UiA University of Agder

Master

A Comprehensive Framework for Patching and Vulnerability Management in Enterprises

An Exploratory Study of How Enterprises Facilitate Patching and Vulnerability Management

Authors

GUSTAV MARTIN KVILHAUG MAGNUSSEN

MATHIAS PETTERSEN

Supervisor

Associate Professor, Marko Ilmari Niemimaa, University of Agder

University of Agder, 2023

Faculty of Engineering and Science

Obligatorisk gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	Vi erklærer herved at vår besvarelse er vårt eget arbeid, og at vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	Ja
2.	**Vi erklærer videre at denne besvarelsen:** • Ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. • Ikke refererer til andres arbeid uten at det er oppgitt. • Ikke refererer til eget tidligere arbeid uten at det er oppgitt. • Har alle referansene oppgitt i litteraturlisten. • Ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	Ja
3.	Vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	Ja
4.	Vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.	Ja
5.	Vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.	Ja
6.	Vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.	Ja
7.	Vi har i flertall blitt enige om at innsatsen innad i gruppen er merkbart forskjellig og ønsker dermed å vurderes individuelt. Ordinært vurderes alle deltakere i prosjektet samlet.	Nei

Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).
Oppgaver som er unntatt offentlighet eller taushetsbelagt/konfidensiell vil ikke bli publisert.

Vi gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:	Ja
Er oppgaven båndlagt (konfidensiell)?	Nei
Er oppgaven unntatt offentlighet?	Nei

Acknowledgement

Firstly we would like to thank our thesis supervisor Associate Professor Marko Ilmari Niemimaa of the Department of Information Systems at the University of Agder. Through our bi-weekly meetings and over email, we were provided with valuable insight and feedback on how to structure our framework, thesis, and schedule. Without his continuous support throughout the semester, we would have been unable to present this thesis. Secondly, we acknowledge the anonymous organizations and the anonymous interview subjects. Their expertise aided the development of the final framework through helpful insight into the patch and vulnerability management processes and feedback on the conceptual framework, which was not available in the literature. Thank you for taking the time out of your busy day to talk to us. We also thank other professors and co-students who found time to give us feedback on our framework or discuss our work.

I, Gustav, would like to thank my parents and friends for their continuous support through the master's thesis process and overall studies.

I, Mathias, would like to thank my parents and my significant other for their unconditional support throughout the master's thesis. The thesis would not be possible without your help.

Kristiansand,
June 2023

Gustav Martin K. Magnussen
Gustav Martin Kvilhaug Magnussen

Mathias Pettersen
Mathias Pettersen

Abstract

As patching and vulnerability management have become a larger part of an organization's routine, its need for proper integration and complexity toward systems has increased. Threat actors continuously seek to develop and perform attacks exploiting vulnerabilities within systems, meaning organizations face the challenge of timely implementing patches to protect their assets. The master's thesis aims at gathering extensive information regarding patching and vulnerability management by integrating a semi-systematic literature review (SSLR), a semi-structured qualitative interview process, and our sense-making. These research methods collect insights from the existing theory and professionals' opinions. The SSLR allowed for gathering relevant studies and sense-making, which were subsequently utilized in developing a conceptual model depicting the vital processes and procedures of patching and vulnerability management based on the theory. As such, the conceptual model was showcased within the semi-structured qualitative interviews, which allowed for unbounded discussions regarding the practices, implementations, and expert input toward the conceptual framework and its improvement areas. The interviews and selection of interviewees allowed for several viewpoints and a wide perspective. Subsequently, after synthesizing the findings from the interviews and additionally gathered theory, the comprehensive framework, which aims to refine and extend the conceptual framework, was developed. The comprehensive framework aims at depicting the enterprises' collective patching and vulnerability management process, along with the intersection of the existing theory. Correspondingly, the framework could be utilized by enterprises to either improve their processes or for enterprises to implement absent processes. The findings highlight a major diversity in the implementation and execution of patching and vulnerability management. Larger companies tend to have more mature processes and employ more automation within their collection of vulnerability information and deployment of patches. Conversely, smaller companies lack the resources allocated to perform needed tasks, which results in a less organized and effective process. The research findings subsidize the existing research gap related to a lack of frameworks depicting the interrelation between patching and vulnerability management and how enterprises currently perform these processes. Additionally, it provides a substantially valuable resource for practitioners, researchers, and enterprises wishing to improve their processes based on an exploratory study assessing the existing literature, experts' opinions, and the design of the conceptual and comprehensive framework. As the comprehensive framework aims to provide a generalized approach and implementation, it can be employed by different-sized businesses while tailored to their needs.

Contents

Acknowledgement ii

Abstract iii

List of Figures vi

List of Tables vii

Acronyms viii

1 Introduction 1

1.1 Rationale and Motivation . . . 2

1.2 Thesis Overview . . . 3

2 Background and Related Work 4

2.1 Patch Management . . . 4

2.1.1 Five phases of Patch Management . . . 4

2.1.2 Patching Frameworks . . . 9

2.1.3 National Guidelines . . . 10

2.1.4 Automation in Patching . . . 10

2.2 Vulnerability Management . . . 11

2.2.1 Vulnerability Prioritization . . . 13

2.2.2 The Common Vulnerability Scoring System . . . 13

2.2.3 Criticism of CVSS . . . 14

2.2.4 Alternative Vulnerability Scoring Systems . . . 15

2.3 Interrelated practices . . . 16

2.3.1 Information Sources . . . 16

2.3.2 The Human Aspect in Patching and Vulnerability Management . . . 17

2.3.3 Identified Obstacles . . . 17

2.3.4 Roles in Patching and Vulnerability Management . . . 18

2.4 Gaps and problems . . . 19

3 Research Approach 21

3.1 Literature Review . . . 21

3.1.1 Methodology . . . 21

3.1.2 Literature Criteria . . . 23
3.1.3 Search Process . . . 23
3.1.4 Organization of Literature . . . 24
3.1.5 Screening of the papers . . . 24
3.2 Qualitative Study . . . 29
3.2.1 Research Design . . . 31
3.2.2 Interview Subject selection . . . 34
3.3 Data Collection . . . 36
3.3.1 Semi-Structured approach . . . 37
3.3.2 Interview Transcriptions . . . 39
3.4 Analysis . . . 39
3.4.1 Verification . . . 41
3.5 Ethical Considerations . . . 41

4 Findings **43**
4.1 The Conceptual Framework . . . 43
4.2 Empirical Findings . . . 45
4.2.1 Information Sources . . . 45
4.2.2 Internal documentation and policies . . . 47
4.2.3 Patch-related Challenges . . . 50
4.2.4 Patch Testing . . . 53
4.2.5 Vulnerability Prioritization . . . 54
4.2.6 Vulnerability-related Challenges . . . 57
4.2.7 Framework-specific Input . . . 59

5 Discussion **63**
5.1 A Comprehensive Framework for Patching and Vulnerability Management in Enterprises 63
5.1.1 Strategic Level . . . 65
5.1.2 Tactical Level . . . 66
5.1.3 Operational Level . . . 69
5.2 Distilled Model . . . 71
5.3 Contributions . . . 72
5.4 Limitations and Future Work . . . 73

6 Conclusion **75**

Appendices **78**
A Interview Guide . . . 79
B Information letter and Consent Form . . . 81

Bibliography **84**

List of Figures

2.1 Patch Management Process Workflow (Huang et al., 2012) 9
2.2 CVSS Metric Groups (FIRST.org, 2021) 14
2.3 CVSS Qualitative severity rating scale (FIRST.org, 2021) 14
2.4 Sources used for discovering available updates (Li et al., 2019) 16
2.5 Cyber Intel Levels (Bautista, 2018) 19

3.1 Process of systematic literature review (Xiao & Watson, 2019) 22
3.2 Screenning procedure following the PRISMA methodology (Page et al., 2021) 26
3.3 Kallio et al. (2016)'s framework for qualitative semi-structured interview guide . . . 38
3.4 Coding categories in NVivo 12 41

4.1 A Conceptual Model for Patching and Vulnerability Management 44

5.1 A Comprehensive Framework for Patching and Vulnerability Management in Enterprises 64
5.2 The Strategic Level of the Comprehensive Framework 65
5.3 The Tactical Level of the Comprehensive Framework 67
5.4 The Operational Level of the Comprehensive Framework 70
5.5 Distilled Model 72

List of Tables

2.1 Comparison of the five phases of patch management 5

3.1 Keywords utilized in the literature review search process 24
3.2 Eligible literature review articles 26
3.3 Qualitative research methods gathered from Fujs et al. (2019) and their suitability with the research 31
3.4 The *"Seven stages of an interview inquiry"* (Kvale, 2011), and its application in the research 32
3.5 Interviewee Demographics 35
3.6 Application of Kvale (2011)'s *"Six steps of Analysis"* in the research 39

4.1 Framework-related input distilled from Section 4.2.7 62

Acronyms

CIO Chief Information Officer. 18

CISA Cybersecurity and Infrastructure Security Agency. 13

CISO Chief Information Security Officer. 18, 54

CMDB Configuration Management Database. 50, 52, 55, 56, 76

CVE Common Vulnerabilities and Exposures. 11

CVSS Common Vulnerability Scoring System. 13–15, 24, 55–57

DSU Dynamic Software Updating. 18

ENISA European Union Agency for Cybersecurity. 1

EPSS Exploit Prediction Scoring System. 15

GDPR General Data Protection Regulation. 44, 66

ICS Industrial Control System. 6, 15

IDS Intrusion Detection System. 12

ISO International Organization for Standardization. 44, 66

NIAC National Infrastructure Advisory Council. 13

NIST National Institute of Standards and Technology. 10, 13, 44, 66

NSD Norwegian Centre for Research Data. 37

NSM Norwegian National Security Authority. 1, 10, 47, 50, 66

NVD National Vulnerability Database. 11

PCI-DSS Payment Card Industry Data Security Standard. 14

1 | Introduction

In a threat landscape report from 2022, the Norwegian National Security Authority (NSM) discussed the national digital security picture in Norway. The report included findings from a series of penetration tests on Norwegian organizations. These findings describe the seven most common vulnerabilities they encountered, with *"Software Error"* as the third most common (NSM, 2022). In this vulnerability category, the most common reason for the threat is that: *"Often there will be available security updates for the software, but the organization will not have installed it yet"* (NSM, 2022, p. 14), displaying the importance of maintaining effective patching and vulnerability management and performing timely patching of organizational assets. Similarly, in European Union Agency for Cybersecurity (ENISA)'s threat landscape report, they recommend the implementation of patch management planning and an organizational push towards updating (ENISA, 2021). Patching and vulnerability management are integral parts of the overall security effort of organizations. Discovery and management of vulnerabilities through patching and accepting risk are necessary components of any organization's security effort to ensure that the organization is as resilient as possible in defiance of ever-evolving threats and experienced threat actors. However, very few frameworks present the interoperability of patching and vulnerability management. Previously developed frameworks such as Huang et al. (2012) and Souppaya and Scarfone (2022) characterize different aspects of the patching processes, while Hore et al. (2023) and Farris et al. (2018) purpose frameworks for the vulnerability management processes. Although these frameworks characterize parts of the patching and vulnerability management process, there are limited representations of the connectivity between patching and vulnerability management and different aspects of information security. Moreover, these frameworks provide a highly suitable description of the other components of patching and vulnerability management without the overall comprehensiveness that is present.

Therefore, given the need for the implementation of patch and vulnerability management, we created the following problem statement as the foundation of inquiry into how organizations implement patching and vulnerability management:

How do organizations implement patching and vulnerability management?

