 2021) Section 10.4.2.
	Where testing is not feasible, this MUST be risk assessed and recorded on the Risk Register.	(DWP, 2021) Section 10.4.6.

A.7 Deployment

Table A.21 - Overview of deployment aspects

	Guideline	**Publication**
Deployment	Where appropriate, accountable parties MUST ensure all patching is applied across the enterprise where necessary. This includes applying application updates or upgrades that include security updates.	(DWP, 2021) Section 10.5.4.
	Patches that only deliver functional change and do not fix a vulnerability MUST NOT be delivered as security patches.	(DWP, 2021) Section 10.1.6
	Entitlement to patch manually MUST be confirmed before applying a patch e.g. open source products that do not have a support package or service wrapper.	(DWP, 2021) Section 10.3.4
	Where possible, accountable parties MUST automate patch deployment across end points. Immutable infrastructure is kept up to date continuously, via updates or upgrades, which achieve the same purpose as patching.	(DWP, 2021) Section 10.5.1
	Any manually applied patches found to have bypassed control mechanisms for installation MUST be subject to a formal review and uninstallation if deemed necessary.	(DWP, 2021) Section 10.1.3
	Patching should be automated wherever possible, and should utilise dedicated service accounts with elevated privileges where appropriate. Manual patching MUST only be conducted by users with enhanced access and/or privileged users.	(DWP, 2021) Section 10.1.1
	[For software services of technical infrastructures it needs to be checked if the latest updates have been deployed, preferably automatically. The deployment of a patch is not automatically, unless special arrangement are made with the software vendor.]	(BIR, 2014) 12.6
	Perform automated deployment of patches to IT devices using enterprise patch management tools. Configure automatic updates of applications whenever possible and appropriate.	(Mell et al., 2005)
	Widespread manual patching is no longer effective for risk and resource management as the number of patches necessary for vulnerabilities grows and threats continue to rise.	(Hoehl, 2013, p. 13)
	Patch deployment varies widely based on several factors, including: • The type of software being updated (e.g., firmware, operating system [OS], application) • The asset platform type (e.g., IT, OT, IoT, mobile, cloud, virtual machine [VM], containers) • Platform traits, such as managed/unmanaged asset, on-premises or not, virtualized or not, and containerized or not • Environmental limitations, such as network connectivity and bandwidth	(Souppaya & Scarfone, 2022)
	Distribute the patch. Distributing the patch to the assets that need to have it installed can be organization-controlled (and occur automatically, manually, or as scheduled) or vendor-controlled, such as delivered from the cloud.	(Souppaya & Scarfone, 2022)
	Install the patch. Installation can occur in numerous ways, including automatically; manually when directed to do so by a user, administrator, vendor, or tool; as a result of other software being installed or updated; and through the replacement of removable media used by an asset. Some installations require administrator privileges, such as installing firmware patches for a system basic input/output system (BIOS).3 Some patch installations require user participation or cooperation.	(Souppaya & Scarfone, 2022)
	Change software configuration and state. In some cases, making a patch take effect necessitates implementing changes. Examples include restarting patched software, rebooting the operating system or platform on which the patched software runs, redeploying the applications, or altering software configuration settings. In other cases, no such changes are needed.	(Souppaya & Scarfone, 2022)
	Organizations should consider adopting phased deployments for routine patching in which a small subset of the assets to be patched receive the patch first. These assets act as canaries (i.e., bellwethers) for identifying issues and determining the likely operational impact of the patch. In effect, this is how the patching gets tested. If the canary assets indicate that the patch should have minimal impact, the deployment can expand to more or all of the vulnerable assets. Significant problems can be addressed before the rollout expands, or a different risk response –	(Souppaya & Scarfone, 2022)

