Security Patch Management - An Overview of the Patching Process and its Challenges in Norwegian Businesses

JØRGEN BARLUND LEFDAL & DANIEL WINGER REISÆTER

SUPERVISOR

Marko Ilmari Niemimaa

University of Agder, 2022
Faculty of Social Science
Department of Information Systems

PREFACE

This thesis is the culmination of our time spent as students in the master's program in cybersecurity management at the University of Agder. It represents the knowledge and skills that we have acquired during that time.

Patching is something that we felt should be the subject of our thesis since we are of the mind-set that, if everyone were better at patching, the state of cybersecurity would be significantly improved. The objective of this thesis is to investigate the challenges associated with the patching process and produce potential answers to those difficulties, particularly regarding the management aspect of the topic.

We would like to express our gratitude to our supervisor, Marko Ilmari Niemimaa, Associate Professor in the Department of Information Systems at the University of Agder, for providing us with direction during the duration of this current semester. We would like to express our appreciation to everyone who took part in the interviews and provided us with new perspectives on the patching process.

Additional thanks go out to Jørgen's girlfriend Maja and Daniel's wife Camilla for providing assistance.

Kristiansand
June 2nd, 2022

Daniel W. Reis

Jørgen Lefdal

Daniel W. Reisæter

Jørgen B. Lefdal

ABSTRACT

Cyber-attacks are growing more frequent and sophisticated, and they are impacting businesses of all sizes. This encourages businesses to utilize safe, flaw-free systems, making them less susceptible to cyber-attacks. The issue is that no system is flawless, and a substantial number of security flaws are discovered regularly. To ensure the system's security, patches are distributed and implemented. Patches can be complicated and implementing them in systems can be difficult. This thesis seeks to identify the challenges that make the patching process challenging and to propose potential solutions.

This thesis was conducted utilizing a qualitative research strategy and methods such as a systematic literature review, to identify existing patching challenges identified by previous research.

We conducted interviews with business professionals who were familiar with the patching procedure and had understanding of cybersecurity. The majority of our interviewees were managers with additional expertise leading patching teams.

Prior study indicated various challenges in the field of patching and urged further investigation into the issue of patching.

Our findings correlated with the current challenges identified by prior research, and we uncovered important new challenges, such as the fact that patches for major vulnerabilities have a tendency to be released just before a holiday, and that legacy systems are notoriously difficult to patch and are sometimes not patched at all. The significance of planning, organization, and communication in the patching process posed additional challenges.

The contribution of this thesis to the patching topic is that we have identified "Planned patch delay" as a patch policy that contributes to a high security posture, provides time for patch planning, and mitigates a number of the challenges that might arise during the patching process.

Keywords: Patch, Security patching, Patch challenges, Patch legacy, Patch meetings, Patch policy, Patch prioritization, Patch process

Table of contents

List of figures

List of tables

1 INTRODUCTION

The quantity of digital solutions' benefits has increased as digitalization has advanced. On the other hand, dangers usually accompany benefits. The incidence of cyberattacks on corporations has increased in recent years. The amount of ransomware in Germany increased 3,256 percent between 2020 and 2021. (Sonicwall, 2022).

Applying security patches to systems for known vulnerabilities is one technique to mitigate threats for systems. Security patch management is a word used to describe this process. Regularly, testers and users identify vulnerabilities, necessitating the rapid development and implementation of security fixes to prevent their exploitation.

This thesis is presented from the perspective of a business that employs a system that has been tailored to its specific needs and incorporates a variety of software and procedures. Typically, the process begins with the discovery of a vulnerability; many, if not most, of the discovered vulnerabilities originate from external sources. The discovery might be from vulnerability-specializing companies like MITRE or FireEye. They disclose discovered vulnerabilities, and the company responsible for the affected software is expected to release a patch to resolve the issue. Businesses who utilize their software receive their remedy as a patch and are instructed to implement it. The business (from our perspective) then applies the fix to its system. From vulnerability to a completely patched system, there are several challenges to overcome. Because many systems are tailored to fulfill unique needs, the patch must be manually implemented by persons knowledgeable with the system, software, and vulnerability.

The patching process has several known challenges, including the fact that it can be complex, that there are no defined "best practices," and that there are few or no guidelines or frameworks for patching.

In most circumstances, however, creating a security patch is not straightforward, and even delivering the patch might be challenging, since it can interfere with system functions in certain instances. Several the greatest cyberattacks that have happened over the years, a vulnerability was discovered, and a fix was developed, yet the assault still affected businesses that had not implemented the patch. In other instances, the cause of the cyberattack could be the patch or update in itself, like in the SolarWinds Orion case (Oladimeji & Kerner, 2021). As in the case of Equifax, cyber-attacks may have both financial and reputational repercussions (Brewster, 2017).

We recognize the need of elucidating the difficulties encountered throughout the patching procedure. To make patching more possible for organizations, it is necessary to outline them and to provide practical solutions.

This thesis is motivated by the paucity of prior research on patch delays. Numerous studies have focused on improving the technical aspects of patching, but the socio-technical element of patching has received very less attention (Dey et al., 2015; Dissanayake et al., 2021). This is the area of study we wish to expand.

1.1 Research questions and problem statement

The research questions serve as the foundation for the objectives of this thesis. In addition to contributing to the creation of the thesis's scope, they act as a constraint so that one does not stray into other subjects of the same topic.

According to our knowledge, the issue of patching varies from business to business, and varied circumstances dictate why, how, and when a company decides to patch its systems. We wish to investigate the relative importance of these elements, which will ultimately determine why, how, and when businesses decide to patch their systems.

After conducting preliminary study on the issue, we chose to formulate the following research questions:

RQ1: What challenges can be found in the patching process?
RQ2: How can businesses overcome the patching challenges?

As stated earlier, this topic is something that has not been researched much, therefore we hope that with these research questions that it will be possible to form a better general understanding on how pathing in businesses work at a business level.

1.2 Disposition

The thesis is organized into six chapters. Where they do a linear progression toward the goal of the thesis.

Chapter 1: Introduction covers the basis of the thesis. Describes and provides an overview of the problem and research questions.

Chapter 2: Theoretical background

Here we present relevant information about the topic to aid the reader's understanding of the topic so that the reader can comprehend how and why the research was conducted. Here, the entire patching process is presented in detail, using diagrams gathered from previous research to illustrate who is responsible for patching and diagrams that visually depict the patching process. This chapter also contains a literature review, which identifies the research that is most relevant to our study topics.

Chapter 3: Research approach

This chapter explains and describes the research approach that will be used. The first section of the chapter describes how we approached the research, how we acquired and identified prior research, and who we interviewed and why. Thereafter, it is described how the data from the interviews was extracted and how it should be handled, as well as how we should choose prior research to establish our thesis on.

Chapter 4: Findings

This chapter presents what data we have gathered from the interviews.

Chapter 5: Discussion

This chapter is the outcome of the findings analysis and literature review. This is where we discuss the challenges of patching and how they came to be. In addition, we discuss the limitations of our own research.

Chapter 6: Conclusion

This is the last chapter where we present the final result of our research as well suggest areas where future research on this topic could be beneficial to investigate.

2 THEORETICAL BACKGROUND

In the chapter on theoretical background, prior research on patch management will be discussed. We conducted a literature review to determine the present state of patch management. The review's results will be presented later in the chapter.

2.1 Background

The attention given to cyber security is increasing at the same rate as the number of daily security breaches, and the frequency of security breaches is increasing at an alarming rate. According to the World Economic Forum's 2018 global risk report, the number of cyber breaches reported by firms has nearly doubled over the past five years, from 68 per organization in 2012 to 130 per business in 2017. (WEC, 2018, p.14) The question "how can we avoid this from happening in the future?" is frequently raised in relation to cyber security breaches. One of the common reasons of the vulnerability that attackers used to get access to the system was a known vulnerability that had not been patched. Sometimes the breached businesses were aware that a patch existed for the vulnerability, sometimes the patch was delayed because it caused an error and required someone to manually examine it, and sometimes it was in the large pile of available patches but was not prioritized to be implemented at the appropriate time.

As our pre-research and exploratory study revealed, one of the obstacles in establishing routines, rules, regulations, and frameworks for patching is that patching varies in terms of how a patch is applied and what system is being patched, according to our research participants. NIST (National Institute of Standards and Technology) attempted to provide a reference on corporate patch management systems as a complement to one of their arguments regarding the significance of patching in their sector. As this guide demonstrates, however, it is insufficient to solve the issues of patching, and considerably more rules addressing patching are required (Souppaya & Scarfone, 2013).

As we will examine in the literature review, various study papers research how patching should be prioritized based on the level of risk associated with a system. This prioritization is contingent on the quality of the security vulnerability analysis of the system; a flawed analysis will result in an incorrect patch prioritization.

In addition, there is research on best practices for prioritizing security patches based on how each security patch ranks in terms of the severity of the vulnerability it addresses. But there is little to no research, readily available guidelines, or frameworks on how well these practices are implemented in the real world, what implications the process carries for management, what preliminary training is recommended, or how to set up a secure patching process that reduces the likelihood of a patch requiring manual intervention or longer test time.

2.1.1 Process

In 2003, Nicastro created a "course of action" - plan that outlines the roles and duties at each level of the patch management process. Also aiding in the comprehension of the entire patching procedure. The model in figure 1 also contains "Track to closure/Exception/Recommendations/Approvals," which is a step related to contact with the senior management and C-level executives; we will return to this topic of our findings later.

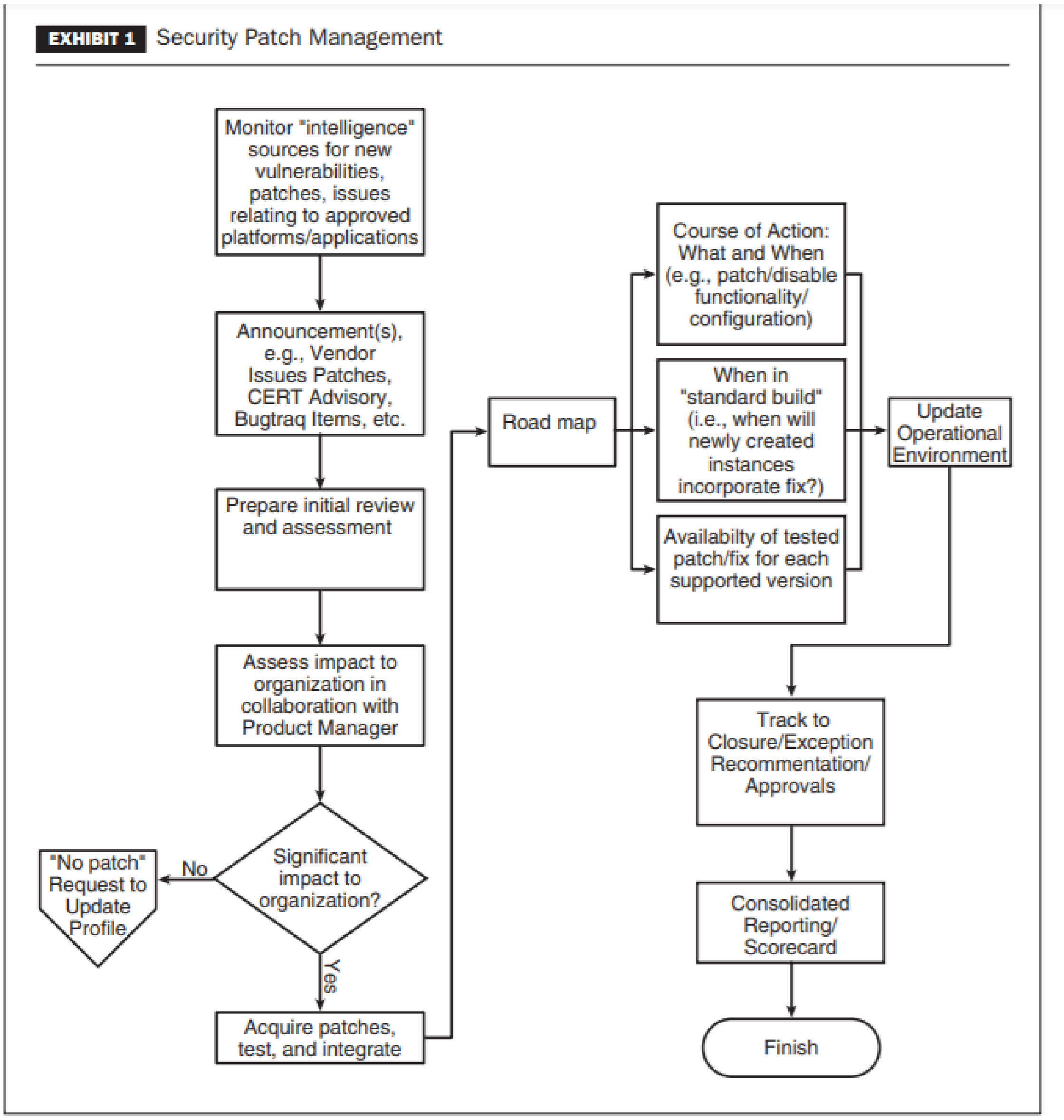


EXHIBIT 1 Security Patch Management
Monitor "intelligence" sources for new vulnerabilities, patches, issues relating to approved platforms/applications
Announcement(s), e.g., Vendor Issues Patches, CERT Advisory, Bugtraq Items, etc.
Prepare initial review and assessment
Assess impact to organization in collaboration with Product Manager
Significant impact to organization?
No
"No patch" Request to Update Profile
Yes
Acquire patches, test, and integrate
Road map
Course of Action: What and When (e.g., patch/disable functionality/ configuration)
When in "standard build" (i.e., when will newly created instances incorporate fix?)
Availabilty of tested patch/fix for each supported version
Update Operational Environment
Track to Closure/Exception Recommentation/ Approvals
Consolidated Reporting/ Scorecard
Finish

Figure 1 Security Patch Management (Nicastro, 2003)