Based on the assumption that there is valuable insight to collect from subject matter experts working with patching and vulnerability management, we developed the following research questions from the problem statement to explore patching and vulnerability management routines and implementations in organizations:

- **RQ 1:** How are organizations facilitating patching and vulnerability management?
- **RQ 2:** How can insight from appropriate interview subjects and theory enhance the patching and vulnerability management process?

In this thesis, we present *"A Comprehensive Framework for Patching and Vulnerability Management in Enterprises"* (Figure 5.1), which seeks to summarize the over-arching processes and coordination that happens in patching and vulnerability management-related work through the combined effort of reviewed related works in literature and gathered insight from subject matter experts. This is performed through an extensive review of patch and vulnerability management-related literature and qualitative research through interviews with subjects that work with patch and vulnerability management. Based on the observations gathered from the literature, a conceptual framework was developed. After being presented with the conceptual framework in the interviews, interview subjects could criticize, comment, and suggest how the framework could be altered to align more with how patch and vulnerability management occurs in an organization. Therefore, in this manner, the feedback on the conceptual framework and insight into how patch and vulnerability management was used to develop the final comprehensive framework.

1.1 Rationale and Motivation

When protecting an organization from digital threats, patching out-of-date assets and handling vulnerabilities is essential. Therefore, organizations must ensure mechanisms are in place to detect and mitigate vulnerabilities by facilitating patching. Alternatively, as every organization will have external-facing systems, there must be patching routines and vulnerability management for residual vulnerabilities. As such, we believe research that produces systems, processes, or procedures that increase patching and vulnerability management effectiveness is definitively valuable. Moreover, to our knowledge, no developed frameworks incorporate the interconnectivity of patching and vulnerability management. Therefore, by creating a framework that integrates patching and vulnerability management, there is a definitive valuable contribution to the overall body of work. Furthermore, such a framework would support organizations with established patching and vulnerability procedures by categorizing and structuring work.

In contrast, it would support less mature organizations as a starting point for patching and vulnerability management work. Additionally, by presenting the framework, we advance the understanding of how patching and vulnerability management is facilitated in organizations and provide an easy-to-follow framework structuring the different components of patching and vulnerability management and associated parts of the security effort. This might motivate more research into effectivizing patching and vulnerability management and influence researchers to look at the interconnectivity of operational information security.

1.2 Thesis Overview

Introduction: In this chapter, an introduction of the thesis and the problem statement is presented, along with rationale and motivation.

Background and Related Work: In this chapter, the discovered background and related work to patching and vulnerability management are discussed.

Research Approach: In this chapter, the research approach for the literature review and the qualitative research are discussed.

Findings: In this chapter, the conceptual framework is presented along with the findings from the qualitative research.

Discussion: In this chapter, *"A Comprehensive Framework for Patching and Vulnerability Management in Enterprises"* is presented, and empirical findings with their application to the framework are discussed. In addition, the work's contributions, limitations, and potential future work are also stated.

Conclusion: In this chapter, a conclusion to the thesis is presented, along with the key findings related to the research questions.

2 | Background and Related Work

This chapter describes the background and related work focusing on the essential aspects of patch and vulnerability management and how organizations implement them. The literature presented in this chapter demonstrates the patch and vulnerability management processes, the most critical aspects of these processes, and the significant identified challenges by these processes. Moreover, the chapter thoroughly describes identified gaps and limitations of the research.

2.1 Patch Management

Businesses that use any software heavily rely on software updates to mitigate vulnerabilities and to keep their operations secure. Consequently, maintaining a sufficient patch management structure is crucial to structure patches and gain an overview of the systems used (Li et al., 2019). System administrators typically perform patch management that manages numerous machines either for their own company or as a service, and they are often referred to as *"keepers of the machines"*. This denotes that system administrators, and other personnel working with patching and vulnerability management, are entrusted by the company to administer the company's vital assets. As such, the patch management process is crucial for a company as failure to patch properly and organization may lead to security breaches (Li et al., 2019). Patch management is ultimately essential for an organization's assets as it prevents the exploitation of vulnerabilities related to the assets. Timely patching of a system is crucial as threat actors exploit vulnerabilities rapidly, resulting in a need for continuous deployment of relevant patches. It is used for identifying, testing, installing, and verifying the security patches relevant to the organization's systems, and is carefully considered to cover the organization's security needs (Dissanayake, Jayatilaka, et al., 2022). As patch management is a broad topic covering several areas, the section covers the core components of performing patch management. These include the phases of patching, the information sources utilized, and how national guidelines and automation influence patch management.

2.1.1 Five phases of Patch Management

Li et al. (2019), Tiefenau et al. (2020), Dissanayake, Zahedi, et al. (2022), and Serio and Gentile (2019) report on five different phases fundamentally prevalent when system administrators perform patch management, and which lay the foundation of the processes within patch management. Table 2.1 depicts each author's formulation of the five phases, where all the phases reported on by each

author are intercorrelated. In addition, the table showcases the unity of phases, where each author recognizes five different phases in the process. The table aims at showcasing each author's definition of the five phases of patching.

Table 2.1: Comparison of the five phases of patch management

Author	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 4**	**Phase 5**
Tiefenau et al. (2020)	Information	Deciding + Preparation	Testing	Installation	Post-Installation
Li et al. (2019)	Learning About Updates	Deciding to Update	Preparing for Update Installation	Deploying Updates	Handling Post-Deployment Issues
Dissanayake, Zahedi, et al. (2022)	Patch Information Retrieval	Vulnerability Scanning + Assessment and Prioritization	Patch Testing	Patch Deployment	Post-Deployment Patch Verification
Serio and Gentile (2019)	Information Gathering and Project Planning	Monitoring and Evaluation	Patch Testing	Patch Deployment	Verification and Reporting

Li et al. (2019) briefly explains the different phases that are present in patch management:

- 1. Learning About Updates
- 2. Deciding to Update
- 3. Preparing for Update Installation
- 4. Deploying Updates
- 5. Handling Post-Deployment Issues

These five phases collectively depict the system administrator's process when undertaking organizational patching management. However, based on the organization, some businesses may perform patching differently due to size limitations, resource limitations, or other internal/external factors that affect the very patching routine. Nevertheless, it is essential to provide specific guidelines and a general framework on what elements to include so that businesses can apply the applicable parts that suit their operations. Following is an elaboration on the five phases:

Learning About Updates revolves around discovering the updates applicable to the systems the organization uses, along with the different sources relevant personnel use to gain an overview of the threats and patches. Discovering update material and patches relevant to a system can be performed using various information sources, where *security advisories* are among the most

utilized form of information source. A security advisory relates to frequent and reliable information regarding patches and other relevant threat information covering the most essential and prevalent threats simultaneously, as showcased in Li et al. (2019). Moreover, Tiefenau et al. (2020) and Serio and Gentile (2019) denote *Information* and *Information gathering* as the primary phases when performing patching, which correlates with previous studies and substantiates that gathering information about patches before implementing them is the primary task.

Additionally, learning about updates does not necessarily mean only using one good source of information but instead using various sources to ensure all applicable information is consumed. If a system administrator or practitioner uses several sources from different vendors, security advisories, mailing lists, and security forums, the chance of missing important information becomes significantly lower. However, following continuous information streams which deliver real-time coverage, such as system alerts, news streams, and mailing lists, ultimately requires less time and effort than actively searching for information on forums and blog posts (Li et al., 2019).

Deciding to Update: After the collection of patches and security-related information is gathered, deciding which patches to go forward with aids in finding the relevance and applicability of a patch. In this process, patch information is filtered by the system administrators and practitioners based on the systems the organization has and the urgency of implementing specific patches. Different information sources may publish several patches, both security-related and not security-related, so prioritizing the security-related patches aids in addressing the critical vulnerabilities that affect the security of the systems. In the interviews conducted by Li et al. (2019), most system administrators prioritized security-related patches contrary to non-security-related patches. Ultimately, assessing a vulnerability's risk and severity contributes to a higher chance of covering the relevant threats for the specific system.

Prioritizing patches that aid in improving the security of the software or system can be challenging as many patches are bundled, meaning one patch may contain a security fix and performance improvements. However, selecting which patches to deploy can be performed by implementing specific criteria to ensure the quality and applicability of a patch. As Serio and Gentile (2019) state, patching can be done automatically if the following criteria are applied. Otherwise, the patch cannot automatically be considered applicable and has to go through a manual review process to assess the risk and applicability of the patch:

1. The vendor/manufacturer has approved the patch
2. The patch has been designed and released for the considered asset with the current firmware/software version
3. The update is essential to resolve some vulnerabilities

Serio and Gentile (2019) report on best practices for patch management of ICS (Industrial Control Systems). However, as the patch management process is sufficiently similar to previous research, the guidelines can be applied to other aspects and industries/branches.

Preparing for Update Installation: After the relevant and applicable patches are found, categorized, and prioritized, the system needs to be prepared with specific tasks to minimize disruption and ensure a successful installation. Preparing a system for patch deployment can be performed using various tasks, but as Li et al. (2019) define the processes in correlation with interviews with system administrators, there are three over-arching categories. These categories outline the most important tasks to perform before applying the patches:

1. **Backups/snapshots:** Take a backup of the environment to roll back to in case problems arise
2. **Prepare Machines:** Alter configurations or dependencies to fit the upcoming update
3. **Testing:** Test the updates in a separate environment to identify unintended behavior and bugs

Testing is a non-trivial task that aids in finding out how the patch affects the given system and indicates what complications the patch brings. Testing an environment with a patch can be performed in numerous ways, with each approach being implemented based on the target system and the systems running. The two main approaches to testing the environment are *staggered deployments* and *dedicated testing environments*. The former relates to performing testing in stages; instead of rolling out patches to all machines in an environment simultaneously, the organization's machines are divided into three main categories: Stage 1, Stage 2, and Stage 3, where each stage represents one set of machines. As such, each stage represents admin machines, developer machines, and production machines, respectively. This approach substantiates that patches should first be implemented on less service-critical machines to assess the behavior and impact before excelling to the next stage (Li et al., 2019). However, this approach is not time-effective as a system administrator has to monitor each system's behavior over time. There may be less risk involved as non-critical machines are impacted first, but it could pose a security risk as production machines are longer exposed to potential attackers while waiting for the behavior results. Contrarily, the latter method involves creating a dedicated testing environment for testing the patches before rolling them out to a live environment. This method ensures that sufficient testing is undertaken to identify unwanted behavior and bugs (Li et al., 2019; Tiefenau et al., 2020).

Deploying Updates is a critical part as it revolves around implementing the patches that have carefully been assessed and tested previously. Based on the system, capabilities, and resources, an environment's patch can be deployed using automated tools, manual review, or a hybrid of the two methods. Deploying patches in a larger environment using automated tools reduces resources having to be spent on specific tasks and may, in some cases, enable system administrators to patch more frequently and autonomously (Li et al., 2019). As such, automated tools for updates are proposed as a solution to countermeasure delays in patching as software is kept up to date without much interaction and supervision (Tiefenau et al., 2020).

A proper patch policy ensures that the organization fully understands how to handle the patches and deployment. Dey et al. (2015) reports on five different patch deployment policies which build up the standard approach of conducting patch deployment in an enterprise:

- **One-for-one policy:** This policy refers to immediately implementing patches after they

become publicly available. For an enterprise, continuously deploying and installing all patches can ultimately be costly and work against the primary intent of having a secure environment (Dey et al., 2015). Contrarily, an enterprise would make the most out of their patching by focusing on vital areas of their systems and carefully assessing which patches are suitable for their environment (Souppaya & Scarfone, 2022).