		like a temporary mitigation – can be planned instead of the patch while the problems are resolved.	
Resource allocation		Section 3.4.2: Organizations should ensure that their enterprise patch management can avoid resource overload situations.	(Souppaya & Scarfone, 2013)
Roles and responsibilities	Administrators	Section 4.3: A patch management technology's administrators should design a solution architecture, perform testing, deploy and secure the solution, and maintain its operations and security	(Souppaya & Scarfone, 2013)
		The updating of the operational software, applications and program libraries should only be performed by trained administrators upon appropriate management authorization (see 9.4.5)	(ISO/IEC, 2013) section 12.5.1
	End-users	Standard business users MUST NOT have the ability to install unauthorised patches on any departmental end points.	(DWP, 2021) Section 10.1.2
		Section 3.2: Organizations should ensure that users cannot disable or otherwise negatively affect enterprise patch management technologies, and organizations should perform continuous monitoring of enterprise patch management technologies to identify any issues that occur	(Souppaya & Scarfone, 2013)
Code		Operational systems should only hold approved executable code and not development code or compilers;	(ISO/IEC, 2013) section 12.5.1
Documentation		All patches, both manual and automated, MUST be recorded. For automated patching via upgrades, a record of the reason for rejecting a product update must be maintained and reviewed on a regular basis, along with a risk assessment.	(DWP, 2021) Section 10.5.2.
		An audit log should be maintained of all updates to operational program libraries;	(ISO/IEC, 2013) section 12.5.1
		an audit log should be kept for all procedures undertaken;	(ISO/IEC, 2013) section 12.6.1
		An audit of patch deployment success and failure rates should be performed after each deployment to identify outliers, and to trace and correct patch installation failures. The asset inventory and discovery features of an enterprise patch management suite will reflect the status of the deployment and provide the statistics to satisfy KPI and Service Level Agreement requirements defined in the organization's strategy.	(TBS, 2022)
		Patches that have not been implemented MUST be reported to the system and risk owner who remains responsible for the exposure caused by the inability to patch.	(DWP, 2021) Section 10.6.3.
		When patches have been deployed, reporting MUST be run to confirm their deployment.	(DWP, 2021) Section 10.6.1.
		Create a database of remediation methods that need to be applied within the organization. Oversee the vulnerability remediation process in the organization.	(Mell et al., 2005)
		Along with having a high-level timeline of when patching, testing, and notifications take place, it is also be important to define the granular details of which systems are patched during what interval. The purpose of this documentation is to outline required patching steps, assign responsibilities, identify system hierarchy and system dependencies, minimize downtime, and provide notification to supporting teams. Having this documented will help minimize confusion and enable specific groups to focus on key priorities.	(Ruppert, 2008, pp. 34-35)

A.8 Post-Deployment

Table A.22 - Overview of post-deployment aspects

	Guideline	Publication
Rollback strategy	A rollback strategy should be in place before changes are implemented;	(ISO/IEC, 2013) section 12.5.1
	Resolve any issues. Installing a patch may cause side effects to occur, like inadvertently altering existing security configuration settings or adding new settings, and these side effects can inadvertently create a new security problem while fixing the original one. Patch installation can also cause operational issues that may necessitate uninstalling the patch, reverting to the previous version of the software, or restoring the software or asset from backups.	(Souppaya & Scarfone, 2022)
Archiving	Previous versions of application software should be retained as a contingency measure;	(ISO/IEC, 2013) section 12.5.1
	Old versions of software should be archived, together with all required information and parameters	(ISO/IEC, 2013) section 12.5.1
Monitoring/ evaluation	The technical vulnerability management process should be regularly monitored and evaluated in order to ensure its effectiveness and efficiency;	(ISO/IEC, 2013) section 12.6.1
	Measure the effectiveness of the patch and vulnerability management program in a consistent manner and apply corrective actions as necessary.	(Mell et al., 2005)
	Verify vulnerability remediation through network and host vulnerability scanning.	(Mell et al., 2005)
	An audit of patch deployment success and failure rates should be performed after each deployment to identify outliers, and to trace and correct patch installation failures. The asset inventory and discovery features of an enterprise patch management suite will reflect the status of the deployment and provide the statistics to satisfy KPI and Service Level Agreement requirements defined in the organization’s strategy.	(TBS, 2022)
	Verify the risk response. This step involves ensuring that the implementation has been completed successfully. For patching, this means confirming that the patch is installed and has taken effect. For deploying additional security controls, ensure they are functioning as intended. For risk avoidance, verify that vulnerable assets were decommissioned or replaced.	(Souppaya & Scarfone, 2022)
	Continuously monitor the risk response. Make sure that the risk response continues to be in place: no one uninstalls the patch, deactivates the additional security controls, lets the cybersecurity insurance lapse, or restarts the decommissioned asset.	(Souppaya & Scarfone, 2022)
	In the last phase of the life cycle, the patch’s deployment can be monitored using automation to confirm that the patch is still installed. For example, monitoring could confirm that the patch has not been uninstalled by a user or an attacker, an unpatched version of the software has not been restored from a backup, and the device has not been reset to a vulnerable factory-default state.	(Souppaya & Scarfone, 2022)
	Another reason for monitoring the deployed patches is to see if the patched software’s behavior changes after patching. As part of a layered security approach to mitigating supply chain risk, this might be helpful at detecting, responding to, and recovering from situations where the installed patch was itself compromised.	(Souppaya & Scarfone, 2022)