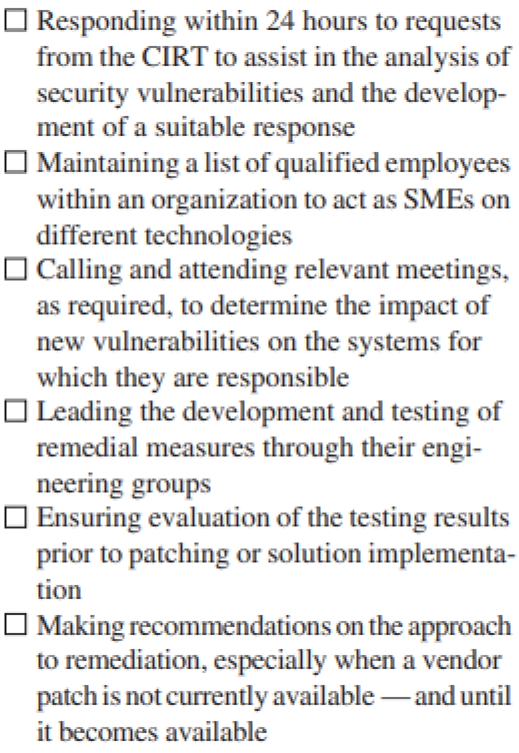
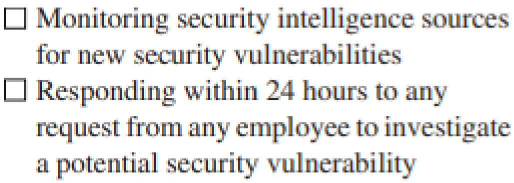
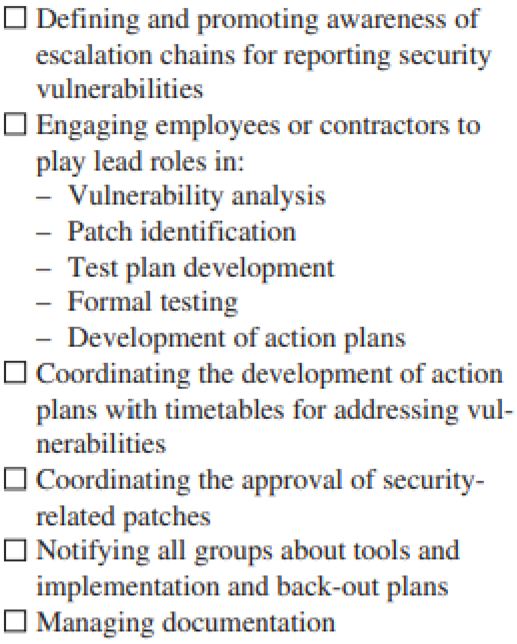
As shown in figure 1 from Nicastro (2003), there are many participants, various duties, and most crucially for the purpose of this thesis, multiple decisions must be made. This approach simplifies the decisions by providing a yes/no answer to some of the options, while others proceed immediately to the next step, which is rarely the case in the actual world.

Despite the article's age considering the rapid advancement of IT technology, most procedures and functions remain the same. However, some of the jobs have been automated or refocused, such as the duty of collecting known vulnerabilities discovered, which has become easier in some respects but more difficult in others. CVE's (Common Vulnerabilities and Exposures) on behalf of Mitre Corporation (CVEMitre, n.d) have made it much simpler to find known vulnerabilities. Additionally, many businesses have various sources that uncover vulnerabilities, matching with their corresponding system. For companies that utilize Microsoft, the Microsoft Security Response Center provides an extra source of individuals that search for vulnerabilities (Microsoft, 2022).

Since 2003, however, the number of vulnerabilities has exploded, making patch prioritization far more essential in the patching process than it was in 2003. In addition, the complexity of online systems and cyberattacks has increased significantly, making patching more complicated and reducing the time between the discovery of a vulnerability and its patch is implemented.

Despite the increased system and cyberattacks complexity. The roles defined by Nicastro are comparable to those of the present day. Describes the responsibilities and collaboration of the CIRT, product manager, and security advisor, this is presented in table 1.

Table 1 Roles and responsibility of the CIRT (Nicastro, 2003, p.8)

Product manager	CIRT
 □ Responding within 24 hours to requests from the CIRT to assist in the analysis of security vulnerabilities and the development of a suitable response □ Maintaining a list of qualified employees within an organization to act as SMEs on different technologies □ Calling and attending relevant meetings, as required, to determine the impact of new vulnerabilities on the systems for which they are responsible □ Leading the development and testing of remedial measures through their engineering groups □ Ensuring evaluation of the testing results prior to patching or solution implementation □ Making recommendations on the approach to remediation, especially when a vendor patch is not currently available — and until it becomes available	 □ Monitoring security intelligence sources for new security vulnerabilities □ Responding within 24 hours to any request from any employee to investigate a potential security vulnerability  □ Defining and promoting awareness of escalation chains for reporting security vulnerabilities □ Engaging employees or contractors to play lead roles in: – Vulnerability analysis – Patch identification – Test plan development – Formal testing – Development of action plans □ Coordinating the development of action plans with timetables for addressing vulnerabilities □ Coordinating the approval of security-related patches □ Notifying all groups about tools and implementation and back-out plans □ Managing documentation

Nicastro (2003) describes the role and responsibilities of the CIRT. In most cases, the role, and responsibilities of the CIRT are likely carried out by cyber security responsibility in smaller to medium-sized businesses, while some of the responsibilities are carried out by an external CERT/CIRT.

As we can see from the roles, however, many of them include coordination of chosen employees and decision-making over course of action. The purpose of this thesis is to determine which judgments these positions could or should not make.

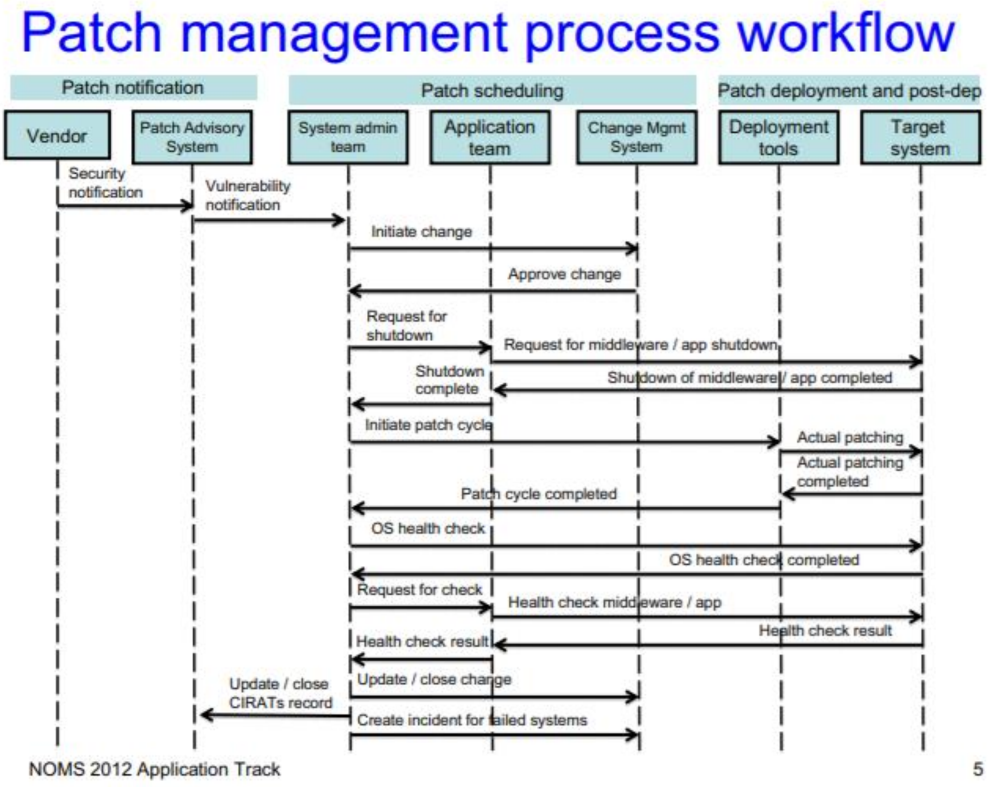


Figure 2 Patch management process workflow (Huang et al., 2012)

In figure 2, Huang, et al. (2012) show a "roadmap" similar to Nicastro et al. (2003) that depicts the patching process, with the addition of who is doing the procedure and its phases. However, this number is intended for large organizations that utilize cloud computing, as the patching procedure involves numerous teams. This diagram is considerably more specific than Nicastro's, depicting the task-closing procedure.

2.2 Literature review

This chapter presents the literature review that was conducted following the thesis. The preferred methodology for this phase is systematic literature review (SLR). This method has clear steps, where we used the steps that had the most uses for our thesis. Starting with describing the methodology following with motivation.

2.2.1 Literature review methodology

There are several ways to conduct a literature review. We utilized Booth & Grant's (2009) article describing the various literature review approaches to distinguish between them and select the most applicable for our thesis.

After examining the benefits and drawbacks, we concluded with a systematic literature review we decided to follow the guide from Okoli & Schabram (2010) visualized in figure 3. As it is a more straightforward strategy that identifies, evaluates, and chooses research that can answer a question or questions that have been clearly articulated. Since the purpose of our thesis is to address the specified research questions, this is the appropriate literature review methodology. A systematic literature review has the features of identifying what is known, making recommendations for practice, identifying what is unknown, and making suggestions for future study. We employ this process to determine what is known and to propose future study based on what is known and unknown. The purpose of the literature review is to identify research gaps, and the purpose of the thesis is to fill these gaps in order to broaden the intellectual foundations of this issue.

.

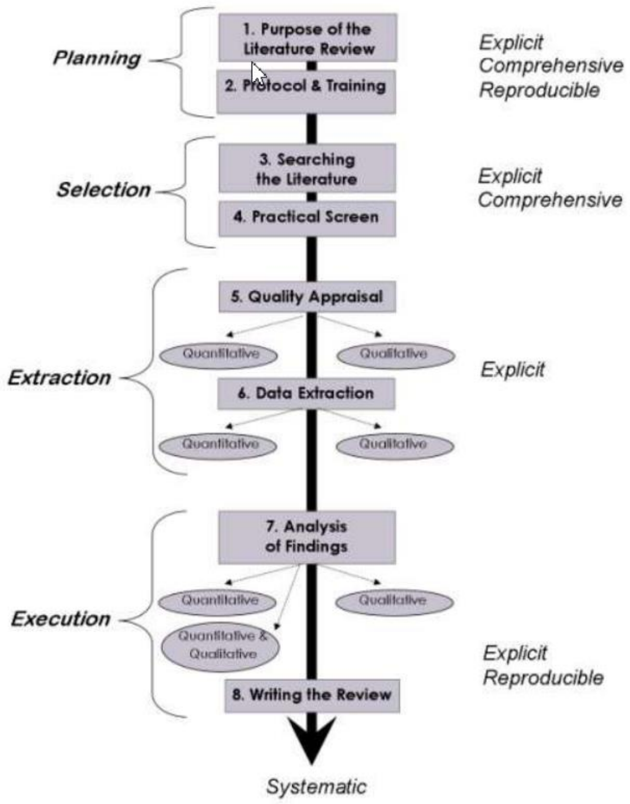


Figure 3 A Guide to Conducting a Systematic Literature Review of Information Systems Research (Okoli & Schabram, 2010)

2.2.2 Literature criteria

For our thesis to have validity we have criteria for what literature we should include in our literature review. Those points should be simple in essence to make it easier to search for literature. The inclusion criteria are criterions that needs to be filled for literature to be included, exclusion criteria are criterions that makes literature unusable for this thesis.

Inclusion criteria

- Full text of peer-reviewed conference or journal article in English that is accessible.
- A study that relates to or addresses at least one phase of the software security patch management process.

Exclusion criteria

- Workshop articles, books, and non-peer-reviewed papers such as editorials, position papers, keynotes, reviews, tutorials, and panel discussions.
- Short papers (i.e., less than 6 pages).
- Not published before 2005, preferably not older than from 2010.
- A study that reports only numerical analysis, algorithms, mathematical techniques related to software security patch management
- A study that is only focused on hardware or firmware.
- A study that is not in the domain of software security patch management (i.e., outside the focus area in Figure 1(a)).
- Full text is unavailable

2.2.3 Search tools

The search engines we used to be able to find literature were Google Scholar and Scopus. These search engines were able to give us fulfilling results as they collect research papers from multiple sources, and we are easily able to sort out the relevant literature within our criteria. The search string used in Scopus can be found in table 2.

The following search queries were used to find literature: AND - query in a query is used to search for articles containing both words in the query.

- Security AND Patching
- Cybersecurity AND Patching
- Security patching, Management, Patch prioritization
- Patch management AND Cybersecurity
- Patch AND Management AND Prioritizing AND Security

Table 2 Search query in Scopus

Search string
Patch AND Management AND Prioritizing AND Security PUBYEAR > 2010 AND (LIMIT-TO (PUBSTAGE,"final")) AND (LIMIT-TO (LANGUAGE,"English")) AND (LIMIT-TO (SUBJAREA,"COMP"))

2.2.4 Quality appraisal

To evaluate the publication's quality, we evaluated a variety of parameters to narrow down the number of sources. With the use of Google Scholar & Scopus you will find research papers from different publication sites. On many of these websites, you can view the number of times this work has been cited; the more citations, the greater the likelihood that the paper is of high quality. We have also ignored articles from other media, since they might be less reliable, and have relied solely on websites devoted to research papers, as these papers are frequently produced by professionals.

2.2.5 Selected literature

As our objective is to determine which business decisions influence the security of patching, it is necessary to examine patching from multiple perspectives. This may cause confusion regarding the overall view and how everything relates to the patchwork motif. Therefore, categorizing the literature into concepts is an essential technique for providing a more comprehensive understanding, shown in table 3.

Table 3 Concepts.