- **Time-based policy:** Instead of patching according to each patch release, a time-based policy revolves around batching several patches together and patching at certain time intervals. This approach contributes to lower costs as an organization's systems experience reduced downtime as several patches are implemented simultaneously. Additionally, allocating resources and planning before the patch implementation is more straightforward as there is a pre-set patch date, where scheduling and planning become more efficient (Dey et al., 2015).
- **Patch-based policy:** A patch-based policy refers to implementing patches when the number of patches reaches a predetermined level. According to Dey et al. (2015), this approach could result in critical patches not being implemented timely as waiting for the number of patches reaching the threshold may take time. There is a superfluous security risk for the business, and it would also result in patching being performed randomly.
- **Total-control policy:** This policy refers to implementing patches when the collective security risk the business is willing to take reaches a certain threshold. Such an approach includes summarizing the existing patches with determined severity levels and comparing the sum to the threshold (Dey et al., 2015).
- **Emergency-control policy:** Similar to a total-control policy, an emergency-control policy is built around the severity scores of patches. However, this policy enforces patches upon the arrival of a patch that exceeds a certain security risk threshold and not the cumulative sum. Critical patches are installed immediately, reducing risk but randomizing the patch cycle (Dey et al., 2015).

The aforementioned policies are only a selection of policies an organization can implement. Since all enterprises are different, with various resources, systems, knowledge, and capabilities, a *hybrid* policy (enforcing a mixture of policies) may be beneficial as one distills the most vital aspects of each policy. For example, a system administrator or practitioner could enforce certain predetermined intervals of the patch routine (time-based policy) and only deploy and install the most critical patches that reach a certain security risk threshold (emergency-control policy).

Handling Post-Deployment Issues: After installing the patches found in the previous stages, handling post-deployment issues significantly improves the overall migration and updating of systems in an environment. As Li et al. (2019) propose in *Preparing for Update Installation*, testing is essential for mitigating troubles related to patch implementation. However, not all inconsistencies or bugs are present when testing and will only show once the patch is deployed in the live production environment.

Tiefenau et al. (2020) reports on post-deployment issues being a minor concern in live systems as there is a low frequency of problems related to the matter. However, keeping track of the deployment status of each patch can contribute to a greater overview of verifying the patches' deployment rate. Additionally, it aids in detecting various post-deployment issues before they materialize into a significant risk. Therefore, Dissanayake, Jayatilaka, et al. (2022) recommend regularly monitoring each patch implementation's status.

2.1.2 Patching Frameworks

Huang et al. (2012) propose a *Patch Management Process Workflow* framework in Figure 2.1 which depicts the over-arching processes of patch management. Following the workflow map, there are three main categories present: *Patch notification*, *Patch scheduling*, and *Patch deployment and post-dep (deployment)*. These categories each represent certain task areas that are initiated and interdependent to ensure all processes are executed correctly. However, even though the model depicts areas of responsibility and suggests how the patch management process can be completed, there is a lack of integration with higher-level management for policy creation/approval and specific processes targeted at the tactical and operational levels. Contrarily, the model clearly explains the vital steps and hierarchy of the processes, which subsequently can be utilized to develop a comprehensive framework, including vulnerability management.

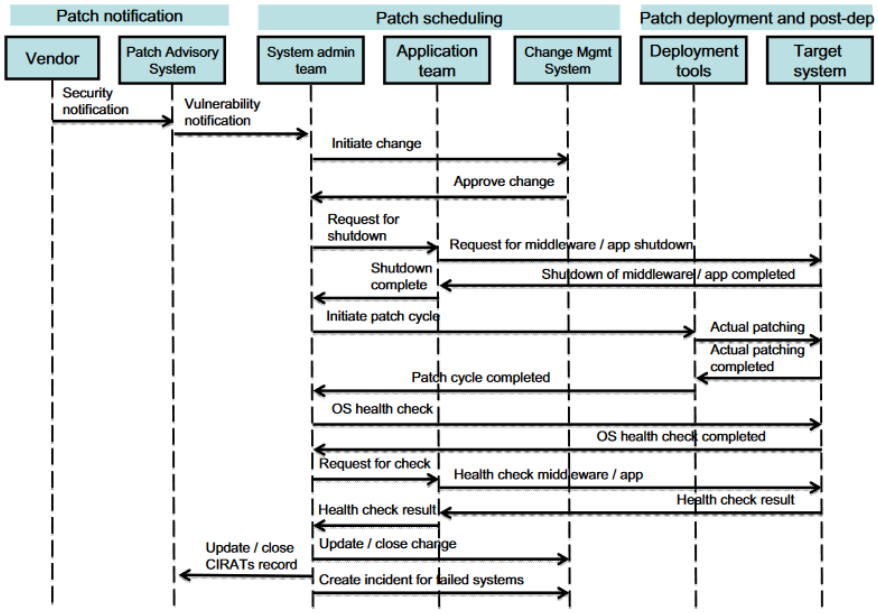

Figure 2.1: Patch Management Process Workflow (Huang et al., 2012)

Research is scarce when identifying frameworks developed for patching and vulnerability management as a unit. For example, Souppaya and Scarfone (2022) provides guidelines for enterprises on performing patch management planning but does not detail the patching processes. As the guidelines are directed at planning, further implementation strategies and vital processes are absent. Therefore, the enterprise guidelines contribute to organizations gaining an overview of what to include in the

patching and vulnerability management process but do not contribute to specific examples of how an enterprise should implement it. Similarly, (Huang et al., 2012) provides a framework for patch management that explains patch processes but not the ingrained processes and liabilities.

2.1.3 National Guidelines

National guidelines for cybersecurity play an essential part in the ecosystem of patching practices. For companies to operate within specific countries or, in some cases, do business with others operating in certain countries, they must adhere to specific laws that mandate cyber security. Therefore, there are definitive guidelines for handling cybersecurity for governmental agencies and private companies operating in specific countries. However, there are many different flavors of guidelines and differences in how these are developed in different countries. Some are closer to a set of recommendations (NSM, 2020), while others can be closer to complete frameworks.

NSM (2020) is a report created by the Norwegian National Security Authority (NSM) as guidelines on how to utilize IT assets within an organization securely. The report is directly created for Norwegian businesses and describes specific principles or actions that Norwegian companies should take regarding improving their IT security posture. Moreover, this report aims to provide an easy-to-understand quantification of the security effort required within each principle to advance businesses' security capabilities. This report is one of the department's primary cybersecurity efforts, with the following principles directly relating to patch management:

- **2.3.1:** Implement a managed centralised regime for security patching
- **3.1.1:** Perform regular vulnerability mapping
- **3.1.2:** Follow services related to vulnerability intelligence
- **3.1.3:** Use automated and centralised tools for managing known threats

Looking towards international standards, the National Institute of Standards and Technology (NIST) provides several standards for information security. Moreover, NIST also develops standards for various topics, including information technology and information security. Relevant for patch management is the NIST SP 800-40r4 report, which describes how to implement enterprise patch management planning in an enterprise. Furthermore, the report aims to motivate changes in mindset towards patching and understanding how it is a preventative measure, recognizing that there will always be problems when patching that can be dealt with by simplifying processes and automation of patching (Souppaya & Scarfone, 2022).

2.1.4 Automation in Patching

Patching a system can be performed manually and automatically, whereas an automatic system is a system that autonomously applies patches given a source of information about a service. Dissanayake et al. (2023) reports on several key aspects of the processes of patching being mostly manual, such as the vulnerability scanning and prioritization phase, the patch testing phase, and

the post-deployment phase. Having several manual tasks may limit a practitioner's ability to gather all necessary information regarding what to patch and which patches to implement, as well as it may limit the quality of the implementation and post-deployment process. Furthermore, several limitations in the current patch automation cycle, such as limited dynamic context factors in vulnerability assessments, a lack of support for handling patch dependencies, and missing information in scanning, may lead to erroneous reports. To further substantiate how automation may contribute to more effective patching and vulnerability management, the participants in Dissanayake et al. (2023) suggest implementing an automated and centralized patching platform that collects patching information and threat information. Additionally, a system that automatically provides an in-depth analysis of the potential impact of patches on the organization's systems may contribute to a more secure environment and allow for more careful integration of patches.

Xu et al. (2022) reports on implementing an automated patch system named TRACER, as industrial patch systems often use databases such as NVD and CVE for the majority of the patching information. As the research concludes, there is a concern for the quality of these vulnerability databases, whereas the main problem lies in the unreliability of the vulnerable version information in the aforementioned databases. Contrary to utilizing NVD and the CVE list, TRACER utilizes NVD, Debian, and Red Hat for its patch information and can notify NVD of missing/incomplete patches. The program is *"useful in practice for security experts to localize patches more accurately and quickly."* (Xu et al., 2022, p. 869). Implementing an automated patch searching system may aid researchers in finding more accurate and complete patches as it reviews code changes related to vulnerability patches and extracts the changes. Similarly, Baiardi and Tonelli (2021) propose an automatic program named Haruspex which creates a digital twin of the system, allowing for testing for vulnerabilities and patching them to see the behavior in the cloned system. Haruspex was ultimately tested by giving it a data set of known vulnerabilities and patches where, compared to already existing techniques, the implementation effectivized the patching process significantly.

2.2 Vulnerability Management

Bautista (2018, p. 234) describes vulnerability management as *"the capability of an organization to effectively identify, report, and reduce weaknesses in the organization"*. The objective of vulnerability management within an organization is to ensure that as many vulnerabilities as possible are detected, mitigated, and accounted for. Identifying vulnerabilities deals with discovery, where a vulnerability is present in a system, whether through automated tools or manual means. Subsequently, reporting describes the discovered vulnerability and decides what measures should be taken. Finally, mitigation represents the reduction of the found weakness associated with the vulnerability. Several frameworks have been developed to illustrate this concept, such as *Deep VULMAN* (Hore et al., 2023) and *VULCON* (Farris et al., 2018). While each framework has specific steps with distinct names, fundamentally, the process is the same. Therefore, this means that while these steps are different, they are still identifiable if the vulnerability management process is followed through a specific vulnerability.

Furthermore, another aspect of the vulnerability management process is that, for a specific vulnerability, there is a definitive start and end of management (whether this is through remediation or acceptance). In contrast, for the process as a whole, it is continuously ongoing. Therefore, on an individual vulnerability-specific level, the processes will be linear, while the vulnerability management process as a whole will always be constant. The concept of this ongoing process is illustrated by the cyclical structure of both *Deep VULMAN* (Hore et al., 2023) and *VULCON* (Farris et al., 2018).

Identify

The components mentioned above of vulnerability management are used in several different frameworks. The identification of vulnerabilities ultimately aims to detect vulnerabilities within an organization's infrastructure. Hore et al. (2023) describes this process as *Vulnerability Arrivals* through the different information sources collected from machine and organization-specific information, vulnerability reports (new and unmitigated), and Intrusion Detection System (IDS) alert logs, while Farris et al. (2018) specifies it as *Vulnerability Analysis Data Pre-processing* where information sources such as an internal list of mission-critical services, monthly Nessus scan reports and lists of un-mitigated/residual vulnerabilities are used. While the definitions of what is used within the two frameworks differ, the content and intention are similar. Both frameworks identify the need for control over organizational infrastructure, usage of vulnerability scans, and control over unmitigated or perpetual vulnerabilities.

Report

After discovering a vulnerability, the next step in the vulnerability management process is to report on the vulnerability and decide what actions should be taken. Farris et al. (2018) incorporates this in *VULCON* through the process of *Optimized Vulnerability Management, Prioritization, and Recommendation systems*, where the gathered material from the identification step is used to determine the further course of action. Similarly, Hore et al. (2023) incorporates this in *Deep VULMAN* in the *Decision-Support Component*. In essence, this phase of the vulnerability management process is where the prioritization, described further in Section 2.2.1, happens. All gathered information sources, whether internal or external, are utilized in this step to make the best-informed decision on what should be done with the vulnerability.