Appendix B – Interview plan

This appendix provides an overview of the **initial set of questions used during the interviews**, as discussion in the synthesis in section 4.1.1.

Function

- What is your function and what is it that you do on a daily basis?
- What is your role within the organization's IT department? Team? System?
- What decisions need to be made regarding security?
- Who else is involved in the team, how are the roles divided, with who do you interact frequently?
- Is it clear who is responsible for what within the team/department, and who has what role?
- How is the communication? Who is involved in the decision making of tasks?

Security

- How is security managed in your department and what is your role in security? What is it that you maintain?
- What needs to be done to make the system secure?
- What does the security of this system mean for the security of the entire organization?
- What is on the systems? what is the nature of the systems you patch?
- What part of security is involved with patching?
- What is it that you patch? what are you patching for, what are you patching against?
- What of the security processes is documented? And in what form (shared, personal notes, etc.)?
- How does the procedure of an emergency patch go?
- How much/often are involved with patching in a day or a week?
- Within your department/systems, how often are patches released?
- Would you say there are delays within the process of your department, and if so, what impact does this have?

What makes a patch happen/ why patching is needed

- Can you describe the last 10 patches you deployed?
  - What makes a patch happen, why is patching needed?
  - Where these easy, difficult, could the process of these be improved?
  - What was the nature of these?
- Does patching take longer than you think it should?
- Do you feel patching is done well, or are there any aspects in or out of your control which could be improved in any way?
- Can you briefly explain the beforementioned patching process and what it looks like, and which steps need to be taken (difference between routine (if any exists) and ad hoc)?
- What procedures are there within the organization regarding patching? Are these based on international standards or on company rules? Are there KPI’s within the team of the effectiveness/efficiency of patching?
- Are any tools used to track and assess the performance of the team?
- Are there guidelines regarding the duration of the steps in the patch process (e.g., release/deployment time)?
- Are there organizational rules, guidelines that limit the patching process (e.g., max server downtime)?

Effectiveness and difficulties

- What are the main trade-offs being made in patching?
- How do you think the patching process could be more effective?
- What do you think are the difficulties or challenges of patching, why are there often delays?
- How do you measure the effectiveness of the process?
    - And the success, as in how 'well' your systems are patched?
- Do you have a practical example of something that went wrong? And how did you act on this, how did you solve it?

Specified components

- Where do you get the information about available patches/ vulnerabilities and where does it come in within the department?
    - how often/ frequency?
- Who assesses whether a patch should be rolled out?
    - Does this need to be approved through multiple teams/levels/levels?
- Is there any form of prioritization, and if so, what criteria do you use to prioritize?
- How do you perform testing? (patching cycles compared to ad hoc)
- What do you do after deployment, with regard to maintenance, monitoring?