Patching in general	Patch prioritization	Patch management	Patch challenges	Legacy patching challenges

Additionally, it helps us optimize our literature search. As we wish to conduct a search that is as comprehensive as possible, we may determine whether we lack research on a certain concept and then search for further literature on that concept so that it has the same grounds as the other concepts.

2.2.6 Data extraction

This is the sixth step in the systematic literature review. Here we are compiling a list of all the literature we have chosen to use and from there we must extract the research that is relevant for our research. (Okoli & Schabram, 2010).

There is however extraordinarily little research done on patch management. One of our main sources Dissanayake et al. (2021 p.794) writes in their data extraction section:

> *In this paper, we present the first, to the best of our knowledge, Grounded Theory study exploring the role of coordination in security patch management*

We have chosen to compile the list in themes for a more efficient and clear view of the research we have collected and had any value in our research, listed in table 4.

Table 4 List of selected literature

Articles	Patch in general	Prioritization	Management	Legacy challenges
Dissanayake, et al. (2020).	X	X	X	
Dissanayake, et al. (2021).	X	X	X	X
Dissanayake, et al. (2022).	X	X	X	
Saieva & Kaiser (2020)	X		X	
Korman, et al. (2017)		X	X	X
Olswang, et al. (2022)	X	X		
Dey et al. (2015)	X		X	
Cavusoglu et al. (2008)	X	X		
Dietrich et al. (2018)	X			
Huang et al. (2012)	X			
Li et al. (2017)		X	X	
Li et al. (2019)	X			X
Igure et al. (2006)	X			X

2.3 Results from literature

Below the results that we have gathered from our literature will be presented. The findings that we have found covers the topic of patching in businesses and locates some of the issues within patch management. These topics include Coordination/Socio technical factors, patch meetings, legacy systems, automation of systems, bad patching and testing environments, patch prioritization and patching policies.

2.3.1 Coordination/Socio Technical factors

The Dissanayake's Theory of the Role of Coordination (2021) (figure 4) is one of the few or only existing figure that examine patch coordination. Their theory attempts to determine what coordination of socio-technical interdependence is in the security patch management process. And since coordination is intimately tied to commercial decision-making and management, selecting how the management and process will be organized is vital to the outcome. As seen in their figure, there are several factors that influence the results/breakdown.

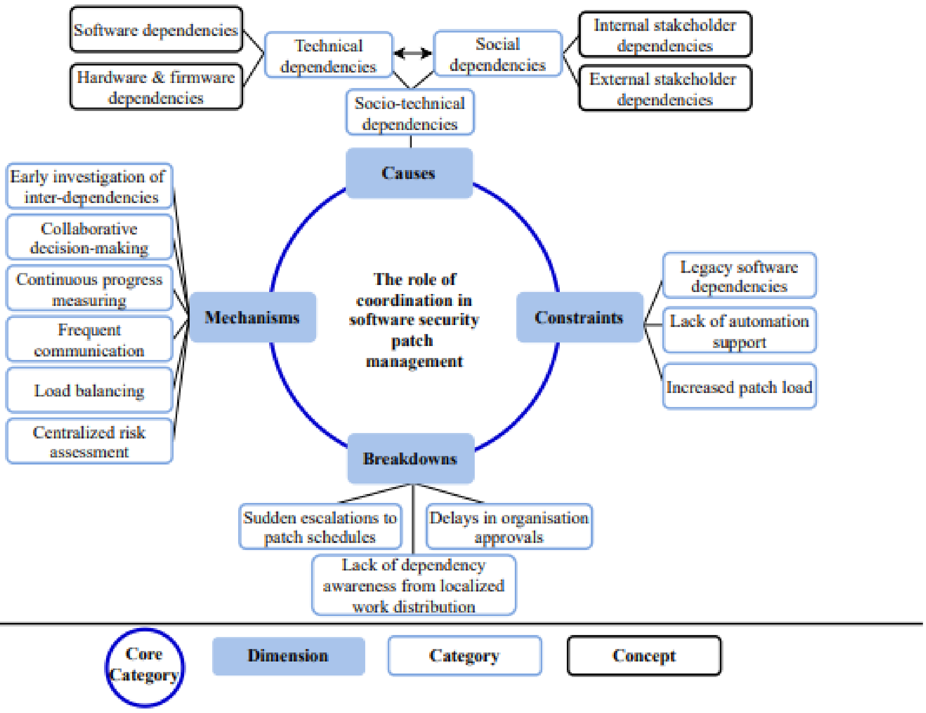

Software dependencies
Hardware & firmware dependencies
Technical dependencies
Social dependencies
Internal stakeholder dependencies
External stakeholder dependencies
Socio-technical dependencies
Causes
Early investigation of inter-dependencies
Collaborative decision-making
Continuous progress measuring
Frequent communication
Load balancing
Centralized risk assessment
Mechanisms
The role of coordination in software security patch management
Constraints
Legacy software dependencies
Lack of automation support
Increased patch load
Breakdowns
Sudden escalations to patch schedules
Delays in organisation approvals
Lack of dependency awareness from localized work distribution
Core Category
Dimension
Category
Concept

Figure 4 The theory of coordination in software security patch management

Even though their study focuses on the health sector, the results are applicable to other industries. The health industry contrasts from other sectors in that the system must be operating at all times, and the repercussions of a patch that disrupts the system are severe (Dissanayake et al., 2021).

2.3.2 The importance of patch meetings

The complexity of server patching on a big scale, when several servers operate distinct sections of the system but are still dependent on one another, increases dramatically. In some instances, there may be legacy software (irreplaceable older software) that does not work with the patch and must be modified to work with newer configurations of the patch; this increases the time, communication, human interaction, and number of developers working on the patch to deliver it within the patch window. In the health sector research conducted by Dissanayake et al. (2021), a monthly patch meeting was the solution. This meeting lasted an average of thirty minutes, and there was also a post-patch meeting. It would be interesting to observe if patch meetings are common in other industries, given that many commercial and technical decisions, as well as the formulation of plans, are made at patch meetings. The more the importance and complexity of the system, the greater the importance of a patch meeting.

This is a significant business decision in and of itself. It does include all stakeholders in the patching process, necessitates prior planning, and demands coordination. This is something we should investigate more.

Collective decision making in patch meetings

An essential conclusion of Dissanayake's theory about patch meetings is that all parties must debate vulnerability assessment and patch priority. Sharing vulnerability knowledge from the cybersecurity duties to other sections of the company has enormous benefits for the business, but it also ensures that business choices regarding patching are based on analysis and a broad skill base (Dissanayake, 2021).

2.3.3 Legacy system patching needs coordination

Positive outcomes of patch meetings include the sharing of expertise and the ability to organize duties among the available personnel. This is especially true in regard to patching legacy systems. This may be because the system being patched is dependent on the legacy portion of the system, or because the legacy system must be reconfigured, modified, or built to accommodate the new system patch. The difficulty with legacy systems, especially in modern times where external developers are prevalent in IT, is that the workforce may lack understanding of how the legacy system was constructed. "What do we do with the old system when applying this patch?" is something that may be discussed during the patch meeting. A solution may be discussed from there. If the remedy is to bring in a consultant, preferably the same one who designed the system in the first place, which is uncommon, the consultant will attend the next patch meeting to gain a better grasp of how the legacy system integrates with the other components of the system. (Saieva et al., 2020; Li et al., 2019)

Legacy systems are difficult to modify, and regular patching is required to maintain the appropriate security posture that is required in this day and age.

Korman et al. (2017) examined how they may circumvent this issue by enhancing the overall security of a SCADA system to alleviate the vulnerability that cannot be patched. Their findings indicated that with adequate network segmentation, a greater level of security may be achieved; but, as they note, this only applies to small flaws; if big vulnerabilities are detected, different options must be taken. Either upgrade the legacy system, or patch the legacy system, which is a laborious procedure. Both options are resource-intensive and expensive.

As another aspect of the findings of Korman et al. (2017), their results demonstrated that SCADA systems are more vulnerable than previously assumed. Since SCADA systems are regarded to be more secure than information systems, this research completes Igure et al. (2006). In addition, Korman et al. (2017) remark that SCADA systems do not require patching unless a serious vulnerability is discovered.

2.3.4 Only larger systems can be automated

As one of Li et al. (2019)'s participants said, "There is no way our small team could manage this many machines without automation" (of patches).

This is a true statement in businesses in larger size with a lot of endpoints in their system. Our findings suggest that there has been an increase in both automatization of "finding available patches" and "deployment of patches."

Live patching/updating, also known as dynamic updating, is a possible solution for systems that require the system to be always available. Live patching eliminates system downtime through the use of automated methods. However, it requires a substantial investment in numerous tools for optimal usage, is only suited for bigger systems, and many of the procedures in the patch process involve human engagement, therefore it is an expensive endeavor. As such, it is a crucial business decision anytime the business should invest in automating processes (Huang et al., 2012).

But there are less resource intensive tools that could be implemented if available. One of these tools is Ksplice made by Oracle. And that tool is designed to implement and deploy smaller and not complicated security updates to a system. This is a good example of a business decision that does not require extensive use of resources or budget other than some training for the employees to use. And will increase the security of the business and the focus of security employees can be allocated on other things and improving the security further.

2.3.5 Bad updates and testing environment

According to the survey conducted by Li et al. (2019), nearly every participant has encountered "a bad update." Multiple factors may contribute to a faulty patch, however in most situations the issue is either that the patch that will be deployed will break something in the system or that the system cannot be updated without necessitating a system rebuild. Either scenario demands extensive effort to implement the update or patch. Participants in a survey conducted by Li et al. (2019) concluded that the problem of faulty updates was too severe, thus they ceased applying patches to problematic system components. This will eventually result in a system that is more susceptible, as several crucial security fixes are not being implemented.

A portion of their members (30/120) accepted the task of performing dedicated testing of updates and patches. Some even included quality assurance teams. They did not elaborate on how efficient it was or why and how they developed a specialized testing environment. However, this is relevant to our study, as having a specialized testing team for patches is a significant business decision for which we would seek responses in our interviews.

In one of their inquiries, they questioned about how they managed security patches that posed implementation difficulties. A participant was not performing patch testing. "Security fixes are required, and if they break something, the issue is resolved downstream." This business decision based on the belief that "someone else will solve it later», can be a recipe for heightened risk. It is difficult to determine how many of these errors are not corrected or how much they contribute to risk escalation, but we may presume that the number is significant.

With this discovery, it will be intriguing to gather data and determine if a faulty update is something to be aware of and, if the potential exists, to take precautions.

2.3.6 Prioritization of patches

Patch Prioritization is the process through which numerous concerns that have been managed in connection to patching are ranked in order of importance: Which of the proposed patches is the most important to implement first? Would it be beneficial to implement this patch? Is it possible to postpone the patch? How complex is the patch, exactly? For each of these patches, all these factors need to be taken into consideration, and then a decision needs to be made.

We found data suggesting that patch prioritization is an important topic within the realm of patching. Since it is a decision that needs to be made regularly and has repercussions for the company.

Patch prioritization based on risk

There are other ways to describe risk in the context of patching, but we cannot use the definition that focuses on the risk associated with a potential attacker while addressing the threat posed by repairs. The National Institute of Standards and Technology defines risk as "a measure of the extent to which an entity is threatened by a potential circumstance or event" Alternately, we must apply the risk definition associated with an action that may enhance the risk of the system. And based on this definition, we can identify the typical risks of patching:

- The patch of an application could “break” the application when applying it.
- It can require the system to be down while applying the patch.
- It could make the dependent applications in the system not work the way they should.
- Misconfigurations could make it so that the system does not operate optimally.
- The patch itself could make other parts of the system it is patching vulnerable.

This requires information security personnel and system administrators to delve further into risk-analysis, as risk-analysis is often an evaluation of the system's vulnerability to an external attack.

Another risk associated with patches is putting the entire system at risk if the patch is not implemented in time before the vulnerability, it addresses is exploited. And this risk is titled "patch frequency"; it must be prioritized since the frequency with which patches are delivered is too overwhelming for the business, so they must decide which patches to deploy immediately, and which can wait. As an example, if a company has ten patches to install to its systems, it must determine which patch addresses the most serious vulnerability, or "which patch reduces the greatest risk?" (Cavusoglu et al., 2008; Li et al., 2017).

Most notably is the Olswang’s et al. (2022) “Prioritizing vulnerability patches in large networks” that brings forth the diverse ways to prioritize patching in large networks. This is more interesting for our thesis, since larger networks require more decisions rather than smaller networks with less systems to patch and prioritize.

2.3.7 Patch deployment policies

When addressing the deployment of patches, many procedures are followed. And best practices vary between industries dependent on the severity of an attack, the availability, and the level of confidentiality in the industry. The policies about when to apply a patch are essential to guarantee that the whole organization is aware of when system downtime may occur, and that recent patching may be the source of any issues that may arise. Patch deployment procedures are not necessarily applied to the entire system; they might vary amongst system components. Therefore, the management's selection of a particular patch deployment policy is a crucial decision. The following list of several types of policies was prepared by Dey et al. (2015).

- One-for-one Policy

  This policy dictates that patches should be applied promptly when a vendor publishes them publicly. This is done to minimize the amount of time the system is exposed.

  The issue with this strategy is that patching can be expensive, and the system may need to go down in order to apply the patch. With this policy, the patch may arrive during the system's peak hours of operation.

- Time-based policy

  This is the most practical approach for achieving predictability, as it is commonly utilized. In a time-based policy, patches are released according to a predefined timetable and in bulk. Many of the major suppliers, such as Microsoft, publish their patches on a schedule and are commonly referred to as "patch-Tuesday," when they release a large update on the last Tuesday of each month.

  One of the benefits of a time-based strategy is that not only is system downtime decreased by sending patches in bulk, but the time of deployment is likely picked when system users are less active.