Mitigate

Mitigation happens after identifying vulnerabilities and determining the proper course of action. This process directly relates to the patch management process, as one mitigation method is security patches. The mitigation processes exemplified in Hore et al. (2023) as the *Vulnerability Mitigation Process*, and in the *VULCON* framework as where the output is utilized (Farris et al., 2018). Hore et al. (2023) includes the mitigation of vulnerabilities as a part of the framework, whereas Farris et al. (2018, p. 5)'s *VULCON* framework provides *"security exposure metrics and vulnerability*

management plan to managers, operators, analysts, and engineers.". Hore et al. (2023) provides a direct implementation of the mitigation process, whereas Farris et al. (2018) provides information based on the previous steps in the process to the relevant employee groups. Regarding the actual mitigation of vulnerabilities, the frameworks clearly recognize it as a step in the vulnerability management process and as the concluding process of vulnerability management.

2.2.1 Vulnerability Prioritization

An integral part of managing vulnerabilities is prioritizing which vulnerabilities take precedence. As many factors play a role in understanding how severe a vulnerability is, deciding which vulnerability should take precedence is not always straightforward. Prioritization is here used to signify which vulnerability is decided, through different processes, to be the most important to handle first and not necessarily which vulnerabilities are not handled at all. Therefore, *vulnerability prioritization* is defined as prioritizing which vulnerabilities should take precedence when deciding which vulnerabilities should be mitigated.

2.2.2 The Common Vulnerability Scoring System

One widely adopted scoring system for measuring the severity of a vulnerability is the Common Vulnerability Scoring System (CVSS). This scoring system was initially commissioned by the United States National Infrastructure Advisory Council (NIAC) in 2003 (Mell et al., 2022) and has since been utilized by several different actors in the information security space, including the National Institute of Standards and Technology (NIST) and the Cybersecurity and Infrastructure Security Agency (CISA). The current iteration of CVSS, CVSS V3.1, uses several metrics to measure a vulnerability's severity. Furthermore, these metrics account for vulnerability characteristics that directly affect how severe possible exploitation can be on a system and are divided into the three metric groupings; *Constant/Base*, *Temporal*, and *Environmental* (FIRST.org, 2021).

- **The Base/Constant** metric groups represent a vulnerability's constant attributes regardless of time or affected systems. This category of metrics is divided into two subcategories; *Exploitability metrics* and *Impact metrics*. The former accounts for metrics that affect the difficulty of exploitation of the given vulnerability, while the latter accounts for the possible impact if the vulnerability is exploited on a vulnerable system (FIRST.org, 2021).
- **The Temporal** metric groups represents the attributes of a vulnerability that could change over time. This metric grouping accounts for the current state of the vulnerability. The state of the vulnerability depends on the availability of measures that affect the severity of the vulnerability; for example, mitigation strategies for the vulnerabilities, like patches and workarounds, decrease the likelihood of exploitation, while available exploitation techniques increase the possibility (FIRST.org, 2021).
- **The Environmental** metric group is a modifiable version of the Base/Constant metric group and can be modified to account for the specific security need of systems within an organization.

This group is highly specific to the individual organizations and should be specialized to the organization (FIRST.org, 2021).

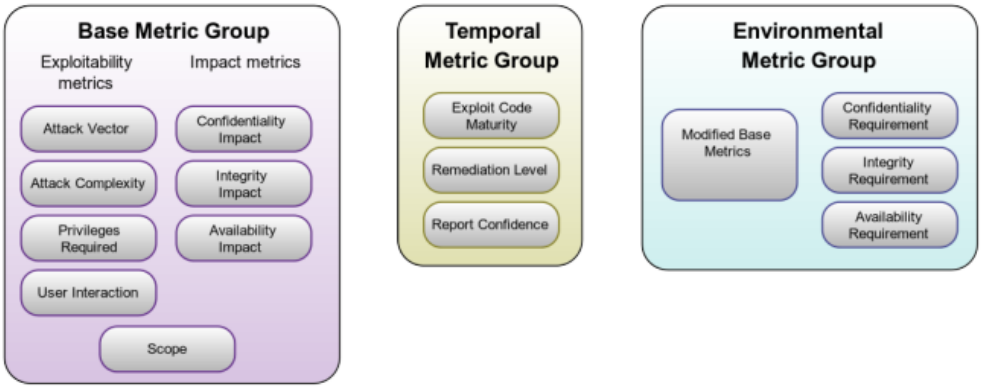

Figure 2.2: CVSS Metric Groups (FIRST.org, 2021)

Based on these metrics, seen in Figure 2.2, the CVSS scoring scheme gives the vulnerability a numerical score between 0.0 and 10.0, where higher is more severe. These scores also have an associated alphabetic score shown in Figure 2.3 and are used to quantify the severity of a vulnerability. One common way to guide patching prioritization through the usage of CVSS scores is a patching policy over a certain CVSS score. Here, a specific score, either a numerical score or an alphabetic one, is selected as the differentiated on whether or not a vulnerability takes priority. One example of this policy being used is in the Payment Card Industry Data Security Standard (PCI-DSS), where compliance, among other requirements, requires handling vulnerabilities with a CVSS score of 4.0 or higher (PCI, 2022).

Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Figure 2.3: CVSS Qualitative severity rating scale (FIRST.org, 2021)

2.2.3 Criticism of CVSS

There have been pointed out several shortcomings in CVSS, including its focus on vulnerabilities in isolation (Baiardi & Tonelli, 2021) and its inability to account for vulnerability population (exploitation rate), patch delays, patching rate, and updating mechanism (Nappa et al., 2015). Baiardi and Tonelli (2021, p. 5) claims that *"any scoring system independent of the target infrastructure only offers a rough, approximated value for a risk-based ranking."*, ultimately pointing out that CVSS

only accounts for vulnerabilities in isolation and does not consider how a vulnerability could be used in the context of an organization. Analysis of patching practices within Industrial Control Systems (ICS) performed by Wang et al. (2017) found that 10% of ICS containing a vulnerability rated critical using the CVSS metrics show a version increase after 60 days as opposed to 70% for low severity vulnerabilities. Logically, a vulnerability rated critical using the CVSS rating scale should take priority in patching, but data from real-life patching behavior shows that this is not the case. Therefore, this can be interpreted to mean that in real-life vulnerability management scenarios, vulnerability prioritization is not determined based on CVSS scores but rather based on other metrics.

The CVSS has also been criticized for its inability to quantify the confidentiality impact of a given vulnerability. For example, Howland (2023) argues that the CVSS does not account for a vulnerability's actual quantified confidentiality impact when calculating its vulnerability score. The given metrics in the confidentiality metric in the base/constant metric group are ineffective at quantifying the actual confidentiality impact of a vulnerability being exploited. For example, Howland (2023) points to CVE-2014-3566, where a man-in-the-middle attack can intercept cleartext transmissions over SSL 3.0. This vulnerability was given a score of 3.4 by the CVSS v3 scoring system but failed to account for the potential confidentiality impact it could lead to, without concern about the potentially sensitive data that could be transmitted with this vulnerability present. With this logic, a low-scoring vulnerability using CVSS could, if exploited in reality, have a significant confidentiality impact.

2.2.4 Alternative Vulnerability Scoring Systems

Fundamentally, there are several ways of thinking when deciding what vulnerabilities should be prioritized in the patching process. The aim of the CVSS method is to calculate a numerical score that quantifies how severe the exploitation of a vulnerability would be. This method provides a good foundation for understanding the potential outcome of exploitation of a vulnerability, thus motivating the mitigation of the more severe scored vulnerabilities. In contrast, it does not account for the reality of vulnerabilities exploited in the wild. Another solution that accounts for the exploitation probability of a vulnerability is the Exploit Prediction Scoring System (EPSS) introduced by Jacobs et al. (2021). This framework for vulnerability scoring focuses on the reality of what vulnerabilities are exploited rather than what vulnerabilities potentially have the worse outcome if exploited, i.e., the risk. For example, data from Jacobs et al. (2021) shows that out of 25,169 vulnerabilities collected over two years (from 2016 to 2018), only 3.7% are exploited within a 12-month window.

Consequently, while vulnerabilities that would score higher using CVSS would be judged as having a potentially more considerable risk to the business, it is not necessarily the vulnerabilities that will be exploited. In a sense, CVSS and EPSS provide different ways of quantifying vulnerability attributes. CVSS puts weight on the potential risk involved with a vulnerability, while EPSS is instead scored based on the potential threat of the vulnerability. Jacobs et al. (2021) recognizes this and argues that one option is to understand that CVSS and EPSS provide different pieces of

the vulnerability prioritization puzzle and that utilizing their strength in an extensive vulnerability management framework is a practical application.

2.3 Interrelated practices

As patching and vulnerability management are part of organizations' greater security efforts, several areas coincide. Moreover, in the literature, patch management, vulnerability management, and other components of the overall security effort do aspects similarly. This means that these aspects of security management are approached equivalently in the literature.

2.3.1 Information Sources

Information sources are a crucial aspect of patching and vulnerability management as it enables researchers and system administrators to gather information about vulnerabilities and relevant patches. Maintaining a proper strategy for collecting from diverse sources may contribute to a broad understanding of the threat landscape and allow for sufficient coverage of vulnerabilities present in applicable systems. Li et al. (2019) reports on information gathering as a crucial part of patch management with several interviews and surveys undertaken with system administrators. Consequently, Figure 2.4 showcases the variety of sources collected from the interviewees. The information sources are used as an attribute to internal vulnerability discovery to aid the researcher in identifying the most critical vulnerabilities. Although some sources are helpful, not all patching and vulnerability information is stored in one location, meaning system administrators must research several sources to gain sufficient information on how to patch their enterprise.

	Source for Update Availability	# Survey Responses	# Interview Responses
1.	Security advisories	80 (78%)	4 (24%)
2.	Direct vendor notifications	72 (71%)	11 (65%)
3.	Professional mailing lists	54 (53%)	7 (41%)
4.	Online forums	53 (52%)	7 (41%)
5.	Alerts from software	41 (40%)	10 (59%)
6.	News	40 (39%)	5 (29%)
7.	Blogs	39 (38%)	5 (29%)
8.	Third-party services	28 (28%)	0 (0%)
9.	RSS feeds	22 (22%)	3 (18%)
10.	Project mailing lists	21 (21%)	0 (0%)
11.	Social media	18 (18%)	1 (6%)
12.	Other	9 (9%)	3 (18%)
13.	No Answer	3 (3%)	0 (0%)

Figure 2.4: Sources used for discovering available updates (Li et al., 2019)

As Figure 2.4 depicts, the three most frequent information sources amongst the interviews and surveys are *security advisories*, *direct vendor notifications*, and *professional mailing lists*. Similarly, Tiefenau et al. (2020) report on participants utilizing similar methods such as mailing lists, supplier information, and third-party services that supply information regarding available patches.

Subsequently, patch management and vulnerability scanner tools like SCCM (Microsoft System Center Configuration Manager) and Nessus were prevalent in providing participants with valuable information regarding their systems.

Jenkins et al. (2020) report on security forums being a vital source of information for interviewed system administrators, where *PatchManagement.org*'s group is assessed over a period to identify effectiveness and usefulness for its members. Consequently, the platform is utilized for sharing patch-related information amongst system administrators and other system practitioners and is a sufficient arena for gathering critical information. Furthermore, as Jenkins et al. (2020) suggests, participating in an online forum with other experts in a field increases the chances of successfully dealing with uncertainties relating to patching and vulnerability management and provides a source of learning and improvement for its members. Contrarily, only engaging in an online forum is not sufficient as there is other patch-related information found elsewhere that might also apply to the target systems.