Appendix C – Overview of interplay of challenges

This appendix provides an **overview of the interplay of challenges and coping strategies**, as discussion in the synthesis in section 6.2

C.1 Organisational challenges (CO)

CO1 – Developments of threat environment

There is a decreasing timeframe between the moment a vulnerability is known to exist, and the moment threat actors start to exploit this vulnerability (Eriksen-Jensen, 2013). Furthermore, there is a high level of unpredictability of when a vulnerability will be exploited. These factors shape the decision space of the timeframe to implement a patch from a security aspect. The significance on the decision to patch quicker is moderate, due to the constraints and barriers that make it troublesome to do so in practice. Furthermore, being in a state of emergency results in the coping strategy of a higher level of cooperation (ES1) and having people on stand-by in weekend and off-work hours (ES2).

CO2 – Behaviour and awareness of system-owners

The high dependency of the behaviour and awareness of system-owners regarding patching is forming a barrier that hinder the decision space from both a security and operational perspective. The lack of threat perception and the lack of perception that patching reduces this threat hinder IT practitioners to convince system-owners to patch their systems. Additionally, a lack of knowledge of how to patch and a lack of prioritization of system-owners hinders IT practitioners to assure that these systems are patched in a timely manner. Because the decision-making process and deployment of patches lies with the system-owners, the significance of these factors is high. Despite the coping strategy of providing pressing advice and guidance on how and why to patch (S3). The total number of systems tied to the network can only be patched entirely if the systems owned by others are patched too.

CO3 – Human capability of IT practitioners

The practice of patching remains to involve human knowledge and expertise. The possibility for human error during the deployment of patches shapes the decision space of the timeframe of patching from an operational aspect of patching. Human error can cause a delay in patch deployment, wherefore the moment of patching can be postponed. The possibility for human error during patch assessment shapes the decision space of the applicability aspect of a patch. If a wrong assessment has been made in the applicability of a patch, a patch is not rolled out even if this might have been a crucial patch. Although both of these factors play a role of influence, their significance is low as the occurrence is low as well. Furthermore, the coping strategy (S4) of letting the assessment be done by multiple people from different teams reduces the likelihood of a wrong assessment. Interesting to notice here, is that the coping strategy for one challenge, can be increasing the significance of another challenge. Involving multiple people increases the need for coordination (CP8).

The need to balance patching and tasks of the day-to-day job of IT practitioners limits the decision space of the timing of patching from an operational aspect. Where patching is not prioritised over other tasks, the timing is restricted to those moments an IT practitioner is not working on other tasks. Though this factor plays a less significant role in the state of an emergency, where it is shown that other tasks are postponed and patching has priority (ES5), therefore the significance of this factor is moderate.

CO4 – Human resources

A factor of high significance is the lack of human capacity that limits the decision space for the timing of patching from an operational aspect. Where individual IT practitioners need to find a balance, the total number of IT practitioners limit the timing decision more substantially. As discussed in the section 4.3.2 the organisational tension between money and resources is currently accepted.

CO5 – Organisational structure

The structure of the organisation shapes the decision space what to patch and when to patch. The decentralisation of the IT environment increases the challenge of knowing the applicability of a patch. Simultaneously, the decentralisation of the IT environment increases the challenge of the timing decision of patching, as can be concluded from the behaviour and awareness of system-owners (CO2). Furthermore, the need to balance security practices and organisational objectives limits the timing decision space from both an availability and security aspect. This tension and the coping strategies used are extensively discussed in section 4.3.2, where it is concluded that a misalignment between what IT practitioners prioritize and what top-level management prioritize has a high level of significance. The significance of this challenge on the decision space is high.