  In the event of patches requiring manual intervention, the predictability of a time-based strategy has the further benefit of not interfering with the staff's other duties. In businesses with a limited number of available personnel who are also familiar with the system's components, it may be difficult to implement policies other than time-based ones. With the chosen date for patching, sufficient time is provided for preparation. This is particularly crucial for small organizations when the patch-team may consist of a single individual.

  A downside of time-based policy is that it can take a long time to deploy fixes that address high-risk vulnerabilities, leaving the system vulnerable to attack, especially if the attacker is aware that the organization has a time-based patch deployment policy.

- Patch-based policy

  This strategy prioritizes patching after the delivery of a predefined number of patches. This causes the various changes to be implemented simultaneously en masse. The advantage of this policy is that in some instances, a greater number of software vulnerabilities in the system become known. The reasons for this could be numerous, such as when the software provider had a team of ethical hackers attempting to hack the software and discovered a greater number of vulnerabilities. When this rapid influx of patches is released, businesses with a patched-based policy will reach the specified number and patch quickly, therefore reducing their susceptibility to newly identified vulnerabilities.

This policy helps to communicate and facilitates the availability of personnel with system expertise. It is less predictable than a time-based patching policy, as the planned number of patches may be met at unexpected periods, as many patches may be released simultaneously.

Additionally, this quantity might be attained at less-than-ideal periods, such as system peak hours or when other IT-staff initiatives require greater urgency.

- Total-control policy

The severity of patches might vary considerably. Whereas some of them involve extremely particular or low-risk vulnerabilities, others include well-known, readily exploitable, high-risk weaknesses.

More vulnerabilities that are identified are made public, and more and more published patches are connected to specific vulnerabilities, where the severity of the vulnerability is graded on a scale.

This is becoming increasingly common, with companies such as Microsoft tying their updates to vulnerabilities and then grading the severity of the vulnerability. The grade is determined using multiple measures based on the type of vulnerability (Microsoft, n.d).

With total-control policy, the system gets patched when the sum of the severity rankings of all available patches equals a predefined number. When the system reaches a certain level of vulnerability, patches are applied.

This strategy reduces the work required for constant patching. Since the majority of published patches are considered to address extremely minor security flaws, this is typically the case. Consequently, the workforce is spared the burden of unneeded patching, allowing the IT department to focus on other tasks.

To establish a secure system, total-control policy necessitates that the specified summary severity number be low, making it, in essence, a patched-based policy. And if the number is high, the system's security is endangered.

In addition, it lacks predictability, as a fix for a high severity vulnerability may be identified and released at any time.

- Emergency-control policy

  This indicates that patches should be applied when a patch with a high enough severity rating that matches or exceeds a preset level of severity rank is published, and then all previous patches since the last time the system was patched should be applied.

  This often entails that the system would not be patched until a significant vulnerability is identified and a patch is created, and then all the minor patches would be administered in conjunction with the important patch. Despite the lack of predictability, it guarantees that the workforce is not required to assign personnel to regularly patch the system.

These are the most often used regular single-metric policies. Even though a combination of these policies is more typical, a simple example demonstrates that a combination of timed-based, and emergency-control is a very logical policy to have, as the system is fairly secure, has high predictability for the workforce, and will be patched when a critical vulnerability patch is released to avoid being at a high risk. These policies are crucial management choices that can affect not just the whole organization, but also the allocation of the personnel and the cybersecurity.

2.3.8 Patch delays

The longer it takes from the moment a vulnerability is disclosed to the public until it is patched, the more susceptible the system is to the vulnerability. With the release of a patch, it is imperative to implement it in the best manner. As previously said, the best practice varies, and determining which is the best is one of the problems this thesis seeks to address. Even if there are several techniques, one thing remains constant: patching quickly is the best practice in the majority of situations (Dissanayake et al., 2022).

Dissanayake et al. (2022) created a list and calculated the potential reasons for a delay in the specified patching procedure in the health sector. Their investigation aimed to determine where the largest delays occurred.

This paper is useful for our research since they compiled a list of delays, allowing us to identify their causes. In addition, they created a description of the usual timeframes in an organization when the patch information retrieval procedure is beginning (patched released):

- Vulnerability scanning, assessment, and prioritization
- Patch Testing
- Patch Deployment
- Post-Deployment Patch Verification

This is beneficial for us when doing the interviews, as we can determine how well structured their patching process is by comparing the time, they utilize to the time listed in Dissanayake et al. (2022) table of typical time frames.

Even though this research is in the health sector, a significant amount of the data is useful to our thesis because it relates to professional applications in all domains. As there is confidential information in the system, the health industry's systems are subject to uptime and confidentiality standards. However, these criteria are relevant to other sectors, such as the energy sector, with less focus on confidentiality.

In their findings, they examined 232 tasks associated with the process of patching, of which 132 were delayed. Many delays were attributable to human factors, which is unsurprising and supports Dissanayake et al. conclusion that patch-meetings are a crucial decision to make to decrease human-related delays. Patch testing and post-deployment patch verification accounted for an equal share of the subsequent delays. This is noteworthy because the publication by Dissanayake et al. (2022) is undertaking this research on medical equipment, which, as they explain, has many legacy systems. Corresponding to the findings of Saieva et al. (2020) study on legacy systems.

In addition, Dissanayake et al. (2022) present several patch deployment strategies in their research. They concentrated on the various phases of deployment. This is a topic on which our study should expand, and since our thesis is about determining what decisions should be made, it is of tremendous help to uncover other strategies that can be implemented to supplement our conclusions. From their strategies we are going to retrieve the strategies that can be applied to a more general patch deployment. These strategies are visualized in figure 5.

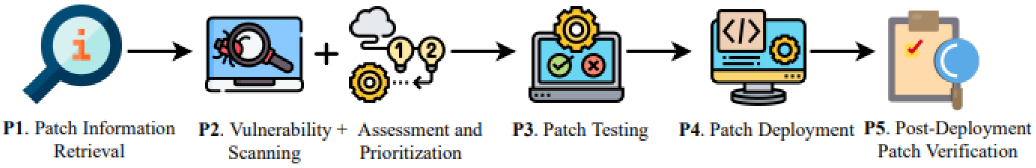

P1. Patch Information Retrieval
P2. Vulnerability Scanning
+
Assessment and Prioritization
P3. Patch Testing
P4. Patch Deployment
P5. Post-Deployment Patch Verification

Figure 5 Patching process (Dissanayake, 2021)

Strategies relating to patch information retrieval.

Their studied practitioners followed a strategy of setting timelines for patch downloads. This can be within a timeframe before for example “patch Tuesday " that is a set of time when the large vendors like Microsoft, Oracle and Adobe publish their patches. This strategy allows sufficient time to plan and coordinate the necessary steps for when the patch window should be, obtain business approval and make additional planning to ensure that the patch testing is sufficient for the upcoming patch. This strategy is correlating with Dissanayake. et al. (2021) conclusion of the importance of patch meeting.

Strategies relating to Vulnerability Scanning, Assessment and Prioritization.

They observed that many of their participants used alternatives to scheduled patching in some cases. Where in cases where they assessed that an alternative strategy was needed due to known reasons. For example, a major upgrade to a critical legacy software. Patching/upgrading a legacy system is an extensive process, as described by Saieva & Kaiser (2020), and doing it while maintaining a secure environment and keeping business operations continuance, is difficult. In such cases, the participants planned alternatives, for example “what to do, when to do it.” This process is something we can research in our thesis, as going outside the plan puts pressure on the management on making quick and decisive decisions.

Strategies relating to Patch Testing.

They found two interesting strategies. The first one is to have defined compliance policies. With the example connected to patching, being to have all legacy systems reboot, even though they are not patched. And developing contingency plans in cases of faulty patches or patches that cause problems with other systems.

In addition, they expand on how policies can mitigate the risk of delays with patch testing policies. For example, a clearly defined policy of dedicating a specific time to identify and modify dependencies and configurations in the patch testing. This is to reduce the chance of a delay happening due to the nature of dependencies in the system, parts of the system being legacy or the complexity of the system.

The other strategy was to have an investigation of prerequisites. This strategy is to avoid runtime errors when applying a patch to the system and thus avoiding a delay. The patch prerequisites that are needed to be initiated to be set up in this case are most often registry changes and preparation package installation. The benefit of the investigation is to mitigate the chance of having to go manually inn during patching to change configurations during patch deployment. (Dietrich, et al., 2018)

Third finding in Dissanayake's et al. (2022) strategies to patch testing was a strategy adopted in preparation for the machines for patch deployment and to avoid delays arising from complexity of patches due to patch dependencies. The practitioners dedicated a specific time to identify and modify the dependencies and configurations during patch testing. An example of this is to cluster patches based on similarity, to reduce time used on manual configurations on single patches. This strategy is known as "Group Policy Object" (GPO). This is also closely related to Olswang et al. (2022)'s strategy on patch prioritization strategies on coupling of patches.

Strategies relating to Patch Deployment.

The strategies of Dissanayake et al. (2022) describes that their practitioners relating to patch deployment correlates to our other literature. The strategies suggested to tackle the challenges of patch deployment like coordination, service availability, business disruption and patch schedules, patch grouping/patch clustering, load loading (adjusting the load on the servers to minimize disruption of service) and failover (having a backup server that can run the system while the server is patching, to continue operations without disruption).

Interesting finding was that their practitioners decided to switch to manual patch deployment for business-critical server patching, legacy systems, complex patches, redeployment of patches that were deployed with an error. The practitioners justified this policy because they wanted to avoid further delays, but as Dissanayake et al. (2022) mentions that strategy can be a source of delay itself.

Agile patch deployment

Dissanayake et al. (2022) participants reported that they used agile patch methodology. This is interesting as agile methodology is widely used in information system development and is therefore a methodology that most employees that work within IT are familiar with. The reason agile methodology in patching is interesting is that an agile patch team can face challenges, errors, or problems in development rapidly.

Strategies relating to Post-Deployment Patch Verification

It is advised to have a defined set of procedures for post deployment patch verification. Those several procedures that their participants used to verify patch status was:

- Monitoring the system for if there were any unexpected issues, performance problems, and unexpected issues.
- Analyzing the system logs.
- Collecting user feedback.
- Scans to verify that the targeted security vulnerability was patched.

An additional strategy to challenge delays and problems with post-deployment patch verification is to have a patching tracker. That is a platform or a software that allows team members to document patches, deployment status, eventual errors, etc. This connects with other challenges with the whole patching process, as many of the challenges are connected to time, coordination, and complexity.

2.4 Summary from literature analysis

To conclude and circle back to our problem statement. We have observed that there are a variety of approaches and policies for addressing unique patching issues. There is some literature that identifies conditions that might lead to patching difficulties.

There is rigorous literature, such as Dissanayake's (2021) theory, which addresses several socio technical aspects. However, I t addressed fewer decisions related to patch management than ideal.

In contrast to being overly technical and lacking in the human element of managing a patching process, Li et al. (2019) study attempts to solve the patching problem using automation and live updates that do not interrupt business operations.

Coordination, delays, complexity, and testing are a summary of the primary reasons of difficulty in managing the patching process.

2.5 Gaps in prior literature

We found a lot of research on the challenges that exist in patching systems, and a lot of patching of end-user's applications. However, on the topic of patch management there is extraordinarily little research done. Therefore, it is hard to identify gaps as this is a very new field of research with few contributors.

One of the gaps is that it is too hard to do research on specialized industry specific software. As this software can be tailor made to each individual business, and therefore the research, learning and results are made less applicable or relatable to other more general sectors or businesses.

Another gap is the human aspect of managing a patching team, there is only one paper by Dissanayake (2021) that revolves around the coordination for the patching team, where the topic of coordination has a potential for much more research in the future. The last gap with the lack of patching research is that there is no literature on topics like outsourcing of patching teams responsibilities

3 RESEARCH APPROACH

In this chapter, we present the methodology that was used to perform the research for this thesis. Proceeding through the rest of the steps in the procedure following the SLR, all the way up to the conclusion of the thesis. Beginning with a summary of the approach taken in the research and moving on to the manner in which the interviews were carried out.

3.1 Qualitative research

We looked at and established assumptions about how we should gather data as our first step in determining how to discover answers to the patching management difficulties. As our data required to be about specialist software, therefore, automated Windows updates are not prioritized.

We settled on establishing the thesis based on the interviews, where the participants are from many businesses and employ specialist software in their everyday operations, and where we want to see difficulties that span across industries.

After reviewing the prior research, our initial plan to narrow the scope of our research to specific types of businesses was modified. According to prior research, participants were not limited by size, specialization, or industry, and since the objective of this thesis is to expand on the field of patching based on prior research, we determined that we cannot limit our research to either the size of businesses or their specializations. This will likely be accomplished by interviewing employees who are familiar with the company's patching procedures.

As previously indicated, we determined that qualitative research would be the most appropriate technique for this thesis since we want in-depth information on how organizations make decisions regarding patching. In order to obtain these answers, we will conduct interviews with many businesses that do security patching in-house to see how they manage receiving a large number of security patches from their software vendor.

We hope that by conducting interviews with businesses, new solutions will be uncovered that were not previously discovered by prior research.

Because we believe the result of our research questions will be in a form that must be interpreted rather than real numbers, our initial inclination is to do qualitative research rather than quantitative research.

For our research data collection method, we employ an interpretivism paradigm in which we construct our research questions and interview guides based on our interpretations of prior research, problems identified by prior research, our subjective understanding of theoretical frameworks, subjective perspectives, and intuitive field understandings.

3.1.1 Semi-structured Interviews

A semi-structured interview approach is ideally suited for study in the field of information technologies in which the researchers wish to determine the why and how decision-making in complex systems are done. Therefore, we decided to do the interview in this manner.

Semi-structured interviews are conducted when there is existing research and expertise on the issue, but the researchers wish to build a deeper insight and increase their knowledge of the topic. Semi-structured interviews are conducted similarly to structured interviews, with the exception that it is permitted to explore when the research reaches a subject or issue related to the topic being investigated (Wilson, 2014, p.24).