Additionally, as Dissanayake et al. (2023) reports after interviewing system administrators on automation in patching, the participants expressed a need for a distinct platform that gathered information from multiple sources. A centralized platform may contribute to fewer resources spent manually searching for threat information and patching information, which can be utilized to perform other manual tasks.

2.3.2 The Human Aspect in Patching and Vulnerability Management

Performing patching and vulnerability management is not solely operated by automated tools but also by system administrators, IT professionals, and other fields working for their companies or being hired by other companies. Accordingly, the human factor is crucial as it lays the foundation for how the configuration of the systems within a company is tended. Dietrich et al. (2018) identifies security misconfigurations performed by system administrators as a growing problem within the patching and vulnerability management field. In their work, they highlight that nearly 77% of the 221 interviewed/surveyed operators had misconfigured a system throughout their time with a company, and nearly 31% state there has been a security incident due to a misconfiguration event. Misconfigurations, as reported by Dietrich et al. (2018), may result from system administrators not reporting a fault for fear of attribution and punishment. Therefore, implementing *blameless postmortems* involves creating a safe space for system administrators to report known misconfigurations without fearing attribution. Additionally, the paper revealed that a lack of knowledge and experience are the two main factors contributing to misconfigurations in an enterprise.

2.3.3 Identified Obstacles

Tiefenau et al. (2020) reports on various obstacles system administrators faced after extensive interviews. The paper reports on *stability considerations* and *downtime* as the two main reasons patching deployment is delayed or interrupted. Here, *stability considerations* refers to how stable and prone to error a system might be after deploying patches for that system, while *downtime* refers

to a system being unavailable. As a mitigation strategy, one participant's suggested mitigation would be to upgrade to a version that allowed for near-to-hot-swap updates instead of patching an old version which requires significant downtime to be patched. Additionally, a lack of information regarding how to perform a patch and expertise on the processes contributes to slower deployment times and a higher chance of misconfiguration. As a result, system administrators must rely heavily on the expertise of third-party vendors and trust the vendor that the classification is set correctly.

Software in enterprises typically relies on a wide variety of libraries and dependencies to run sufficiently, where a patch on the software may cause dependencies to be outdated and break. Consequently, the leading software breaks and becomes unusable without providing a patch for the dependencies that match the software. Participants reported that this type of obstacle significantly delays the patching process and contributes to slower deployment times (Tiefenau et al., 2020). Contrarily, Li et al. (2019) suggests implementing a DSU (Dynamic Software Updating) as a strategy for effectivizing patching deployment. DSU allows for performing live updates without the need for restarts or any significant downtime. This might aid system administrators with timely implementing patches; however, DSU has not been widely tested. Therefore, the long-term effects or complications following this implementation are unknown and could potentially impact specific servers or systems.

2.3.4 Roles in Patching and Vulnerability Management

As with all other components of an organization, the patching and vulnerability management process requires directives on how it should be facilitated. These directives are often developed and imposed on different levels of the organization's structure. As patching and vulnerability management are fundamentally a part of the organization's information security strategy, the other areas of responsibility will be relatively similar between different components of the overall security effort. Therefore, the roles associated with patching and vulnerability management within an organization will be similar to that of other elements of the organization's information security effort. Bautista (2018) proposes a layered model for leadership of cyber intelligence as seen in Figure 2.5. In this model, the leadership of cyber intelligence is divided into levels encompassing the different responsibility areas. The levels of responsibility can be compared to the difference in workload and authority held by an Officer, Manager, and Analyst and are divided as such:

- **The Strategic Level** of leadership comprises the organization's responsibility of facilitating the cyber intelligence effort by identifying critical infrastructure and associated threats and vulnerabilities, mobilizing the needed resources, and developing policies and procedures. At this level, one would typically find higher-ranking information and security employees corresponding to Chief Information Officers (CIO) and Chief Information Security Officers (CISO) (Bautista, 2018).
- **The Tactical Level** of leadership's responsibility is to support the strategic levels efforts into cyber intelligence, further organization of resources, and implementing policies and procedures. IT and Security managers would supervise this leadership level (Bautista, 2018).

- **The Operational Level** of cyber intelligence leadership covers supporting the higher levels, collecting the needed information to organize resources effectively, and enforcing the policies and procedures. Typically, this level would be facilitated by Security Team leads and Analysts (Bautista, 2018).

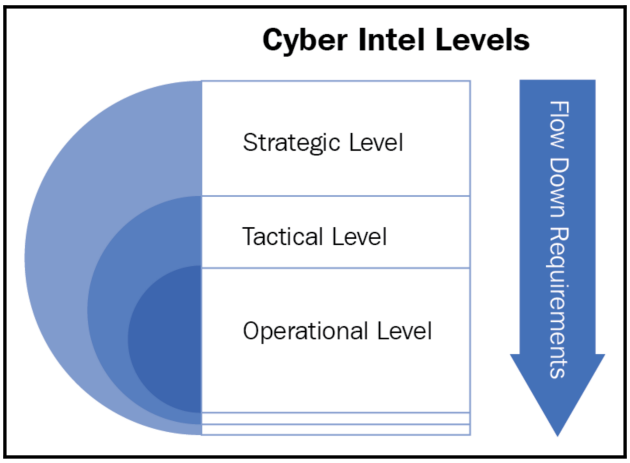

Figure 2.5: Cyber Intel Levels (Bautista, 2018)

The described model above describes the different areas of responsibility within cyber intelligence and could alternatively be applied to other aspects of information security. Describing information security using Strategic, Tactical, and Operational levels is not an original idea and has been applied to different areas of information security (Ahmad et al., 2014; Schulze, 2020). Therefore, it would stand to reason to assume that this idea also would be applicable from a patching and vulnerability management point of view. Moreover, depending on the size of an organization, it would be logical to assume that the available resources for information security would limit the number of overall employees working with information security. As a result, an employee could be responsible for several areas of information security, including patch and vulnerability management, and the described methodology would still apply to the organizational structure. This layered approach does not limit one employee to one specific role, instead leaving it up to the specific organization implementing the model.

2.4 Gaps and problems

Patching and vulnerability management are areas that continuously develop and are often integrated differently depending on the enterprise. After undertaking a thorough literature review of the research performed around patching and vulnerability management, several noticeable gaps and problems have been identified that influence the scope and feasibility of the study while also justifying the need for a framework that accounts for these limitations in the literature.

By conducting a literature review of the topic mentioned earlier, it is evident that there is a notable gap concerning the presence of a comprehensive framework describing the over-arching and underlying

processes of patching and vulnerability management in enterprises. The existing literature overviews specific areas that contribute to insight and value. However, there is a lack of literature describing the relationship and interdependencies of patching and vulnerability management. A framework depicting the processes of information gathering, patching, vulnerability management, testing, and deployment may contribute to organizations being able to review their current strategy and validate their approach with a framework based on relevant literature and interviews with relevant peers. Although there is a lack of a comprehensive framework depicting and elaborating on the totality concerning patching and vulnerability management, Huang et al. (2012), with their framework on the patch management process workflow (Figure 2.1), is the closest the literature has gotten to a complete framework. However, essential processes such as an elaborate information-gathering process and specific information regarding C-level influence technical tactical and operational-level processes are absent. Moreover, also the connectivity of patching and vulnerability management is not present in this framework.

One gap in the literature that made it challenging to paint a complete picture of how patching and vulnerability management is handled is the lack of research into how it is structured. During the literature review process, no source described the different roles involved with patch and vulnerability management. Therefore, to account for the gap in the literature, the structuring of patching and vulnerability management is compared to other components of an organization's security effort in Section 2.3.4. The understanding is that the patch and vulnerability-related work structure is the same, or at least comparable, to other parts of the security effort. Therefore, models detailing the structuring of other security operations aspects should apply to patch and vulnerability management.

3 | Research Approach

The research approaches proposed in this chapter aim to describe the suitability and justification for why specific research methods are utilized to answer the research questions. The literature review and the qualitative study with interviews are correlated as the findings from the literature review are utilized to develop relevant questions and topics in the interview process. Conversely, the qualitative study aims to affirm the findings from the literature review. As such, the methodologies aim to gather as much relevant information as possible to ensure the interview process is accomplished with relevancy. The chapter describes the approaches used in the literature review and the qualitative research, with argumentation for the suitability of the methods.

3.1 Literature Review

A literature review ultimately aids the researcher in gaining an understanding of the literature's content within a specific topic area. Additionally, a literature review may provide a sufficient overview of the existing gaps in the literature for researchers to explore and showcase the existing literature's research findings to build conceptual models or frameworks on (Snyder, 2019). The literature review's findings are utilized to answer the research questions and provide a knowledge base of the existing literature on patching and vulnerability management. In this context, a semi-systematic literature review is conducted as a baseline for finding and assessing the relevant literature utilized in this thesis. As patching and vulnerability management have no right or wrong approach but depend on the organization's resources and capabilities, this method aids in evaluating the different approaches to provide unbiased information gathering.

3.1.1 Methodology

For conducting a literature review, a baseline method has to be integrated to aid the researchers in identifying the key aspects of the information gathering and analysis. Following Xiao and Watson (2019), their summarized model from Kitchenham and Charters (2007)'s key points on how to conduct a literature review is followed as it provides a clear and concise step-by-step approach to the key aspects to consider for developing a successful literature review, as depicted in Figure 3.1. The model formulates each step, which later has been applied by the researchers in the method of conducting a literature review. As the figure proposes, the number of articles gets incrementally smaller as assessment and literature criteria are enforced (as explained in Section 3.1.2 and Section

3.1.5), resulting in relevant and applicable papers. Furthermore, the quality of the papers is assessed after performing the inclusion and exclusion criteria before key aspects, takeaways, findings, and results are extracted. Even though Xiao and Watson (2019)'s process map relates to a systematic literature review, they state that *"Despite differences in procedures across various types of literature reviews, all the reviews can be conducted following eight common steps"*. Thus, the reason for choosing this process map is that it provides a sufficient overview of the key processes while clearly defining each step included in each area. Furthermore, this process map is utilized as a guide for performing each step in the proposed order, given that each step relies on the proper execution of its last step. The findings from the main part, *"Conducting the review"*, is an iterative process towards planning the review as each finding and analysis can aid in formulating the problem and identifying gaps.