C.2 Procedural challenges (CP)

CP6 – Collaboration

Security patch management is a collaborative practice, where dependencies of different internal and external stakeholders form barriers in the decision space of when to patch. The existence of dependencies identified in section 3.2 is confirmed and amplified by the challenges identified from the qualitative data. First, inter-dependencies are present between different teams within the IT department. There are dependencies between different operational teams (e.g. systems, infra, applications, and data management) that are mostly existent because of system dependencies (CT12). These dependencies hinder the timing of patching from both an operational and applicability perspective. If a systems application patch is dependent on a certain level of OS patch, while that OS patch is not in place, the dependency of another team hinders the timeliness of patching the application.

Second, there are dependencies of external stakeholders (e.g. end-users) to determine the moment of patch events for the central IT services. These hinder the timing of patch deployment from an availability aspect. These factors are therefore influenced by the need to balance security practices and organisational objectives (CO5) and the organisational restrictions of available patch moments (CP9). The coping strategy of equipping systems with High Availability (HA) (S9) helps to deal with this. Third, as discussed with CO2, the behaviour and awareness of system-owners is a significant dependency that increases the need to collaborate effectively from a security aspect. The need for certainty about responsibilities and accountability of the IT department and system owners forms a barrier for the timing of patching from an operational aspect. This challenge has a positive relationship with the challenge of behaviour and awareness of system-owners (CO2). If there is more certainty about responsibility and accountability, the behaviour and awareness of system-owners increases towards more timely patching. This dependency is moderated by allocating the decision-making rights to the IT department in case of emergency (ES7).

Lastly, the practice of decision-making through multiple levels of the organisations hinders the timeliness of patching from an operational aspect. This is dealt with by informal decision-making on the go (S8). Interestingly, this factor limits the significance of another challenge (CO3), the possibility for human errors to occur. Though this factor plays a less big role in the state of an emergency, where it is shown that collaboration is done more quickly due to an organisation-wide emergency change process (ES7), therefore the significance of this factor is moderate. The overall challenge of collaboration is of high significance as it impacts the state of other challenges (i.e., CO2, CO5).

CP7 – Communication

The collaboration challenge (CP6) brings the existence of another challenge, that of communication. Because of the inter-dependencies, communication regarding the time of patch event hinders the timeliness of patching from an operational aspect. Organisation-wide procedural guidelines and processes (S10), however, these are highly distributed and are not that suitable. Another factor part of the communication challenge is the dependency of external stakeholders as information source for vulnerability notifications and patch criticality levels. This factor influences both the timing of patching as well as the decision to patch at all, from a security aspect. This is slightly managed by using multiple channels of information (S11). The significance is high, as the availability of information shapes the outcome of the decision-making process.

CP8 – Coordination

The collaboration challenge (CP6) discusses the need for certainty about responsibility and accountability for IT department and system-owners. Similarly, coordination within the IT department needs to establish responsibilities and authority of decision-making. The lack of having this clear hinders the timeliness of the patching decision, from both an operational and a security aspect. Furthermore, lack of clarity in responsibilities and authority of decision making of the IT department within the entire organisation hinders the timeframe of patching, from an availability and security aspect. This is linked to the tension between security and availability discussed in the challenge of the organisational structure (CO5). This indicates that establishing clear roles and responsibilities, therefore reducing the challenge of coordination, could lead to limiting the challenges of a decentralised organisational structure and the need to balance security practices and organisational objectives. Although some organisation-wide procedural guidelines and processes (S10) are used, the significance of this factor is still high.

Another coordination challenge is the lack of a centrally arranged patch information retrieval process. This shapes the space of the patch decision from an operational aspect. When information retrieval is scattered throughout the IT department, the challenge of collaboration (CP6) and the challenge of communication (CP7) increases. This however does not influence the availability of information to become aware of the existence of vulnerabilities, as multiple IT practitioners from different teams are gathering information is parallel. This is partly managed by the coping strategy of informal decision-making on the go (S8). This challenge is therefore more targeted at the efficiency (i.e., timing) decision than the effectiveness (i.e., awareness of patch existence), wherefore the significance is moderate.