Strengths

It provides a mechanism to steer the interview back to the topic if it strays too far from the topic, it provides interviewers with flexibility, and it permits a broad comparison between interviews. These are the strengths of semi-structured interviews. Other strengths include the ability to uncover complex topics through probes and clarifications; it provides a mechanism to steer the interview back to the topic if it strays too far from the topic

Potential challenges with Semi-structured interviews

It is common for semi-structured interviews to succumb to what is known as the "interviewer effect." This is a phenomenon in which the interviewer's conditions have an impact on the quality and quantity of information gathered from the interviewee by the researchers. These factors could include the medium (or mediums) in which the interview is done, the time of day, the age and history of the person being interviewed, and any other relevant demographic information.

Preparation to the interviews

As you would in person, structured interviews should be conducted through a media that allows for two-way conversation. However, the participants for this study project are likely to be situated in distant regions of Norway, making travel a needless cost. Consequently, our alternatives will be phone or videoconferencing. As a result of the COVID-19 epidemic, society has been compelled to learn how to utilize video-conferencing-tools extensively; thus, we expect that our interviewees would welcome our use of videoconferencing as an interview-medium. To get in contact with interview subjects, we sent out an email asking different businesses if they could arrange a meeting discussing our topic, the email that was sent out is in appendix C. Of those that responded we had to distribute a consent form for the interview subjects to sign, which is listed in appendix B.

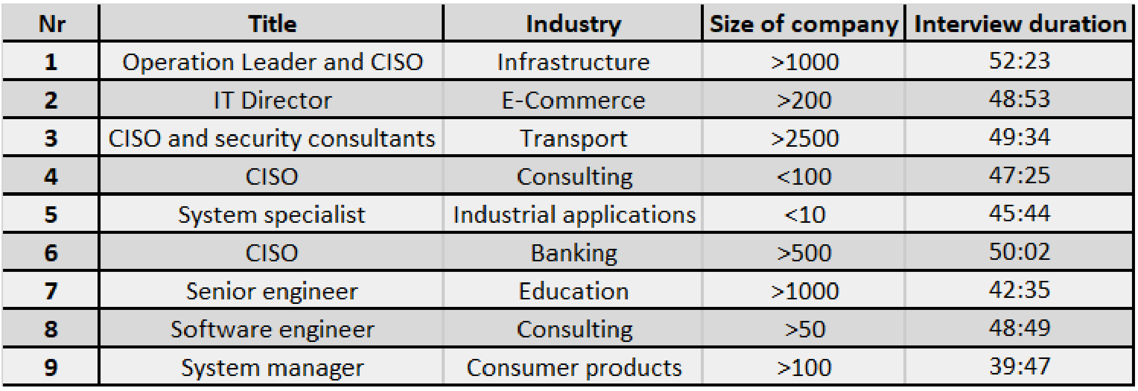
As we are doing semi-structured interviews, we must also prepare for extra questions to refocus the conversation on the initial subject and themes. This will ensure that we obtain the answers to our chosen question (Wilson, 2014, pp. 17-21).

3.2 Interviews

Throughout the semester, we conducted nine interviews with businesses of varying sizes. All the interviews were performed digitally, as many of the participants were not local and that was the preferred method of the businesses. The interviews would typically begin slowly, although this might be due to the questions provided in appendix A. After 15 minutes, however, the talk began to flow effortlessly. We frequently asked follow-up questions that were not included in the interview guide, which is why the interviews typically lasted over 50 minutes.

Below, in table 5, we give various details regarding the businesses with which we conducted interviews. The list has been generalized to preserve privacy.

Table 5 List of interview subjects



Nr	Title	Industry	Size of company	Interview duration
1	Operation Leader and CISO	Infrastructure	>1000	52:23
2	IT Director	E-Commerce	>200	48:53
3	CISO and security consultants	Transport	>2500	49:34
4	CISO	Consulting	<100	47:25
5	System specialist	Industrial applications	<10	45:44
6	CISO	Banking	>500	50:02
7	Senior engineer	Education	>1000	42:35
8	Software engineer	Consulting	>50	48:49
9	System manager	Consumer products	>100	39:47

3.3 Research design

Following Huberman and Miles's (2002) description of exploratory research as a case study, we utilized a case research design. We determined early on that we would investigate the topic of "Security patches." Based on previous research, we determined that the patching process was sufficiently intriguing to warrant further investigation due to the range of practices and absence of frameworks. Since our thesis was not a continuation of a previous project, we built the project with a loose initial design that followed the thesis throughout and covered the greatest amount of territory during the patching process.

We opted against developing a conceptual framework for the topics we would not investigate. But we were going to analyze the patching phases defined by Dissanayake et al. (2021) model of the patching phases and the model of delays conceptual framework.

3.3.1 Validity, reliability & quality

In Huberman & Miles (2002), they provide a set of criteria for evaluating the validity, reliability, and quality of articles. In forming our thesis, we consulted the sources that we have cited.

The validity of the findings is supported by the fact that they do not differ significantly from and contradict other papers in the area. Our findings are not novel, but the discussion section explains how they contribute to the body of knowledge in the patching field. The findings are straightforward and easily comprehensible, and the statements have not been altered or manipulated to fit our narrative. The predictions and hypotheses we make in the conversation are related to the results and based on them.

Our research's reliability is founded on the fact that our literature is credible, peer-reviewed by fellow students, and derived from reputable sources. The information we have obtained is not contradictory; rather, it expands or fills in gaps in previous research. The research questions have a purpose and rationale for their creation. The thesis's characteristics are stated in an accessible style. Our selection of participants was not from a specialist sector or business, but from a broad industry, therefore the findings are reliable.

To evaluate the quality of our thesis, we are adopting Huberman & Miles' (2002, p.314) criteria. The findings contain an in-depth explanation of the participants' opinions, are written in an easily understandable format, and are consistent with and may be related to the findings of previous research. We describe why we picked the samples and why we developed the interview guide in the thesis. The interview guide is based on our own assessment of earlier research papers and "future research" phrases from prior study. The frameworks utilized in the thesis are based on credible, peer-reviewed, and credible sources.

3.4 Data analysis

In order to examine the interview data, it was necessary to sort the obtained information. This was in the form of recordings of the interviews, which we could subsequently transcribe to extract the interviews' essential themes. The interviews were conducted according to the interview guide and the method given for semi-structured interviews. As a result of the interviews, we were able to identify crucial data that may provide greater insight into our research questions as well as data that connected with previous research. In order to comprehend the interview data, we transcribed the interviews to determine the primary themes: Legacy systems are challenging, Patching process, Patch Delays, Communication, and Patch prioritizing. From these themes, we classified significant findings.

4 FINDINGS

Following and continuing the research already done on the patching theme, our findings from the interviews did not have many surprises in terms of how they patched. With our research on patching, we expected that some of the participants did not prioritize patching, while other participants were better than expected.

4.1 Legacy systems is difficult to patch

A significant amount of industrially specialized software required significantly more time to acquire and implement patches for vulnerabilities. This was a particular worry of one of the attendees who dealt mostly with industrial computer systems. There, the software provider was required to publish the patch and customize it for the particular industrial system. When the vulnerability is of a critical severity, fixes are distributed quicker. In circumstances where the vulnerability is not critically significant, but potentially severe, it may take up to six months to a year for the information to be transmitted to the participant.

Updates are difficult to implement on legacy systems. The participants with legacy systems said that the patching method for their legacy systems differed from that of the other systems that they typically patch. Therefore, the patching team had to allocate more time before beginning the patching job. Additionally, legacy system patches are not frequently issued making it an unplanned unusual procedure.

According to several participants, the topic to replace the legacy system with a new one is ongoing discussion. There are several reasons why a business is not updating its legacy systems to a new system, and each business and sector has its own reasons. The most prevalent explanation is that the cost is too expensive and updating a system that is likely not linked to the internet does not pose a sufficient danger to warrant an upgrade. In addition, upgrading historical systems necessitates extensive procedures, as they are sometimes intertwined with other systems that are incompatible with modern systems.

Legacy systems most often run on LAN networks, are not linked to the internet, and are typically defended by perimeter fortifications, such as a castle-and-moat policy, rather than internal defenses. A participant in one of our interviews stated:

> *You need to be physically on the premises in order to breach the system, however, a handful of gravel thrown in the electronics would do even more damage*

Because of this, several of the participants told us that the security of these systems in themselves is almost non-existent and as long as you had physical access to the systems, you had full access. However, another participant stated that:

> *It is always possible to infiltrate these systems, however it is not very cost efficient to protect them.*

It demands enormous effort to do without interfering with daily activities. As noted, before, the process of patching a legacy system frequently needs the system to be offline. Many of the participants' old systems were on LAN networks, and therefore must be taken down to apply modifications. alterations that frequently require human programming.

Even in the event of legacy system patching, the same team frequently conducts patching, according to the participants. They did so since it required the formation of a team, the initiation of coordination, and a time-consuming procedure; nonetheless, the length and complexity of patching a legacy system necessitates expert expertise. When asked if the patching team has ever been delayed in patching a legacy system, the response was that it occurs rarely but is possible.

4.2 Communications and planning in a business

We discovered that most businesses we examined had some form of connection with software vendors. This is because numerous applications were adapted to their own requirements, necessitating particular patching requirements. This taught us that the size of the business frequently correlates with the speed of communication between the supplier and the business. In addition, the size of the software supplier would indicate the effectiveness of the communication. The communication might consist of anything from meetings with providers to the sending of an email.

Patch meetings are utilized to varied degrees, depending on the degree of software specialization and the quality of communication with the software supplier. Some participants prepared and coordinated with their vendor, and on the eve of patch day, they held a patch meeting with the patch team and received instructions from their vendor on how to install the patch with the least amount of difficulty. Many participants had a regular group of patching personnel. Most smaller organizations noted that patch meetings were utilized in situations where the patch procedure was not normal, whereas bigger businesses held patch meetings more frequently. In the event of severe serious vulnerabilities, such as the Log4j(CVE-2021-44228) flaw that shook the IT industry in 2021. All the participants held multiple extraordinary patch meetings to eliminate the vulnerability as rapidly as feasible.

All the same businesses informed us that the reason they scheduled these meetings was to prevent system or production disruption. In addition, this was done to ensure that the patch was compatible with other patches.

Furthermore, it is also revealed that the communication between the patch teams and the rest of the business can be lacking. Sometimes the patching team tells the relevant parties that some systems will be patched and sometimes not. This has led to some issues because of incompatibility of some systems that were not the one that got patched.

> *There have been times where some systems were offline for a while because of a faulty patch. In hindsight this could have been prevented, but wisdom after the fact is easy.*

4.2.1 Human resources

We wished to determine who is responsible for each step of the patching procedure. Is a single individual responsible for patching? Are specific patches allocated to personnel with understanding of the system component the patch addresses? Larger firms have highly regimented patching procedures, with a specific patching staff performing the work. For our smaller company members, there was no team, only one individual. This individual successfully patched a computer by relying on expertise rather than precise procedures. Having the same team or employee install patches increases productivity and reduces the need to review patching rules and procedures.

Occasionally, communication between the IT department and C-level executives can be challenging. Several of the participants informed us that the IT department is mostly autonomous and that, as a result, the majority of the time the communication is excellent. As the management are not IT specialists, they can only assess project expenses and the IT expert's remarks. One of the participants stated:

> *When you present a project to the management, you have to show them red and green numbers, as that is what they understand.*

4.3 How interview participants described the patch process

We asked the participants how they had experienced the patching process, what challenges they had faced, and how they worked around them to successfully patch to collect knowledge about the patching process and to widen the range of challenges that are currently acknowledged.

4.3.1 Vulnerability retrieval process

We wanted to know how the participants discovered that one of their software included a vulnerability. This varies depending on the size, industry, and quality of their security awareness and security infrastructure. The participants who were on the smaller size indicated that they were notified late and through mail.

In other instances, smaller firms were not told of the vulnerability, but rather that a fix was available for a vulnerability, avoiding the step of reporting that a vulnerability existed.

For larger organizations, according to the participants, this was a frequent practice; they often received an email stating "we have uncovered a small vulnerability; a patch will be ready soon." According to the participants, this was a good approach that made the business feel important, that they were included in the process, and that made it feasible to anticipate a patch. In instances where major vulnerabilities were identified, the size of the company did not matter; they were notified, and communication between software vendors was more consistent and regular.

4.3.2 Patch information retrieval process

Most of the interviewees had a partner who would advise them about newly available patches, although the frequency and method of delivery varied. We discovered that larger companies had more partners and, hence, received more patch-related information. Some firms had both a SOC (Security Operation Center) and a CERT (Computer Emergency Response Team) that advised them of critical fixes. Because not all firms had a SOC or CERT that could offer information, they had to utilize alternative services to obtain the same data.

According to almost all the participants, there was always someone interested in security, and consequently, someone who already worked in the industry typically had the most up-to-date information. In addition, the majority of participants subscribed to the newsletters of prominent security companies such as FireEye.

4.3.3 Assessment

We asked the participants if they had any asset-tracking software or tools, as well as if there were any available patches. Such a program would significantly reduce the amount of time spent checking for incoming mail containing information that the vendor has published a patch. In addition, it is possible and has occurred for the vendor to neglect to inform all consumers of available fixes.

Participants reported that they had controlled assets, but that the asset management software does not monitor for upgrades.

4.3.4 Patch Testing

Regarding testing, how thoroughly the participants examined whether the system functioned correctly once the patch was implemented varied. Participants with more industrial controller systems (ISC) reported needing to test their systems significantly more frequently than those with IT systems. In addition, we were informed that the Microsoft fixes were never tested, as was the case for all participants:

> *We can never test the patches from Microsoft better than Microsoft anyways, therefore we choose to trust their patches.*

4.3.5 Planning

We asked the participants if they utilized or implemented any frameworks or standards to ensure that their patching procedure was consistent each time. The smaller businesses of the participants did not have as much in writing, since the individuals engaged in patching had knowledge of the process, and it was quicker if a change in the process did not necessitate updating a standard as well as the patching procedure.