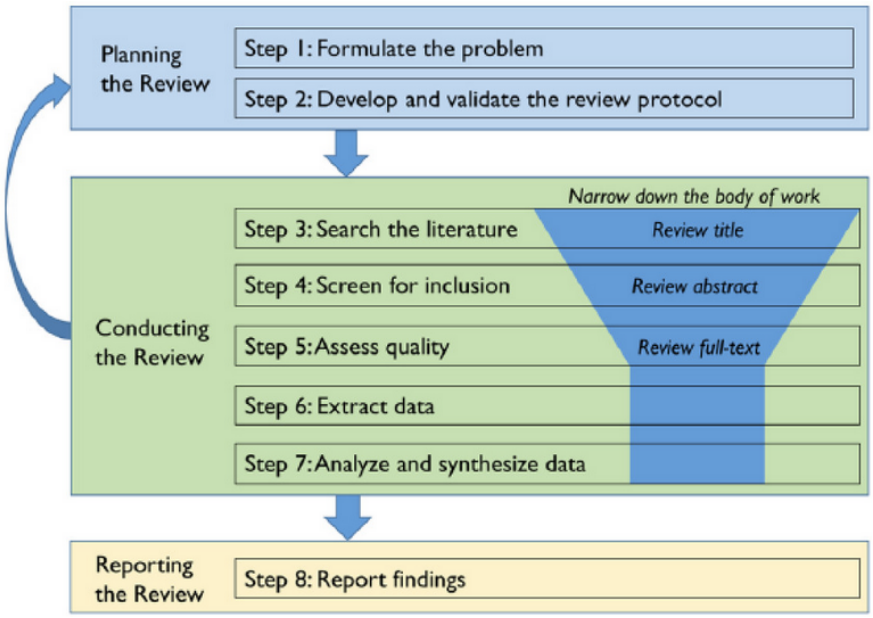

Figure 3.1: Process of systematic literature review (Xiao & Watson, 2019)

Xiao and Watson (2019)'s process map is utilized as a guideline for performing the semi-systematic literature review, where key aspects are included. In the context of our study, the formulation of the problem statement relied on the existing literature around a framework for patching and vulnerability management. Consequently, the aim was to identify the gaps in the problem statement and revolve the research and results around developing a framework. Furthermore, the literature was searched based on the inclusion and exclusion criteria, with a continuous assessment of whether or not the articles were relevant to the topic of study. Then, the crucial parts of the papers were read to extract the data, where key points and findings were noted to analyze the data later. Lastly, after the search was finished and the key points were gathered, comparisons and findings were prepared and presented in the literature review section.

For the methodology of how to categorize and structure the literature review, Watson and Webster (2002)'s idea of utilizing a concept-oriented approach contrary to an author-oriented approach is

employed in the report. A concept-oriented approach allows for a broad sample of articles not bound to a specific author while simultaneously allowing for certain key concepts within the topic area to be prevalent. The approach relates to identifying relevant concepts as literature is researched and mapping relevant literature with each concept. This creates an overview of the concepts and the papers that fall into the specific categories and is utilized as a baseline within the body of the literature review.

3.1.2 Literature Criteria

The research in the thesis is mainly limited to papers released between 2013 – 2023, but some exceptions are made where the literature is still relevant. Older papers might contain outdated information, but to gain a thorough understanding of the development of patching and vulnerability management, a handful of older papers are included as applicable. Only the most recent recommendations and data are used for government-released reports or reports surrounding yearly recapitulations. Lastly, only papers that are written in English or Norwegian are considered. The complete list of the research criteria for the papers are:

- The papers must not be older than ten years, but exceptions are made if the content is still highly applicable to the topic
- The papers must be written in English or Norwegian
- Only the latest releases on reports are utilized to ensure updated information
- The papers should be peer-reviewed to ensure high-quality articles

3.1.3 Search Process

The semi-systematic literature review is carried out by primarily assessing high-quality literature to ensure a certain standard of articles within the report. The higher the standard of the articles, the higher the likelihood of the article being well-written with proper explanations of concepts that can be utilized in the report. For the literature research, assessing articles in electronic databases is the primary search method. The primary databases used for searching are the ACM Digital Library, Google Scholar, and IEEE. The aforementioned databases contain academic articles with excellent search functionality for more precise searches. Secondly, as Watson and Webster (2002) advocate, backward citations (also referred to as snowballing) is a concept of reviewing the citations of the preliminary articles to review if there are more applicable and relevant papers to be used. Contrarily, a forward search can be utilized, which revolves around reviewing the articles that have since cited the reviewed papers (Xiao & Watson, 2019). This gives an overview of which authors and papers have built on the previous literature.

Each electronic database has its search query syntax where one may specify certain aspects or topics to get a distilled result, contrary to only searching for one keyword. Below are two examples of Google Scholar and ACM Digital Library showcasing the possibilities of the search syntax. These

two examples are also utilized as search queries within the report:

Google Scholar:

```
patch management OR vulnerability management intitle:patch* OR vulnerabilit*
```

ACM Digital Library:

```
[[All: patch*] OR [All: vulnerabilit*]] AND [[Title: patch*] OR
[Title: vulnerabilit*]]
```

Overall, most of the search process revolved around identifying specific keywords applicable to the topic area and finding papers that included those keywords (while simultaneously passing the former requirements). Collectively, the use of keywords aided in finding relevant and sufficient literature that could be used in the further analysis of the paper. Table 3.1 showcases the keywords used in the literature search, which revolve around the area of patching and vulnerability management:

Table 3.1: Keywords utilized in the literature review search process

Keyword
Patch Management
Vulnerability Management
Common Vulnerability Scoring System (CVSS)
Enterprise/organization
Patching/Vulnerability Framework
Vulnerability Prioritization
Misconfiguration

3.1.4 Organization of Literature

A particular set of tools is implemented to effectivize and uphold standards to ensure the literature is adequately stored and organized. Therefore, the Mendeley reference manager is used for automatically collecting and transferring citations. The entries are stored as BibLaTeX entries, allowing seamless integration within the reference manager of the editor LaTeX. Additionally, all papers found are stored in a table within a Word document in the university cloud to keep track of the papers found and analyzed. Within the Word document, initial judgment and a conceptual grouping are prevalent to aid the organization within the literature review regarding conceptual categorization.

3.1.5 Screening of the papers

Subsequently, when the preliminary articles are collected utilizing the requirements proposed in Section 3.1.2, the articles are thoroughly screened. As the collective amount of articles is of a high volume, following a specific procedure helps filter out irrelevant papers or papers whose scope or

content does not fit the contribution criteria. As such, we performed a two-way review is performed, which includes (1) assessing the literature through a coarse filter to include relevant papers and (2) performing a full-text review of the articles found in the first step, as reported on in Xiao and Watson (2019). In this stage, the research should be inclusive, and if there is doubt about whether or not a paper should be included, it should be included in the first stages as it may contain valuable information after reading the full text. Moreover, as Xiao and Watson (2019) state, the screening approach for papers is mostly trivial whether a systematic approach is used compared to a semi-systematic approach. Therefore, a similar screening method is utilized in the report. For the coarse assessment phase, the papers' abstract, introduction, and conclusion were assessed to identify the relevancy and fulfillment of the literature search criteria. As the abstract is a complete paper summary, this part is assessed first. If any information is unclear or insufficient, the introduction and conclusion are also evaluated.

Figure 3.2 depicts the continuous and final screening of the reports found within selected databases. The model is developed in relation to Page et al. (2021)'s PRISMA model, designed to depict the screening process within a systematic literature review. As such, the model showcases the entire process from identification, to screening, to the final included articles. The literature search started with 165 articles collected from commonly known databases, where the primary databases used for the search were ACM Digital Library, IEEE, and Google Scholar. Duplicates across the databases were removed, along with articles that required paid access or were inaccessible for other reasons. The initial screening stage involved excluding records based on irrelevancy in terms of the title, abstract, introduction, and conclusion, whereas the extended screening revolved around reading the records' full text to assess the paper's suitability again. Subsequently, 38 articles remained after finishing the extended screening. Furthermore, the articles were screened based on three categories: *article quality*, *relevancy*, and *repeated topics/content*. Eighteen records were then left. As for identifying studies via other methods, meaning citation searching (also referred to as snowballing), which revolves around assessing the references of relevant articles, five more articles were discovered, making the total eligible articles 23.

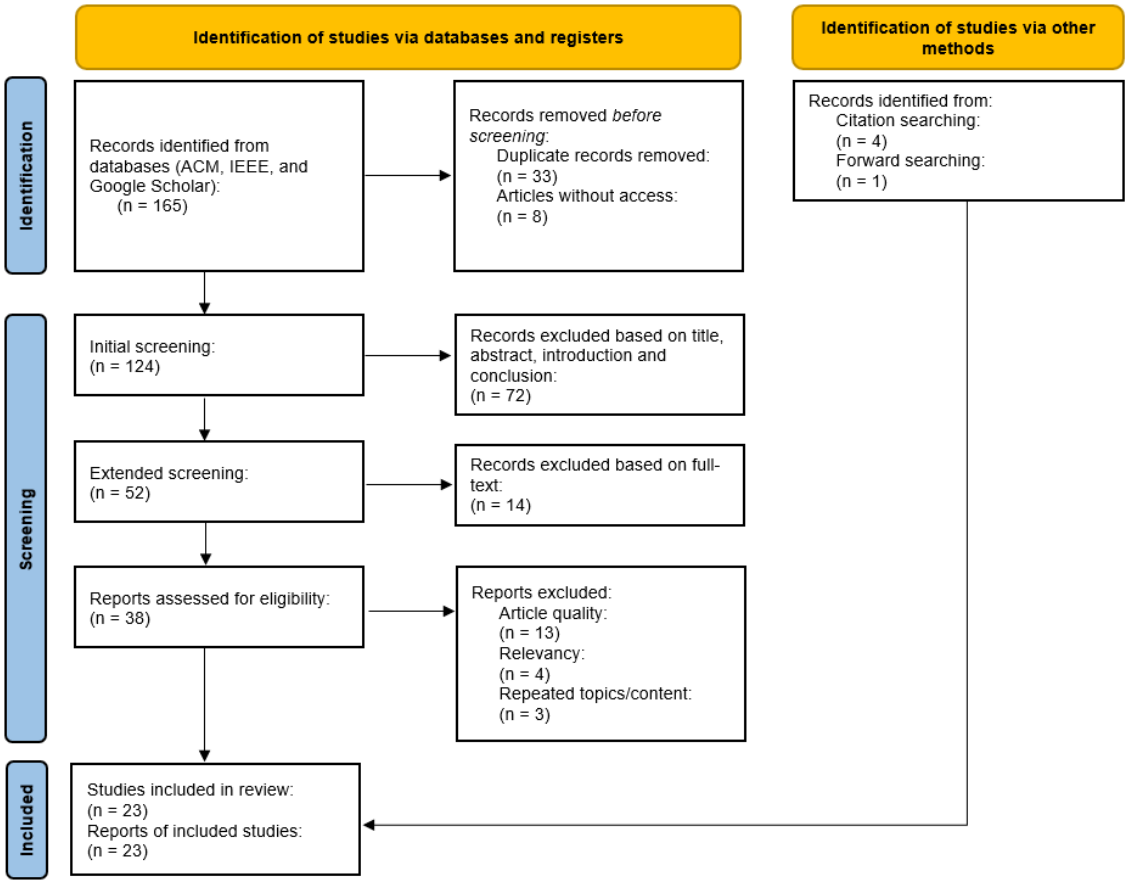

Figure 3.2: Screenning procedure following the PRISMA methodology (Page et al., 2021)

Below in Figure 3.2 are the 23 eligible sources that are built on as a foundation in the literature review process after the screening process with inclusion and exclusion criteria are performed:

Table 3.2: Eligible literature review articles

Author (Year)	Title	Journal/Source	Keywords
Ahmad, A., Maynard, S. & Park, S. (2014)	Information security strategies: Towards an organizational multi-strategy perspective	*Journal of Intelligent Manufacturing*	Information security strategy, Deterrence, Prevention, Compartmentalization, Deception, Defense in depth
Baiardi, F., & Tonelli, F. (2021)	Twin Based Continuous Patching to Minimize Cyber Risk	*European Journal for Security Research*	Model-based, Adversary emulation, Digital twin, Vulnerability, Patch schedule
Bautista, W. J. (2018)	Practical Cyber Intelligence: How action-based intelligence can be an effective response to incidents	-	-