CP9 – Procedures and guidelines

Closely related to the challenge of coordination (CP8), there is a challenge to establish suitable procedures and guidelines (S10). For one, a lack of availability of information throughout the patching process shapes the timing decision from an operational perspective. When information is needed to collaborate, communicate, or coordinate, the absence of it makes it troublesome to determine the patch moment. For example, when there are known system dependencies (CT12), but it is not known who to call in which situation, efficiency gets lost.

Secondly, a lack of Key Performance Indicators (KPIs) to indicate the effectiveness of patching shapes the timing decision of patching from a security aspect. Often 'as fast as possible' is the main aim to deal with the challenging developments of the threat environment (C01). However, this is a utopia and unrealistic aim. Determining 'what's good enough' and what KPIs are relevant is a difficult, but much needed first step in using metrics as a way to keep track of how effective the patching process is. If no one in the organisation knows what the aim is, IT practitioners cannot be accounted for. When the aim is to go for 100% at all times for all systems, it does not contribute to a stimulant as it is known that that percentage is not reached anyway. This is therefore a factor of high significance.

Thirdly, closely related to factor of needing to balance security practices and organizational objectives (CO5), organizational restrictions of available patch moments (e.g. change weekends) limit the decision space of the patch moment from an availability aspect. However, as discussed in section 4.3.2 in a state of emergency the influence of this factor is reduced as emergency patches can be deployed outside existing change moments (E12).

Fourth, in line with the challenge of behaviour and awareness of system-owners (CO2) and the dependency of system owners to patch their own systems (CP6), lack of a defined process for patching these non-central IT systems shape the space of the patching decision from a security aspect. The significance is therefore high, as tackling this challenge is able to ease the other two challenges. Lastly, the lack of clarity and standardisation of an emergency-response procedure forms a barrier of the timeframe of patching from an operational aspect. This targets the challenge of collaboration (CP6), communication (CP7), and coordination (CP8).

C.3 Technical challenges (TC)

CT10 – Patch quality

As earlier identified in literature (Dissanayake et al., 2022), patch deployment can include side-effects that might harm a system's functioning. The presence of this shapes the patch decision, and the moment to patch from an operational aspect. The coping strategy to test in a test-environment (S13) or wait on confirmation of other organisations (S14) helps reduce the risk of this challenge, but at the same time clashes with the factor of the decreasing timeframe between vulnerability awareness and exploit by the threat actor (CO1). This indicates that while a coping strategy reduces the significance of one challenge, it can be in conflict with another challenge.

It is furthermore possible that the existence of one challenge causes another challenge to occur. Because of the decreasing timeframe between vulnerability awareness and exploit by the threat actor (CO1), vendors often release emergency patches without proper testing due to being in a hurry to fix a vulnerability. Moreover, this increases the possibility for occurrence of side-effects of patch deployment. This vicious circle of needing to patch quickly, but at the same time not too quickly because of potential side-effects because of the urgency to release a patch vendors experience indicates the interplay between different socio-technical factors and the challenges these entail.

CT11 – Patch availability

The absence of available patches shapes the patch decision from an operational aspect, simply because patching cannot be deployed. The significance of this challenge is high, and results in the need to make additional decisions. One of the decisions and a coping strategy is the possibility to take other mitigating measures (S15) such as isolation or shutdown of a system. However, this then impacts other challenges such as collaboration (CP6), communication (CP7), and coordination (CP8). This indicates that a coping strategy can reduce the significance of one challenge, it can increase the need to address other challenges.

CT12 – System dependencies

The challenge of collaboration (CP6) is caused by the existence of system dependencies. There are two main factors involved; the lack of knowledge of system dependencies shapes the decision to patch and the moment of patching from an applicability perspective, and the presence of known dependencies between applications and databases of servers and networks hinders the space of the patch decision and timing of patching from an operational aspect. Especially when these dependencies cause the patches to be deployed in sequence, the timeframe of this deployment is impacted. Furthermore, no clear overview of dependencies increases the challenge of communication (CP7).