Significantly more processes were defined as policies in the participants' larger businesses. This is since there are teams performing the patching.

When participants opted to patch their systems varied greatly. Some adopted the strategy of patching as soon as fixes became available, while others took a more cautious approach, waiting a week or more to see if other businesses in the sector saw any difficulties with the patch before patching themselves. There were participants that adhered to the time-based policy regardless of the severity of the patch. All participants, however, stated that vulnerabilities with substantial risk were regarded exceptional and were patched as quickly as possible outside of schedule.

4.3.6 Patch deployment

According to the majority of companies we questioned, patch deployment times vary depending on the type of patch and the patch's source. All the participants who utilized a Windows-based device said that Microsoft's updates were always immediately deployed, however their implementation time may vary. However, most participants assured us that Microsoft's security fixes will be installed within a week.

Typically, patches were delivered in a test environment or on a limited subset of users to identify any potential problems. Some companies that worked primarily with ICS informed us that updating these systems was not a top priority, as new patches for these systems may arrive every six months, and it could take six more months to tailor it to their specific needs. A specific participant informed us:

> *The process from when a patch arrives to when it gets implemented takes a minimum of one year*

4.4 Businesses delays the patching

We questioned the participants on the causes of patching delays or postponements. This aimed to expand on Dissanayake et al. (2022) research on patch delays, however the sector that was investigated was the medical industry, therefore we were curious whether other industries had similar patching issues.

4.4.1 The planned delay

The most prevalent reason cited by participants for delaying a patch was a planned delay. They waited a period of time, often two weeks, to determine whether other businesses that had applied the patch had had any issues. In certain instances, a patch may contain defects or be incompatible with particular systems. This is a perfectly fair reason to delay a patch, and it may be advantageous for a business if other businesses with comparable systems sort out the issues, especially if some of the participants need 100 percent uptime. This planned delay can save time and resources. This method was more prevalent among the participants' smaller businesses, whereas the bigger enterprises did not apply this strategy.

4.4.2 Other projects

According to the participants, another explanation for the delay of a patch was that there were other projects with a greater priority. One participant stated:

> *[...]This is never in the case of security patches.*

In the event of patches that fix some functionality, it is probably not crucial to patch immediately, but it is still not recommended to delay a patch.

4.4.3 Patch load

Sources of delay are not directly linked to the patch or system, but with trends of the cyber-attacks. That being that there is a strategy used by hackers to attack the system just before a holiday or a period where the business is too busy to follow the normal procedures.

> *In most cases we do not have to prioritize patches, and patches get implemented quickly by our staff. But it is typical that just before Christmas times, the time where most of the risk for cyber-attack is the highest, a lot of patches are deployed, forcing us to work overtime to implement them before the staff takes Christmas holiday.*

This sudden surge in patches would put many businesses behind on the patching schedule. Combined with all the other reasons for delays and in addition if the business has complex systems, where patches are difficult to implement, it will put the business further behind.

4.4.4 Not compatible patches

In some cases, the participants said that it has happened that they choose to not implement a patch, as the patch made the software not be compatible with other parts of the system. This was truer to the businesses that had special software tailored to their business or industry. The most common reasoning for this was:

> *The systems don' t really need security updates that often, since it is walled behind several layers of security, and most often requires a physical presence.*

We also learned that it was common to skip patches to these special software's, as they often were incompatible with the current production systems.

4.5 Patch immediately policy

There were a few of the participants that told us that as soon as a patch was available, it got deployed. What they told us was:

> *We have a policy with patches that we would rather have them break, rather than have vulnerabilities in our systems. Usually, the outcome of a breach is more severe than if a patch fails.*

This claim was, however, not the standard for in our segment of participants. The ones that did have this policy, said that this had never resulted in any major issues and the smaller issues it had caused were easily fixed.

4.6 Server provider

A portion of the participants had service providers who were responsible for upgrading, maintaining, and operating the servers and technical aspects of the system. In certain instances, the server provider performed the patching, although in the case of the participants, there was continual contact between the patch team and server provider. The reason the participants gave for having a server provider was that they had superior knowledge of the system they had, and in many cases, the server provider they had chosen was an expert in their type/sector/type of system; as a result, the participants felt it was the optimal choice in terms of cost, security, efficiency, and availability. In addition, this was especially important for smaller businesses as it was a cost-effective option who wanted to digitize their operations.

4.7 Conclusions of findings

To summarize the findings that expanded the current knowledge around patching we created a model visualizing this in figure 6:

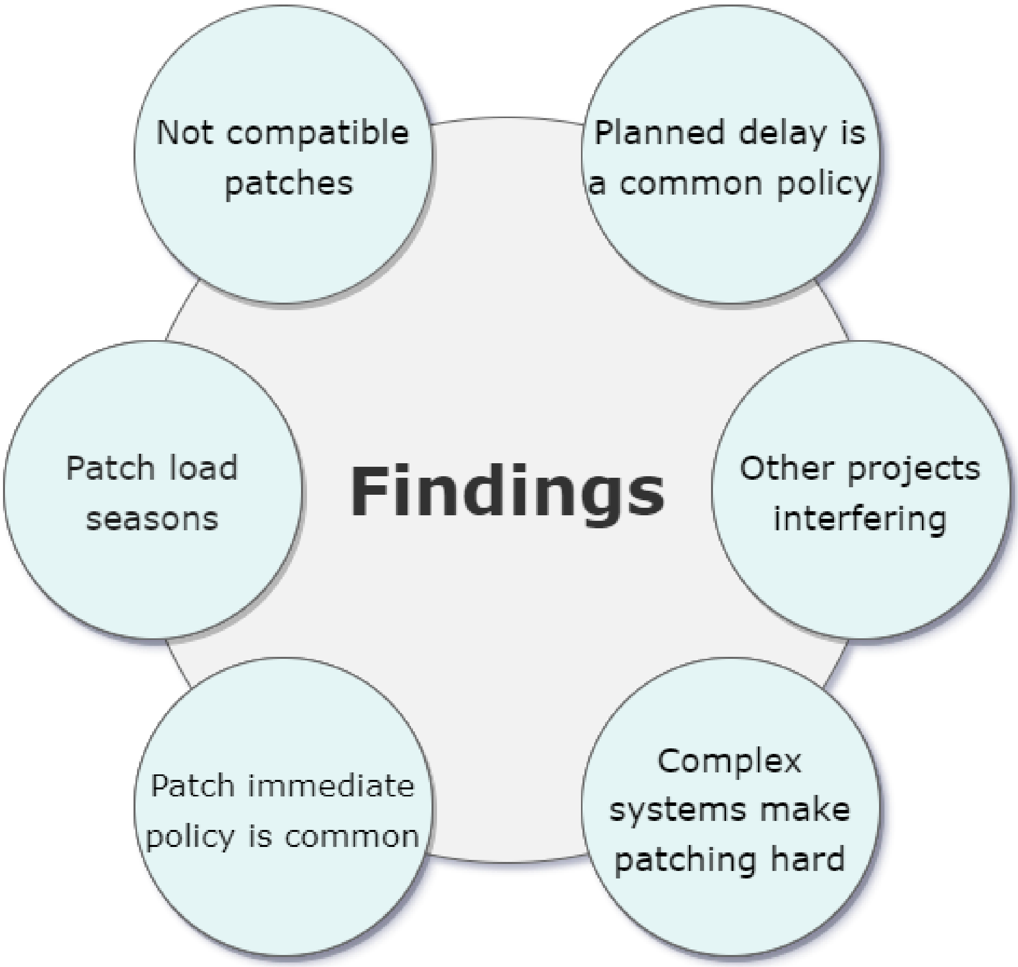

Findings
Not compatible patches
Planned delay is a common policy
Other projects interfering
Complex systems make patching hard
Patch immediate policy is common
Patch load seasons

Figure 6 Results of the findings

These were the findings we found that were not covered in earlier research and had an impact on the research theme. We experienced that many of the things that the interview participants talked about had been discovered by earlier research, which is contributing to the validity of the findings. The findings that are described are visualized above to make it simpler to have an overview.

5 DISCUSSION

This chapter is the result of the findings chapter and the literature review. This section discusses the difficulties of patching and their origins. Also discussed are the limitations of our own study.

The purpose of this discussion is to determine what the best practices are, what the challenges are, and how these challenges may be overcome. The most significant findings that augment the present body of knowledge on patching concern a strategy that we refer to as "the planned delay." Many of the issues we cover are neither novel nor groundbreaking in patch management, but they do add to the current body of knowledge on the subject and provide insight into how various businesses patch differently.

5.1 When is it best to patch programs?

What are the "best practices" for patching and prioritizing patches? The findings show that determining when to apply a patch is not as straightforward as one may assume. As the practice of "patch immediately" was not something that any of the participants in our survey practiced, nor did they intend to adopt it as a policy.

The issue is that several factors determine "when to patch," and there is no single "best practice" for the optimal timing to apply patches. Nonetheless, as research such as Olswang et al. (2022) indicates, there are several policies about when and how to patch.

As a result of our findings about the timing of patching, we may offer an additional patch policy to Olswang et al. study on patch policies from 2022. "Wait for others, then immediately patch." In our examination of the relevant literature, we did not come across any patching policies that resembled this one. The only recognizable option is "time-based patch policy," which has the same benefits as waiting to see whether other businesses have difficulties if the timing is correct. However, if the patch is issued near to the intended "period" for patching, the advantages would be diminished.

5.2 Patch meetings avoids mistakes

Patch meetings are, according to our findings, a "better practice" for overcoming many of the obstacles we have raised in this thesis. Even though the majority of the smaller businesses in in this survey did not conduct patch meetings on a regular basis, they did so in exceptional circumstances if the vulnerability was of a severe nature.

Li et al. (2019) article that raises the possibility of faulty patches being applied or the vendor sending incompatible patches. Patch meetings have the potential to decrease this risk for the business, since personnel with a comprehensive understanding of the system can express their thoughts and follow the philosophy of "more eyes to find the problem"

Communication issues inside the business are the obstacle that patch meetings aid in mitigating the most. Since it provides employees with a forum for dialogue, and if top management is in attendance, they are also kept in the loop. Patch meetings are an effective solution to the problem Dissanayake et al. (2021) Particularly when the system being patched is either a legacy system, a particularly complicated system, or just a tough system to patch. Therefore, it is of tremendous assistance to have individuals who are knowledgeable about both the system and the vulnerability being fixed.

As the participants were more motivated to use patch meetings to avoid system errors and compatibility errors, and it is logical that a patch meeting would reduce the errors generated when applying a patch, we conclude that this is because conducting a patch meeting requires significantly more planning than regular meetings, and as a result, fewer errors are generated.

In addition, when a patch meeting is conducted, the C-level executives are better aware of the decisions taken. As meetings are a common type of routine in the business world, a meeting report is required, so the C-level executive can decide to do something differently or give the patch-team the green light.

When confronted with a major vulnerability, such as Log4j(CVE-2021-44228), one attendee underlined the value of patch meetings. Even though communication with CERT is uncommon among smaller enterprises, they were lucky to have a CERT that aided them in cases of serious vulnerabilities. In many instances, smaller businesses must contact someone from outside the business to address the vulnerability.

5.3 Communication and C-level decisions

The answer to the question "Do we have the expertise to repair this?" is often determined by a C-level executive, and it might be difficult to provide an accurate response. If this subject had been brought at a patch meeting, it would have been beneficial to address it in order to get a quicker, more accurate resolution. However, none of the participants believed that C-level executive approval for any activity related the patching process was a cause of issues or delays. This contradicts previous study by Dissanayake et al. (2021), which highlighted leadership approval as a factor of delay. This may be because our research selection was less confidential than that of Dissanayake et al. (2021) health-sector subjects. The participants in our thesis also reported that, contrary to the findings of Dissanayake et al. (2021), C-level decisions were always taken swiftly in circumstances when serious high-risk vulnerabilities were rectified.

5.4 Patch process in businesses

The patch process is one that may look extremely different from one business to the next, and this can be attributed to the fact that each organization's computer systems and industry of operation are unique. In addition, as we have discussed previously on patch prioritizing, risk is another element that might play a role in determining a course of action.

5.4.1 Vulnerability retrieval & Patch information retrieval

We received a variety of replies from the participants. However, it represented the current reality in which cybersecurity companies that work with businesses give priority to larger businesses and notify smaller businesses about vulnerabilities later. As the participants in this study were from industrial sectors with customized industrial control systems (ICS), notifications about a vulnerability are not sent automatically via email, but rather are sent manually via email. This may be less frequent for businesses with more common software and IT-systems. Therefore, the industry and size of the business explain the delay as smaller businesses are probably a lesser priority then the larger customers for the cyber security businesses.

One may argue that cyber security firms are wrong to set priorities for the smaller businesses. Nonetheless, it is an understandable occurrence, and because the risk and repercussions for smaller businesses are often lesser, it is not an issue that has to be addressed, but something that could be significantly enhanced.

The patch information retrieval portion of the procedure also differs according on industry and business type. With the exception of high-severity vulnerabilities, many firms who collaborate with a cybersecurity-specialized firm were not informed of most vulnerabilities until a patch was available. However, the participants stated that many employees with security-related roles inside the business received newsletters from external sources, such as FireEye, detailing vulnerabilities, and patches. Which demonstrates that the cybersecurity culture is beneficial for workers with cybersecurity-related obligations, but it should not be something they feel compelled to perform on their own time, and if it is a mandate, it should fall within their responsibilities. According to our findings, businesses in the industrial sector believed they needed an elevated level of cyber security, but the patching process was difficult, time-consuming, and frequently incompatible with other software.

5.4.2 Patch testing & Patch prioritization

When discussing patch prioritization, NIST describes the various risks associated with deploying patches to a system.

We addressed these risks while discussing patch prioritization with companies. One of the things we asked all interviewees was whether they use a testing environment for new patches, and all of them responded affirmatively, although in various ways. Some businesses utilized a virtual machine (VM) to test the new patches, while others decided to test on a subset of personnel over a brief period of time.