Dey, D., Lahiri, A., & Zhang, G. (2015)	Optimal Policies for Security Patch Management	*INFORMS Journal on Computing*	Security, Vulnerability, Patching, Patching Policy, Exploitation cost, Setup cost, Disruption cost
Dietrich et al. (2018)	Investigating System Operators' Perspective on Security Misconfigurations	*Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*	Computer systems, System operations, Operators, Administrators, Security, Misconfiguration, Vulnerabilities, Human factors
Dissanayake et al. (2022)	Software security patch management - A systematic literature review of challenges, approaches, tools, and practices	*Information and Software Technology*	Security patch management, Vulnerability management, Systematic literature review
Dissanayake et al. (2023)	An Empirical Study of Automation in Software Security Patch Management	*Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering.*	Security updates, Patch management, Vulnerability management
Dissanayake et al. (2022)	Why, How and Where of Delays in Software Security Patch Management: An Empirical Investigation in the Healthcare Sector	*Proceedings of the ACM on Human-Computer Interaction*	Patch management, Security updates, Delays, Socio-technical research
Farris et al. (2018)	VULCON: A System for Vulnerability Prioritization, Mitigation, and Management	*ACM Transactions on Privacy and Security*	Cyber-Security Engineer, Vulnerability Data, Patch Management, Cyber-Security Operations Center (CSOC), Mixed Integer Constraint Optimization
Hore, S., Shah, A., & Bastian, N. D. (2023)	Deep VULMAN: A Deep Reinforcement Learning-Enabled Cyber Vulnerability Management Framework	*Expert Systems with Applications*	Cyber vulnerability management, Vulnerability prioritization, Security resources optimization, Deep reinforcement learning, Integer programming, DRL defender framework
Howland, H. (2023)	CVSS: Ubiquitous and Broken	*Digital Threats: Research and Practice*	Threat and vulnerability management, Security standards, SCAP, CVSS, Remediation prioritization

Huang et al. (2012)	Patch management automation for enterprise cloud	*2012 IEEE Network Operations and Management Symposium*	Patch management automation, Enterprise cloud, Middleware, Application testing, Enterprise customers, Operating systems post-update, Human operator, Patching process, VM restoration
Jacob et al. (2021)	Exploit Prediction Scoring System (EPSS)	*Digital Threats: Research and Practice*	Vulnerability management, Vulnerability exploits, Machine learning, EPSS
Jenkins et al. (2020)	"Anyone Else Seeing this Error?" : Community, System Administrators, and Patch Information	*2020 IEEE European Symposium on Security and Privacy (EuroS&P)*	Human factors, Security usability, Technology social factors
Li et al. (2019)	Keepers of the machines: examining how system administrators manage software updates	*Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security*	-
Mell et al. (2022)	Measuring the Common Vulnerability Scoring System Base Score Equation	*NIST*	Computer, Common Vulnerability Scoring System, Error, Expert opinion, Measurement Measuring, Metrics, Network, Scoring, Security
Nappa et al. (2015)	The Attack of the Clones: A Study of the Impact of Shared Code on Vulnerability Patching	*Proceedings - IEEE Symposium on Security and Privacy*	Software vulnerabilities, Patch deployment, Shared code, Client applications, Vulnerability exploits
Schulze, M. (2020)	Cyber in War: Assessing the Strategic, Tactical, and Operational Utility of Military Cyber Operations	*International Conference on Cyber Conflict, CYCON*	Cyber in war, Military cyber operations, Levels of war, Strategic cyber attacks, Tactical cyber, Small-n case study
Serio, L., & Gentile, U. (2019)	Survey on international standards and best practices for patch management of complex industrial control systems: The critical infrastructure of particle accelerators case study	*International Journal of Critical Computer-Based Systems*	Industrial control systems, ICSs, Patch management, Critical Infrastructure, Particles accelerators, Critical computer-based systems International standards

Souppaya, M., & Scarfone, K. (2022)	Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology	*NIST*	Enterprise patch management, Patch, Risk management, Update, Upgrade, Vulnerability management
Tiefenau et al. (2020)	Security, availability, and multiple information sources: Exploring update behavior of system administrators	*Proceedings of the Sixteenth USENIX Conference on Usable Privacy and Security*	-
Wang et al. (2017)	Characterizing and Modeling Patching Practices of Industrial Control Systems	*Proceedings of the ACM on Measurement and Analysis of Computing Systems*	Industrial Control Systems (ICS), Shodan, Vulnerability Patching
Xu et al. (2022)	Tracking patches for open source software vulnerabilities	*Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*	Open-source software, Vulnerability patches, Patch tracking

3.2 Qualitative Study

This exploratory study aims to understand how organizations approach patching and vulnerability management, facilitate it, and what challenges they experience with patching and vulnerability management, to apply the assembled observation to the literature findings in Chapter 2 further to develop the conceptual model for patching and vulnerability management. Therefore, the following research questions are developed to acquire the required information to ensure that the research satisfies these needs.

- **RQ 1:** How are organizations facilitating patching and vulnerability management?
- **RQ 2:** How can the insight from appropriate interview subjects and theory enhance the patching and vulnerability management?

A qualitative study seeks to collect information on a specific topic through the experiences and options of relevant subjects. Alternatively, Hammersley (2013, p. 12) defines qualitative research as;

> *A form of social inquiry that tends to adopt a flexible and data-driven research design, to use relatively unstructured data, to emphasize the essential role of subjectivity in the research process, to study a small number of naturally occurring cases in detail, and to use verbal rather than statistical forms of analysis* (Hammersley, 2013).

Therefore, following Hammersley (2013)'s definition of qualitative research, several key charac-

teristics can be extracted. Firstly, qualitative research follows a less structured method for data collection, and as a result, the collected data is comparably more unstructured than other research methods. Secondly, the analysis of qualitatively collected data focuses more on the subjectiveness of the collected responses. Finally, the extracted findings for the research are more suitably analyzed by looking at the context of the data rather than presenting statistical findings. Moreover, Hollstein (2011, p. 3) describes the qualitative method as *"difficult to comprehensively account for"* and attributes its *"heterogeneous research landscape"* as the cause. In other words, the qualitative method has been used to describe several flavors of *"social inquiry"*. Consequently, a qualitative study can be conducted through several different methods. In their research on qualitative research methodologies within cybersecurity, Fujs et al. (2019) identifies seven different methods for performing a qualitative study. These are *Action Research*, *Case Study*, *Delphi method*, *Focus group*, *Grounded theory*, *Interview*, and *Observation*. However, Hollstein (2011, p. 3) lists *"different forms of observation, interviewing techniques with low levels of standardization (such as open-ended, unstructured interviews, partially or semi-structured interviews, guided or narrative interviews), and the collection of documents or archival data"* as the different qualitative methods. Therefore, qualitative research can be used to describe several methods for collecting information about a specific subject.

Performing a qualitative study using interviews as the method for data collection is the most suitable approach given the aim of the research. Although a quantitative research approach will provide measurable data on a specific subject, such as the utilization of a specific patching tool across organizations, it does provide the needed data for this exploratory study. Answering research questions through quantitative analysis will retain that social facts have an objective reality (Kamolson, 2007). By contrast, a qualitative approach would present an insight into what, how, and why something is done within an organization, such as why a specific patching tool is implemented. Proving an objective reality about something can inform decisions, but understanding why something is done is imperative to the research. Therefore, out of the two examples presented, the latter would provide the most useful observation.

Between different qualitative research methods, there are arguments to be made for interviews being the most suitable approach. Research done by Fujs et al. (2019) indicates that interviewing is the most commonly used qualitative research method for cybersecurity research studies in organizational cybersecurity. Moreover, quoting Langley and Meziani (2020, p. 1); *"Interviews are so common within the field that they are largely taken for granted—the obvious default method for the qualitative researcher."* Alternatively, looking past the given norm of how qualitative research is done within the subject area of cybersecurity, the other determining factor is what this research is being used for.

As defined in research question 2, a part of the wanted output of the research is insight from appropriate subjects on patching and vulnerability management. Performing quantitative research to satisfy this research question requires an understanding of patching and vulnerability management within organizations that is not present. Moreover, the same predicament also applies to other

qualitative research methods. Consequently, a number of the defined qualitative research approach immediately become unsuitable, while others become unrealistic based on the given limitations of the thesis. Below in Table 3.3, the qualitative methods collected from Fujs et al. (2019) are assessed based on their suitability for the given research. The chosen method, interviews, is highlighted with a short justification for its suitability.

Table 3.3: Qualitative research methods gathered from Fujs et al. (2019) and their suitability with the research

Research Method	**Suitability**
Action Research	Action research becomes unsuitable for the given research because of its practical nature. Action research requires the researcher to try the theory in real situations (Avison et al., 1999).
Case Study	Case study is inappropriate for the research because no case has been assigned or discovered.
Delphi Method	The Delphi Method for qualitative research focuses on using subject area experts for reaching a consensus (Grime & Wright, 2016). For the given research, it is unrealistic that a total consensus on how patching and vulnerability management is handled is reached.
Focus Group	The Focus group approach requires the assembly of a group of subjects and allows for discussion within the group (Rabiee, 2004). This method is not suited based on the interview subjects' constraints.
Grounded Theory	The created research questions and the expected outcome of this research do not fit within the expected outcome of the grounded theory method.
Observation	Inquiry of the research subject matter through observation. This thesis does not have a partner company, so observing patching and vulnerability management in real-life scenarios is unrealistic.
Interview	Interviews with relevant subject matter experts. It is realistic to gather relevant subjects and feasible given the other constraints on the research.

3.2.1 Research Design

This research investigates how organizations perform patching and vulnerability management by questioning relevant interview subjects about internal policies, procedures, and routines. These relevant interview subjects are characterized by their expertise in patch and vulnerability management and extensive experience working with the topic areas in their respective organizations. The selection of interview subjects is further discussed in Section 3.2.2. Moreover, the scope of this research is inquiring subject matter experts of organizations on how they manage patching and vulnerabilities and discussing the framework described in Section 4.1 through interviews. Therefore, what is

considered inside and outside of scope is specified below:

Inside of the Scope

- Conducting a qualitative study on patching and vulnerability management through interviews
- Interviewing employees with appropriate experience in patching and vulnerability management in Organizations
- Performing analysis on the anonymized data produced from the interviews.

Outside of the Scope

- Any other form of research, qualitative or quantitative
- Interviews with employees without specific patching or vulnerability management knowledge

The research described in the thesis follows Kvale (2011)'s *"Seven stages of an interview inquiry"* methodology. The seven stages of this methodology, with an explanation of its usage in the research, are seen in Table 3.4:

Table 3.4: The *"Seven stages of an interview inquiry"* (Kvale, 2011), and its application in the research

Stage	Description	Implementation
Thematizing	Initial establishment of *what* and *why* something is being researched	Initial themes discovered during the literature review processes in Chapter 2, further developed into the themes in the interview guide seen in Appendix A
Designing	Development of the plan of how the research is carried out and how it should acquire the required knowledge in an ethical manner	Design inspired by Kvale (2011), formulated through the seven steps in Section 3.2.1. Using a qualitative semi-structured approach discussed in Section 3.3.1
Interviewing	Actual conduction of interviews	Satisfied through the interview processes
Transcribing	Transcription of the interviews for analysis	Transcription of the interviews used in the findings in Chapter 4
Analyzing	Analyzation of the transcribed interviews	Analyzation inspired by Kvale (2011)'s *"Six Steps of Analysis"*, further discussed in Section 3.4
Verifying	Verification of the findings, determining the validity of the gathered information (Kvale, 2011)	Verification done through the consensus of interview subject responses, described in Section 3.4.1

Reporting	Description of the findings and methodologies used for the research	Description of methodology throughout Section 3.2 and reported in the findings in Chapter 4

The seven stages, described in Table 3.4, ensures that the research is designed thoroughly, that all the needed data is collected, that the data is verifiable, and that it is collected ethically. Moreover, the stages in the methodology in Kvale (2011) deal with specific parts of the interview process and are accordingly in the research design. These stages sometimes get intertwined, meaning that parts of the interview process might fit into more than one stage in this methodology. Therefore, while generally Kvale (2011)'s methodology is followed, divergence from is done if appropriate for the given research. The study's overall design follows Kvale (2011)'s methodology, but the individual parts are altered to be more suitable for this study.