CT13 – Hardware resources

Similar to human resources (CO4), hardware resources form a constraint for the timing of patching from an operational aspect. To deal with the vast number of patch releases, patch filtering on relevance is used (S16) to reduce the burden on hardware resources. This relates to tension 3, where financial capacity increases the challenge of doing patching with limited hardware resources. In this case, solving this challenge can ease another challenge, that of the possibility of side-effects during deployment (CT10). By having more hardware resources, suitable testing can be carried out by the organisation itself, wherefore the dependency on other stakeholders to confirm a patch decreases. Even further, more hardware resources limit the significance of the development of the threat environment (CO1), as testing can be done more quickly.

CT14 – Complexity of systems

Apart from system dependencies (CT12), the complexity of systems themselves is a challenge. A large number of unique servers that all need different patches applied manually shapes the timeframe of a patch decision from an operational aspect. In addition, the existence of a large number of legacy systems that is 'unpatchable' shapes the decision space of patching from an applicability aspect. Furthermore, lack of knowledge of how these systems function and the criticality of these systems shape the decision to patch from an applicability aspect as well. All of these factors are of high significance, as the impact on the decision is great and at the same time there is little room to address this challenge in a short time, other than accepting the situation as it is (S0).

CT15 – Automation tools

Literature suggests the usage of automation tools (Dissanayake et al., 2022) to ease certain other challenges as the need for human resources (CO4) and the need for collaboration (CP6). This shapes the decision space of the timing of a patch decision from an operational aspect. The significance of this factor is however low, as the patching decision is not influenced by this factor as much. A factor of moderate significance is the need for a 'human-in-the-loop' to assess the applicability and relevance of patches to the organisation's system. This limits the decision of the timeframe from an operational aspect, as full automation of the patch process is not possible.

CT16 – Asset overview

A lack of a complete asset overview and complete information of the known assets (e.g., owner, patch status) shapes the decision space of both the decision to patch and the timeframe to patch, from an applicability aspect. When it is not known what assets are present, it cannot be known what patches are applicable to the systems. Similarly, not knowing the patch status of a system or the owner of a system makes it more troublesome to patch timely. Additionally, the dependency of other department (e.g., faculties) and their knowledge of who owns a certain system is a challenge that shapes the timing element of the patch decision from an applicability aspect. This is tied to the challenge the organisational structure (CO5) brings. Here, addressing another challenge, that of procedures and guidelines (CP9), can help to reduce the significance of this challenge.

Appendix D – Exemplary list of security patch metrics

This appendix included an **exemplary list of metrics** recommended by publications of standardisation and advisory bodies in Table D.23, as discussed in section 5.1