This indicates that many firms are concerned about patches causing problems with their present systems. However, when we questioned if there had ever been significant problems with their patches, everyone said that it had never been a problem since they always had backups to revert to.

As most businesses reported using a testing environment, it is in our understanding that this was the only way that they did any patch testing. As a result of the SolarWinds case, we hypothesized that more businesses would conduct more thorough testing by examining source code on accessible patches. However, when questioned, the most frequent responses were that there was neither sufficient time nor sufficient information to know what to look for.

When discussing critical patches, we received differing responses from various businesses. Because we wanted to determine how significant a vulnerability must be for it to be given a high priority. As the most recent significant vulnerability, the Log4j vulnerability was the topic brought up by most of the businesses. It was noteworthy that some businesses claimed they quickly arranged a meeting to discuss how to patch their systems, even though it was a late Friday, while others indicated they would postpone the process until they receive advice from their partners. Clearly, the scale of the business and, more crucially, the amount of information they possessed impacted how soon they would solve the problem.

5.5 Planned delay in businesses

A significant portion of our thesis is devoted to investigating why organizations opt to delay system patching. Since that is the major issue of Dissanayake et al. (2021) work, and as mentioned in the article, it was the first research report on patch management coordination. As a result, we were highly interested in expanding our understanding of patch delays when we questioned the participants of our interviews. Based on our research, we have determined that various companies have vastly diverse patching policies. From what we have discovered, many businesses claim to have a policy of immediate patch. However, we learnt through the interviews that this is not always the case.

What was common for all businesses was that Microsoft patches were very quickly implemented, because of the statement that "Microsoft knows security better than us." It might take anything from an instant to a week to deploy these security patches.

A prominent factor stated by several interviewees for delaying patches was the necessity to schedule system patching around uptime. Participants informed us that, to maintain a high uptime on their systems, they would frequently schedule patch implementations throughout the night, when the requirement for uptime was less crucial.

Several participants stated that in order to maintain a 100 percent uptime, they would run multiple instances on virtual machines to verify that patches were functioning as planned. This might cause delays, as the testing step would need many days.

We anticipated that a number of participants would tell us that business management caused delays in patching, but we were informed that this was rarely an issue because the security team had largely unrestricted control over patching, it would often require a system upgrade for management to be involved.

5.5.1 Human delays

The most common reason for the delay of patches, as stated by Dissanayake et al. (2022), is related to human factors. During our interviews, we received comments that were quite similar to this, and we were told that because patches are not applied automatically, there will be times when the patching team is unable to release the patch, which would result in a delay in its implementation.

5.5.2 Technical delays

Because some of the IT systems of the interviewees were controlled by a server provider, a new level of communication has formed. Consequently, companies lost direct access to their systems and were forced to rely on the service provider for modifications. This meant that if urgent alterations were necessary and the supplier was absent, there may be complications. This might generate significant problems for businesses who employ similar technology solutions. Another disadvantage of this method is that you must rely on the provider to make the necessary patches or revisions.

5.5.3 Patch load

As noted in the findings, patch load occurs when the frequency with which patches are provided exceeds the business's capacity to deploy them. This phenomenon is especially prevalent during the holiday season, which much of the western world considers to be one of the most stressful periods of the year. Since attackers are aware that companies are most susceptible and preoccupied during this time, attacks are more prevalent during this time. This growth affects every area and industry, from private persons to corporations and government agencies.

This delay is due to a mix of trend, lack of preparation, and lack of error margin. This is a choice for which management would need to consider in advance, and there should be a debate regarding the risk allocation involved with postponing a patch during the Christmas break.

5.6 Legacy systems are not secure

The participants provided significant insights on the general condition of ICS systems. As there would always be legacy systems in the vast majority of businesses, it was logical to assume that many ICS systems contained legacy systems. However, the most intriguing conclusion was that most participants believed their legacy systems to be secure, while having a healthy and positive attitude regarding cyber security. As many legacy systems remain offline and cybersecurity is constructed around them, the systems are protected by "castle-and-moat-security."

The usage of "Castle-and-moat" security is no longer recommended as a best practice, since if there is an attack and the attacker gains access to the system, the door is open, and the attacker may freely roam around within the system. The participants did express worry over an out-of-date security strategy, but this was not a current priority because digitization is an ongoing process. The issues that we noticed were:

- Legacy patches were released irregularly, not possible to predict when a patch was released.
- Legacy patches were unique, the routines that were used on regular patching were not applicable.

The irregularity problem is related to business planning. In addition, the issued patches for legacy systems are a collection/bulk of vulnerabilities whose release is triggered by a major vulnerability. Thus, necessitating interruption of regular business operations, postponement of present initiatives, and the need to establish priorities.

The difficulty with uniqueness is that the personnel who generally patch the systems must learn knowledge of legacy systems, a better understanding of how the patch is constructed, and knowledge of how to implement the patch. This does not account for the actual patch implementation, which will likely take longer due to the likelihood that it will be performed manually. All these issues make legacy system patching far more complicated, time-consuming, and resource intensive.

However, there is no apparent answer to these two issues, as they are unavoidable necessities that cannot be circumvented without digitalizing the existing systems. The sole option is to limit the business impact by incorporating a sufficient margin of error when planning and managing patching, so that there is sufficient capacity to apply a legacy patch without disrupting typical patching schedules. These two issues might be a reason to seek an alternative patching policy.

5.6.1 Legacy patches not being compatible

The incompatibility of the patches with the system's newer components was an additional difficulty encountered by the participants when fixing outdated systems. This prevented the participants from patching, since it would have required too much work.

This issue stems in part from the fact that legacy patches are delivered sporadically, but also from the fact that the patches that are released are bulk patches including several patches for vulnerabilities. Theoretically, as it is a bulk-patch, it should boost the desire and need to install the patch immediately; nevertheless, the participants stated that they did not do so and instead depended on their security infrastructure. Over time, this would likely make the old system a greater risk and reduce the business's overall security.

The participants have reached this conclusion because implementing the patch would need too many resources. When viewed from the outside and in the context of the entire situation, this choice appears to be the consequence of bad planning and management. When the cost of workers, time, and schedule leaves little margin for error to debug and install incompatible patches, not finding other solutions may appear to be delaying the inevitable. Thus, allowing the status of the legacy system to gradually deteriorate to a point where the business is obliged to modernize the system.

5.7 Practical applications

The practical implications of the discussion and conclusion of this thesis might be that managers working with system patching will have a deeper understanding of the obstacles and problems associated with patching and will thus be able to make better decisions when the time comes to patch.

In addition, what we have learned about patching legacy systems might aid future business and research choices on whether or not to digitalize a system.

We hope that this thesis will add to and raise awareness of the significance of patch meetings. Most participants in our survey did not have standards or policies for patch meetings, yet the patching team conducted them on a daily basis. We hope that firms would either develop rules to increase the efficacy of these meetings or incorporate them into current frameworks and procedures.

5.8 Extended patch process model

Figure 7 is an expanded version of the patching process model presented in Figure 5 by Dissanayake et al. (2021). In this expanded model, we have incorporated discoveries that we considered warranted inclusion. We have added the vulnerability disclosure, as this is the first step in the patching process, as no new patches will be released until a vulnerability has been revealed. The decision section has also been introduced following "Patch information retrieval." This is a decision on the severity of the vulnerability and the appropriate response. Log4j was the example given by several of the participants. This was a critical vulnerability, thus under our model we would opt to hold an emergency patch meeting. This approach elects to bypass some of the typical steps of a patching procedure, as it is imperative that the patch be installed as quickly as feasible. Specifically, vulnerability + scanning, evaluation and prioritization, and patch testing are omitted.

We have also introduced an additional decision following the testing phase. Due to the fact that certain patches are not implemented for various reasons, it is logical that there be an option to reject a patch.

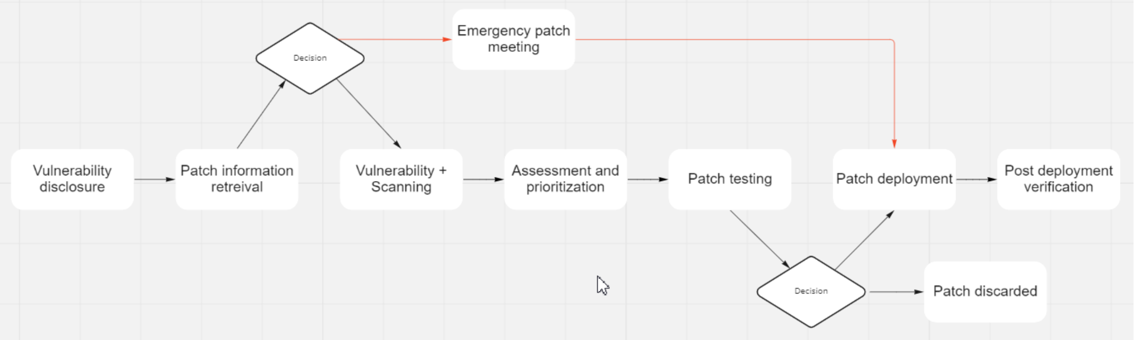


Figure 7 Our model of an extended patch process

5.9 Limitations

As with other scientific studies, it is essential to be aware of the limitations of the study. In order for future research to avoid or alleviate these restrictions, this is necessary. The most significant constraint of our thesis is the scarcity of existing research and frameworks in the subject of patch management, as this is not as highly prioritized as the patching process itself. This resulted in us doing a limited literature review in which a few essential publications had a greater impact than other studies, but also led us to depend more on our own results than on previous literature. This restricts our study in that it lacks viewpoints in some areas. Moreover, if the literature on which we based our study were extremely prejudiced towards the patching procedure, this research would also become shrewd.

A further limitation of our thesis is that many of the people with whom we contacted were unwilling to discuss their patching or cybersecurity practices. This is a frequent answer when conducting research on cybersecurity since many do not prioritize it sufficiently. Additionally, we experienced difficulty receiving responses from businesses. 72 emails were sent to various businesses, but only 16 responded, and of those 16, only nine agreed to an interview.

Additionally, as this thesis is expected to be produced in a semester, time is a constraint, as a few months are insufficient to do extensive research. Therefore, if given additional time, this topic might potentially include more comprehensive material.

6 CONCLUSION

This chapter will present the results from the findings and discussion chapter. From this thesis we wanted to answer the following questions:

- What challenges can be found in the patching process?
- How can businesses overcome the patching challenges?

To address these questions, we must refer to previous research and our own findings. Both research questions are difficult and cannot be answered simply, but the problems may be broken down into several challenges, each of which has a suggested solution.

To limit the scope of the research, we will only present answers to a subset of the unique challenges that we identified in our findings, as well as a subset of the challenges that bring further information to prior research.

Some of the challenges are technological in nature, while others are more business oriented. One of these challenges is communication with software vendors as well as communication inside the business. This was stated by several participants in our interviews. As this might result in complications if patches are not successfully conveyed, businesses that receive regular patches from their suppliers should work to establish a clear communication channel with them through an agreement.

Regarding legacy systems is a further difficulty in the topic of patching. This was a common topic, as many of the participants had legacy systems that were essential. The greatest difficulty with these systems is that relatively few patches are being developed for them. Moreover, according to the participants, these patches rarely work since they must be adapted to the unique system. Instead of prioritizing the patching of these systems, everyone resorted to using additional layers of security from outside the legacy system itself.

Wait to patch - strategy

Waiting for others to patch is a realistic choice for business with insufficient personnel to manage complex patching operations. When implementing this plan, it is crucial to keep in mind that delaying the patch puts the business at risk. Therefore, when patching can no longer be delayed, a detailed risk analysis of the business must be done.

Patch meeting

Numerous patching concerns include communication, patch prioritization, resource allocation, employee competency, misconfigurations, and complicated patches. According to the literature and our findings, a patch meeting might alleviate many of these challenges.

Patch meetings are a useful practice for both small and large businesses, which are listed in table 6. The greatest benefit of a patch meeting is that meetings help with tackling tough projects, whereas the purpose of a patch meeting is to patch a difficult project. These are the topics that patch meetings help with:

Table 6 Topics that help in patch meetings

Communication	C-level executive - Patch team communication
Higher chance to implement correctly	Less chance of misconfigurations
Less unplanned delays	

Ideally, a C-level executive or a manager should be present when holding the patch meeting. This is to guarantee that if the patch will have any implications on everyday business operations, the executive may either lessen the interruption or defer the patch. Patch meetings reduce the likelihood of encountering problems with the patch that cannot be resolved quickly. Since there will be more individuals with distinctive experiences and abilities. With extra eyes on the patching procedure, the likelihood of misconfigurations and unanticipated delays may be reduced. This is due to the patch meeting policy, which forces staff to actively plan for the patching process, rather than simply obtaining the patch and dealing with difficulties as they arise. Instead, they must prepare in advance, which will naturally result in fewer errors and hence fewer unanticipated delays. Because it mitigates so many of the challenges patch meetings confront, patch meetings are seen to be an excellent practice.

6.1 Future research

As stated previously, there is insufficient study on the issue of patching management, and this thesis merely expands the body of knowledge on a restricted number of findings. Many additional aspects of patching, in general, require further investigation. We consider the lack of study on the socio-technical element of patching to be peculiar. This comprises everything the "patching team" performs, as well as how they operate, are taught, and are managed, as well as resource management. Since much study has been conducted on socio-technical elements of IT in general, there are several linkages to the patching team's operations.

There are other intriguing aspects that deserve further investigation. The automation of patches using AI is intriguing. We anticipate that this will ultimately occur as technology continues to progress. Perhaps research should be performed to determine how AI might aid in reducing patching procedure delays.

.

7 REFERENCES

Brewster, T. (2017, September 7th) A Brief History Of Equifax Security Fails. Forbes. https://www.forbes.com/sites/thomasbrewster/2017/09/08/equifax-data-breach-history/?sh=34801c6677c0

Cavusoglu, H., Cavusoglu, H., & Zhang, J. (2008). Security patch management: Share the burden or share the damage? *Management Science*, *54*(4), 657-670. https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=31629939&site=ehost-live

CVEMitre. (n.d) Glossary. Retrieved 13.04.2022 from https://www.cve.org/ResourcesSupport/Glossary

Dey, D., Lahiri, A., & Zhang, G. (2015). Optimal policies for security patch management. *INFORMS Journal on Computing*, *27*(3), 462-477.

Dietrich, C., Krombholz, K., Borgolte, K., & Fiebig, T. (2018, October). Investigating system operators' perspective on security misconfigurations. *In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1272-1289). https://dl.acm.org/doi/pdf/10.1145/3243734.3243794

Dissanayake, N., Jayatilaka, A., Zahedi, M., & Babar, M. A. (2020). Software security patch management-A systematic literature review of challenges, approaches, tools and practices. *Information and Software Technology*, 144, 106771.

Dissanayake, N., Zahedi, M., Jayatilaka, A., & Babar, M. A. (2021, August). A grounded theory of the role of coordination in software security patch management. *In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 793-805).

Dissanayake, N., Zahedi, M., Jayatilaka, A., & Babar, M. A. (2022). *Why, How and Where of Delays in Software Security Patch Management: An Empirical Investigation in the Healthcare Sector*. arXiv preprint arXiv:2202.09016.

Epic.org. (n.d.,) *Equifax Data Breach*. Collected 01.02.2022 from https://archive.epic.org/privacy/data-breach/equifax/

Grant, M. J., & Booth, A. (2009). A typology of reviews: an analysis of 14 review types and associated methodologies. Health information & libraries journal, 26(2), 91-108. https://doi.org/10.1111/j.1471-1842.2009.00848.x

Huang, H., Baset, S., Tang, C., Gupta, A., Sudhan, K. M., Feroze, F., ... & Ravichandran, S. (2012, April). Patch management automation for enterprise cloud. In 2012 IEEE Network Operations and Management Symposium (pp. 691-705). IEEE. DOI: 10.1109/NOMS.2012.6211988

Huberman, M., & Miles, M. B. (2002). The qualitative researcher's companion. Sage Publishing.

Igure, V. M., Laughter, S. A., & Williams, R. D. (2006). Security issues in SCADA networks. *Computers & security, 25*(7), 498-506. https://doi.org/10.1016/j.cose.2006.03.001

Korman, M., Välja, M., Björkman, G., Ekstedt, M., Vernotte, A., & Lagerström, R. (2017, April). Analyzing the effectiveness of attack countermeasures in a scada system. *In Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids* (pp. 73-78). https://doi.org/10.1145/3055386.3055393

Li, F., & Paxson, V. (2017, October). A large-scale empirical study of security patches. *In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2201-2215). https://doi.org/10.1145/3133956.3134072

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*. In Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019) (pp. 273-288).

Microsoft Security Response Center. (n.d.) Security Update Guide. Collected 25.03.2022 from https://msrc.microsoft.com/update-guide/vulnerability)

Neto, N. N., Madnick, S., de Paula, A. M. G. & Borges, N. M. (March 2020) A Case Study of the Capital One Data Breach (Revised). Working Paper CISL# 2020-16

Nicastro, F. M. (2003). Security patch management. Information Systems Security. Perspect., 12(5), 5-18. https://doi.org/10.1201/1086/43808.12.5.20031101/78486.2

NIST. (n.d) Risk. Collected 01.06.2022 from: https://csrc.nist.gov/glossary/term/risk

Okoli, C., & Schabram, K. (2010). A guide to conducting a systematic literature review of information systems research. Collected from https://www.researchgate.net/publication/228276975_A_Guide_to_Conducting_a_Systematic_Literature_Review_of_Information_Systems_Research

Oladimeji, S. & Kerner, S.M. (2021, June 16th). SolarWinds hack explained: Everything you need to know. TechTarget. https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

Olswang, A., Gonda, T., Puzis, R., Shani, G., Shapira, B., & Tractinsky, N. (2022). Prioritizing vulnerability patches in large networks. Expert Systems with Applications, 116467. https://doi.org/10.1016/j.eswa.2021.116467

Saieva, A., & Kaiser, G. (2020). Binary quilting to generate patched executables without compilation. *In Proceedings of the 2020 ACM Workshop on Forming an Ecosystem Around Software Transformation* (pp. 3-8). DOI: 10.1145/3411502.3418424

Sonicwall. (2022). 2022 SonicWall Cyber Threat Report. https://www.infopoint-security.de/media/2022-sonicwall-cyber-threat-report.pdf

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

Wilson, J. (2014) Essentials of Business Research. (2nd ed.) Sage Publishing.

World economic forum. (2018) *The Global Risks Report 2018 13th Edition*. https://www3.weforum.org/docs/WEF_GRR18_Report.pdf

8 APPENDIX

Appendix A - Interview guide

Interview guide themes	
1-2	Setting ground on role & responsibility, team & business
3	Outsourcing
4	Communication
5	vulnerability/ assets tool
6	CVE
7	Vulnerability analysis
8	Framework, guidelines or policies
9	Patch prioritization
10-11	Patch decisions
11	Testing
12	Automation
13-15	Patching with partners/suppliers
16	Legacy patching

Q1. What is your role in the business?

Q2. What is your role within the patching process?

Q2.1 How many people are involved in the process?
Q2.2 How many works with security in your business?
Q2.3 How many/if any have a focus/works with security when a patch is applied to the system?

Q3. If security is outsourced, are you confident that they continually check if there are any security patches for any of the applications in your system?

Q4. When a security patch is implemented, can you walk us through how the communication is done? Who talks to who? What department is in charge?

Q5. Does your business have a software or tool that keeps track of available patches for the applications/services that your business uses?

Q5.1 Does your business have a tool that keeps track of assets and checks for vulnerabilities that those assets are affected by?

Q5.2 If yes, how does this work for you? If not, would that be interesting for the business?

Q6. Is there anyone in your business that looks at vulnerabilities that are discovered? (Common Vulnerabilities and Exposures, CVE's)

Q7. Does your business continually do vulnerability analysis? And are those taken into consideration when patches are applied?

Q8. Do your business follow any frameworks to how to patch, prioritized patches or something connecting a framework to the process?

Q8.1 If yes, how does that work for you? If not, would that be interesting for the business?

Q8.2 Are there any guidelines, policies regarding patching?

Q8.2.1 If yes, how do these affect the process?

Q.8.2.1.1 Could you think of a policy that would help the patching process.

Q9. Does it happen that there are too many patches?

Q9.1 How are they prioritized? By risk? By how fast they are to deploy/ how easy to deploy?

Q9.2 Does it happen that a vulnerability is a great risk, but the patch will be too complicated/too hard to implement?

Q9.3 When a security patch is available for one of your applications, can you come up with any reasons why it shouldn't be implemented ASAP?

Q9.4 Do you have any systems that need to be operational at all times?

Q9.4.1 How do they get patched?

Q10. Who decides if a patch should be implemented?

Q10.1Do you conduct patch meetings? Meetings with the people involved to make the process as efficient and effective as possible?

Q10.2Does management ever overrule if a patch should not be implemented?

Q11. Does it happen that a patch gets postponed due to it interrupting another project?

Q11.1If yes, what are the requirements for it being postponed? meaning how important does the other project have to be?

Q12. Does your business have a testing environment before deploying the patch?

Q13. Does your business automate some of the processes regarding patching?
Q13.1Do you believe that automation in patching is something to strive for?

Q14. Does it happen that a patch disrupts another project due to bad communication.

Q15. How well is the communication with the businesses that deliver the patch?

Q16. Does your business have legacy systems that do not get security updates anymore?
Q16.1Are any of them causing any problems as a dependency when newer parts of the system get an update?
Q16.2Has legacy dependency caused any patches on the newer system not be able to update?

Appendix B – Consent form

Vil du delta i forskningsprosjektet

"Security patching and management decisions"

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å utforske rundt temaet sikkerhetspatching, og hvilke valg som blir tatt i forbindelse med dette. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Vårt formål med denne oppgaven er å utforske om sikkerhet i bedrifter blir prioritert, og hvilke beslutninger som blir tatt rundt dette temaet. Noen av problemstillingene vi ønsker å utforske er: om forretningsbeslutninger kan påvirke den generelle sikkerheten til bedrifter, hvordan patcher blir prioritert og om forretningsbeslutninger påvirker hvor effektivt nye patcher kan bli implementert.

Dette er en masteroppgave som blir skrevet av to studenter, Daniel og Jørgen.

Hvem er ansvarlig for forskningsprosjektet?

Oppgaven er utformet i samarbeid av vår masterveileder i prosjektet Marko Ilmari Niemimaa som er førsteamanuensis ved Universitet i Agder.

Hvorfor får du spørsmål om å delta?

Masteroppgaven har behov for 5-15 respondenter med ulik tilknytning og erfaring opp mot forskjellige fagfelt som har relevant tilknytning til standardiserte rammeverk. Det kan være relevant med både teknikere, undervisningspersonell og evt. ledere for et bredt og mangfoldig perspektiv på fagfeltet.

Hva innebærer det for deg å delta?

Som følger av restriksjoner og lover knyttet til Coronaviruset, vil intervjuene bli gjennomført digitalt, via video- og ringetjenesten Zoom.

For å levere på normert tid må intervjuene gjennomføres innen 30. april i år, og gjerne før. Hvert intervju er av omlag en times varighet. Intervjuene er semi-strukturelle, som betyr at det er forberedte spørsmål, men må man være beredt på at oppfølgingsspørsmål kan forekomme for ytterligere utredning

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Dataene som blir samlet inn har ingen interesse eller intensjon om å identifisere deltakere i
forskningsprosjektet. I intervjuene kommer det til å bli tatt i bruk diktafon og eller digitalt lydopptak
via screen recording for transkribering. Dataene som blir innhentet vil bli analysert og presentert i
masteroppgaven, disse dataene vil da være i en anonymisert fremstilling, hvorav alt innhentet rådata
via screen recording og eller via diktafon vil bli slettet. All innsamling og behandling av intervjudata
vil være i tråd med retningslinjene til UiA og Norsk senter for forskningsdata (NSD).

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger
Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- Dataene som blir behandlet i denne studien kun være tilgjengelige for prosjektansvarlig, Marko Ilmari Niemimaa, samt oss som studenter og databehandlingstjenesten Zoom.
- Videre er det også iverksatt tiltak for å sikre at ikke uvedkommende får tilgang til data under og etter intervjuene:
  -Møtelenke vil ikke deles åpent.
  -Møtet vil være passord beskyttet.
  -Det vil bli benyttet lobby/venterom for å slippe inn riktige personer i møtet.

-Under behandlingen av vil dataene bli lagret på et SD-kort på en diktafon fra det. digitale intervjuet. Så fort transkriberingen er ferdig vil dataene bli omgjort til datafunn "koder" og rådataene fra intervjuet vil bli slettet i form av at SD-kortet blir destruert.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?
Opplysningene anonymiseres når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er 10. juni 2022. All rådata vil bli destruert underveis i prosjektet, videre skal annen data fremstilt i prosjektet anonymiseres innen prosjektslutt.

Hva gir oss rett til å behandle personopplysninger om deg?
Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag Universitetet i Agder har Personverntjenester vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter
Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med:

- Masterstudent, Daniel W. Reisæter ved Universitet i Agder, kontaktes på (danitr17@uia.no) og eller på telefon: +41376459
- Masterstudent, Jørgen B. Lefdal ved Universitet i Agder, kontaktes på (Jorgel17@uia.no) og eller på telefon: +4790102103
- Førsteamanuensis Marko Ilmari Niemimaa ved Universitet i Agder, kontaktes på (marko.niemimaa@uia.no) og eller på telefon: +4738141842

- Vårt personvernombud: Johanne Warberg Lavold ved Universitet i Agder, kontaktes på (johanne.lavold@uia.no) og eller på +4738141328

Hvis du har spørsmål knyttet til Personverntjenester sin vurdering av prosjektet, kan du ta kontakt med:

- Personverntjenester på epost (personverntjenester@sikt.no) eller på telefon: 53 21 15 00.

Med vennlig hilsen

Marko Ilmari Niemimaa	Daniel Reisæter og Jørgen Lefdal.
(Forsker/veileder)	(Masterstudenter)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet og har fått anledning til å stille spørsmål. Jeg samtykker til:

□ å delta på intervju med studenter.

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

----

(Signert av prosjektdeltaker, dato)

Appendix C – Email sent to interview subjects

Hei!

Vi sender denne meldingen fordi det forhåpentligvis er relevant å kontakte dere, i forbindelse med et forskningsprosjekt ved Universitetet i Agder.

Vi er en gruppe på to som studerer Cybersikkerhet med spesialisering på ledelse. Vi ønsker å utforske de forskjellige valgene som blir gjort i forbindelse med oppdatering av forskjellige deler av server og system applikasjoner i bedrifter, såkalt patching. Hvordan de beslutningene kan ha noe å si på sikkerheten og driften til systemet. Dette er hovedsakelig rundt det organisatoriske rundt temaet, og ikke like mye om det tekniske. Og i den anledningen så håper vi at dere har kunnskap som dere har lyst å dele om det temaet.

Målet med prosjektet er å stadfeste hvordan man kan komme frem til best løsninger for patching i bedrifter, med høyest mulig IT-sikkerhet og ulemper for bedriften.

Det hadde vært gledelig å ha et intervju med dere når det passer. Intervjuet vil bli over Zoom eller Teams. Selve intervjuet tar mellom 30-50 minutter, og all data vil selvfølgelig bli helt anonymisert. Dere vil også få et informasjonsskriv som er mer i dybden på hvilke data vi samler og hvordan vi bruker det. Før vi kan ordentlig komme i gang med intervjuet, må vi ha deres signatur og samtykke.

Håper på svar!

Mvh Jørgen og Daniel