Table D.23 - Recommended metrics

Measure	Metric	Publication
Patch Program Maturity / Implementation	- Response time for accepting vulnerability notification - Response time for risk assessment - Response time for testing - Change Management violations - Change Management changes/reschedules - Patch packaging duration and level of effort - Patch infrastructure readiness to patch - Planned patches in place	(Hoehl, 2013, p. 15)
	- What percentage of the organization's desktops and laptops are being covered by the enterprise patch management technologies? - What percentage of the organization's servers have their applications automatically inventoried by the enterprise patch management technologies?	(Souppaya & Scarfone, 2013)
	- The percentage of systems and applications within the organization inventoried and covered by automated patch management	(TBS, 2022, par. 27)
Risk	- Number of Patches - Number of vulnerabilities - Number of Network Services	(Mell et al., 2005)
	- Susceptibility to attack - Duration of patch delivery - Number of exemptions - Planned patches not in place - New computers missing patches - Emergency patching - Unpatched, unauthorized software missing patches	(Hoehl, 2013, p. 15)
Efficiency and effectiveness	- Response Time for Vulnerability and Patch Identification - Response Time for Patch Application (critical/ noncritical) - Response Time for Emergency Configuration Changes	(Mell et al., 2005, pp. 3-3)
	- How often are hosts checked for missing updates? - How often are asset inventories for host applications updated? - What is the minimum/average/maximum time to apply patches to X% of hosts? - What percentage of the organization's desktops and laptops are patched within X days of patch release? Y days? Z days? (where X, Y, and Z are different values, such as 10, 20, and 30) - On average, what percentage of hosts are fully patched at any given time? Percentage of high impact hosts? Moderate impact? Low impact? - What percentage of patches are applied fully automatically, versus partially automatically, versus manually?	(Souppaya & Scarfone, 2013)
	- How often hosts are automatically checked for compliance - How often asset inventories are automatically updated - The minimum/average/max time to patch X percentage of hosts - The percentage of systems patched within X, Y, Z days after deployment - The percentage of operational hosts within the organization fully patched at any given time - The number of extreme impact, high impact, medium impact, low impact hosts and/or unpatched vulnerabilities on organizational hosts at any given time - Average time elapsed between a patch's availability and its production implementation per level of rating - The percentage of hosts patched automatically vs. partially (in the case of patches bundled in a package) vs. manually - The percentage of patches deployed within the suggested deployment schedule	(TBS, 2022, par. 27)
Cost	- Cost of PVG - Cost of tools - Cost of services - Cost of rework or redeployment	(Hoehl, 2013, p. 15)
	- Cost of the Patch and Vulnerability Group - Cost of System Administrator Support - Cost of Enterprise Patch and Vulnerability Management Tools - Cost of Program Failures	(Mell et al., 2005, pp. 3-4 - 3-5)
	- What cost savings has the organization achieved through its patch management processes? - What percentage of the agency's information system budget is devoted to patch management?	(Souppaya & Scarfone, 2013)

Appendix E – Evaluative questions to assess measurements

This appendix provides an overview of the **UCISA Information Security Management Toolkit** (UCISA, n.d.) in Table E.24

Table E.24 - Evaluative questions to assess measures

Is my measure ...	
RELIABLE?	Can it be consistently measured in a repeatable way? Is it reproducible? If NOT, are there known explanations of sources of uncertainty which are acceptable to all stakeholders (even if they dominate the values)?
SUSTAINABLE?	Is it cheap to gather? If it needs to be computed frequently, is the metric's source data cheap to gather? Can it be quickly evaluated? Are the costs of evaluation low enough that it is useful for those who will use it? Is solid data readily available?
MEASURABLE?	Can it be expressed as a cardinal number or percentage? Is there an accepted unit of measure? Can it be accurately measured? If NOT, is the distance between "true" and "real" measurement acceptable to stakeholders? Is it precise enough to be useful? Are measurements current enough to be useful, or time-stamped to a precision that makes them traceable?
OBJECTIVE?	Are measurements free of influence from the measurer's will or personal feeling? Is it unbiased? Can it be determined as being correct in an objective way? Is the process or system for collecting measurements correct according to its specification?
SCOPED?	Is it contextually specific? Is the domain in which it applies clearly defined? Conversely, does it overlap with other measurements? Is it meaningful to stakeholders, and does it reflect the meaning of what it is expected to be measuring? Is it relevant to stakeholders? Is it easy to interpret?
INSTRUMENTABLE?	Can it be automated through tool support? Is it sufficiently non-intrusive? Is the measurement process scalable? Is the measurement process portable to other environments? Can the measurement process, and environment being measured, be adequately controlled?
TRANSPARENT?	Can it be proved that it actually measures what it is supposed to? Can real evidence be gathered to demonstrate that it meets objectives? Is there an intended audience within or outside the organisation? Can the distance between the specified state ("should be"-state) and the real operational state ("as is"-state) be known? Is it objective, rather than subjective?
PROGRESSIVE? (Information Assurance)	Over time, can it demonstrate progression toward a goal? Is it possible to compare a measurement to previous measurements, targets, or benchmarks? Does it relate to a specific business goal? Are targets linked with achievable expectations? Are there stakeholders within the organisation capable of creating, using, and refining it?