In Kvale (2011)'s methodology, the *Thematizing* is critical as it defines what is being researched. This stage aims to explain the intent behind the interview inquiry and determine what themes should be investigated, specifically focusing on what data should be collected and why (Kvale, 2011). This stage is realized for this research by creating an interview guide. This interview guide has been developed to ensure that the interviews are carried out appropriately and is seen in Appendix A. A more detailed description of how the interview guide was developed is discussed in Section 3.3.1. The interview guide contains the problem statement, the research questions, interview themes, interview techniques, and the interview questions for the research. Relevant for the *Thematizing* stages are the interview themes developed. These were designed for the research based on the research problem and, more specifically, the research questions. Moreover, it attempts to quantify what knowledge is wanted for the research through generalized themes. These themes were subsequently used to develop the interview questions used in the interviews.

Intertwined in the *Thematizing* stage is the *Designing* stage, where the actual design of the research is defined. This stage describes explicitly how the wanted knowledge is collected, along with considerations of the moral implications of the investigation. The interview techniques from the interview guide are specifically crucial for this stage. These techniques are developed to extract as much useful information from the interview subjects as possible while still keeping the subject comfortable with talking about the different themes, allowing the interview subject to elaborate on specific themes, and continuously getting the subject content on asking about potentially sensitive subjects. Another essential part of the *Designing* stage of the methodology is how the data is collected, which is further discussed in Section 3.3.

Another vital factor to consider for the given research area is the sensitivity of some of the discussed data. Therefore, the developed questions remain general because they gather objective facts about the organization's patching and vulnerability management process while still avoiding collecting specific and potentially sensitive details. In other words, the questions are specific enough to gather the required information without being too unambiguous to prevent harming the organization if

the information is public. In this scenario, harmful is the potential damage the information could attribute to the organization, financially or in the sense of security if made public. Moreover, since the collected data in this research could potentially be harmful if the organization and the interview subjects are made public, data anonymization is ensured. The anonymization is crucial as it ensures that the interviewed employee or the organization cannot be identified from published data. The ethicality of data collection is also within the realm of keeping data anonymous, which is further discussed in Section 3.5.

The third and fourth stage of Kvale (2011)'s methodology is *Interviewing* and *Transcribing*. These stages go hand in hand and are satisfied through the interview processes and the subsequent transcription of these interviews. Moreover, the interviews will be in Norwegian or English, depending on the subject's vocabulary and wishes. The methodology followed for the interview process follows the semi-systematic approach, which is discussed in Section 3.3.1. The *Transcribing* stage is further discussed in 3.3.2. Furthermore, the *Analyzing* stage of the methodology is discussed further in Section 3.4, while the *Verifying* stage is discussed in Section 3.4.1. The final stage of the methodology, *Reporting*, is realized for this research through Chapter 4, which reports on the findings from the interviews.

3.2.2 Interview Subject selection

As discussed in Section 3.2.1, the scope of this research is looking at organizations and their approach to patch and vulnerability management. Therefore, for an organization to be applicable, it had to operate its entire patch and vulnerability management efforts themselves or at least parts of them. Furthermore, there was an emphasis on the assemblage of different maturity levels within the chosen organizations. Although there is a distinct advantage in talking to more mature organizations, especially regarding specific policies and procedures, less mature organizations have a unique insight concerning the framework's applicability to organizations with less established routines. Alternatively, organizations of different levels of maturity provide distinct types of feedback. The expected acquired knowledge from the more mature ones is how the actual patching and vulnerability management is handled and the suitability of the framework to their organizations. On the other hand, less mature organizations will supply opinions on how applicable the framework is, given their unique insight of looking at the framework without an overpowering presence of other structured methodologies. Therefore, the chosen organizations are primarily on the side of more mature because the insight of less mature organizations is expected to be somewhat similar. Having more mature organizations, conversely, will provide a consensus on how to handle patching and vulnerability management maturely.

Specific characteristics are needed for the interview subjects chosen to add value to the research. Firstly, a subject with long field experience is advantageous but optional. Subjects with less experience will have valuable insight regarding how coherent the framework is to less experienced users. This is important as the framework should be understandable and usable for all experience levels. On the other hand, subjects with more experience will better understand how the patching

and vulnerability management needs are realized while potentially having experiences from multiple organizations. Secondly, the interview subject must inhabit a position that handles patching, vulnerabilities, or an associated area, ensuring that the interview subjects have valuable insight into patching and vulnerability management. Furthermore, this will also ensure that the collected data verifiably have some validity.

To ensure that appropriate interview subjects are chosen and sufficient sampling is satisfied, a mixture of *Purposeful sampling* and *Convenience Sampling* is used. *Purposeful Sampling* is used in the research by assembling interview subjects through the researchers' perception of appropriate interview subject candidates. Moreover, this is used as the approached interview subjects are gathered using the researchers' insight into applicable organizations and, by extension, employees. Similarly, *Convenience Sampling* is used as an extension on *Purposeful Sampling* as a last resort, where the convenience of interview subjects trumps expertise, given the timeframe and available relevant interview subjects (Suri, 2011).

Table 3.5: Interviewee Demographics

Pseudonym	**Role**	**YIR (YOE)**	**Business Area**	**Business Size**	**#**
Back_Developer	Senior Backend Developer	0,5 (19)	IT Security	Small	A
Lead_Security	Senior Manager	1 (26)	IT Consulting	Large	B
Infra_Advisor	Senior Advisor	2,5 (16)	IT Consulting	Medium	C
Sec_Engineer_1	Senior Security Engineer	15 (30)	Technology and Manufacturing	Large	D
Sec_Engineer_2	Security Engineer	3,5 (12)	Technology and Manufacturing	Large	D
Lead_IT	Team Manager IT	12 (14)	Technology and Manufacturing	Large	D
Sec_Officer	Security Officer	7 (17)	Banking	Large	E
Lead_Application	Application Manager	5-6 (22)	IT Consulting	Large	F
Infra_Engineer_1	Infrastructure engineer	1,5 (5)	IT Consulting	Large	F
Infra_Engineer_2	Infrastructure engineer	3 (18)	IT Consulting	Large	F
SOC_Analyst	SOC Analyst	1 (1)	IT Consulting	Large	F

As shown in Table 3.5, 11 subjects have been interviewed. Each subject has been given a pseudonym that matches their role within their respective organizations for easier reference in Chapter 4. Each organization has been described through its business area, with alphabetical distinctions, and

through its business size. This size is defined in the following manner; *Small* being fewer than 50 employees, *Medium* being between 50 and 249 employees, and *Large* being 250 or more employees. For the research, the interview subjects are from six different organizations. Every organization had one participant apart from the Technology and Manufacturing and the IT consulting (Company F), which had three and four individual participants. Each of the selected subjects inhabited a role that either had responsibilities in patching or vulnerability management, or both. Correspondingly, at a minimum, all the chosen organizations facilitated patching and vulnerability management services for some of their systems or offered solutions. Moreover, the experience of the interview subjects is signified through *YIR(YOE)*, which respectively means *Years in Role* in the organization and overall *Years of Experience*, denoting the relevant experience of the interview subject both in patch and vulnerability management and otherwise in their career.

3.3 Data Collection

Alsaawi (2014) describes four types of qualitative interviews; *Structured*, *unstructured*, *semi-structured*, and *focus group*. Conversely, while Qu and Dumay (2011) also recognizes structured, semi-structured, and unstructured as qualitative interview approaches, they instead put focus groups in a subcategory, where having interviews in focus groups is an additional choice. In the same way, Alsaawi (2014, p. 3) also describes focus group interviews as capable of being *"[...] structured, semi-structured or unstructured"*. For this study Qu and Dumay (2011)'s categorization of the qualitative interview approach is used.

Structured and unstructured interviews are opposite approaches to collecting qualitative data in many ways. Performing structured interviews includes considerable effort in the preparation phase, where interview questions are developed and followed tightly through the interview. This means that data collected from structured interviews contain information on particular themes and, therefore, could be more diverse (Alsaawi, 2014). By contrast, unstructured interviews will include some prepared questions, but the emphasis is on having a free-flowing conversation with the subject. As a result, the collected data from unstructured interviews can vary from interview subject to interview subject (Alsaawi, 2014). Semi-structured interviews incorporate elements from both the structured and unstructured approaches. Questions used within semi-structured interviews are planned, but conversation can stray from these questions if the interview requires it. Therefore, data collected using the semi-structured method can be mixed. Some parts will have the same form as structured interview data, while others will be without structure. The main advantageous characteristic of the semi-systematic approach is its flexibility, which allows the interviewer to gain insight that might be lost in the structured approach while still having general control over collected data (Qu & Dumay, 2011).

For our research, the most logical approach is the semi-structured approach. The gathered insight from the literature review provided a good starting point for formulating themes and developing interview questions. Nevertheless, while the literature review provided a good starting point, it

could have offered more conclusive statements about patching and vulnerability management that could be proven through quantitative or qualitative research. Instead, it explained what patching and vulnerability management involve within an organization. Therefore, following a structured approach might leave out information not first found in the literature. Conversely, following an unstructured approach would not use the insight collected from the literature. Therefore, following a semi-structured approach is the most suitable method as it allows for structure while still being open to deviation.

Interview Subject Consent

When performing research that includes the collection of data in Norway, a data management plan has to be developed to maintain the privacy of interview subjects. For the research, the data management plan has been filed and approved by NSD under the name *"Masteroppgave: Patching og Sårbarhetshåndtering"*. A part of this process was the development of a consent form which is included in Appendix B. This consent form allows the interview subjects to understand the interview process and how the collected data is handled. In addition, information about the interview subjects' authority to change any data collected is included. This process is facilitated through the interview subject being able to make any revisions to the transcribed interview before any data is used in the analysis. Therefore, the interview subject has the final say in the data available for analysis. This means that the interview subject can alter incorrect details mentioned in the interview and remove those that might reveal who the interview subject or their organization is.

Moreover, the consent form also explains what anonymization efforts are to be taken for the interview subjects to ensure that identification without context is as unlikely as possible. This is guaranteed by removing details deemed too much identifiable for the organization or the interview subject and by substituting information that is important for providing context but might be too recognizable with generic titles, such as the organizational and interview subject name, as shown in Table 3.5. The ethical considerations regarding anonymity are discussed further in Section 3.2.1.

3.3.1 Semi-Structured approach

When approaching the interview process in a semi-structured manner, some pre-planning has to be carried out to ensure that the interview process reaches the required outcome. Although the semi-structured methodology requires less planning than the structured approach, there still needs to be a thorough plan for conducting the interview. In other words, there needs to be a structured plan that ensures that the correct questions are asked and that the interview subject is comfortable with sharing information while still being open to going outside of the plan to discuss important themes in the interview process. Therefore, an interview guide had to be created to guarantee that our research met these requirements. To ensure that our research can fully account for the needed parts of an interview guide, we decided to follow Kallio et al. (2016)'s framework. Kallio et al. (2016)'s framework describes five phases for the development of a semi-structured interview guide and is depicted in Figure 3.3. While this framework provides a good starting point for developing the

interview guide, it does not govern how the interview approach is carried out. Instead, it provides a structured suggestion that should be followed when applicable.

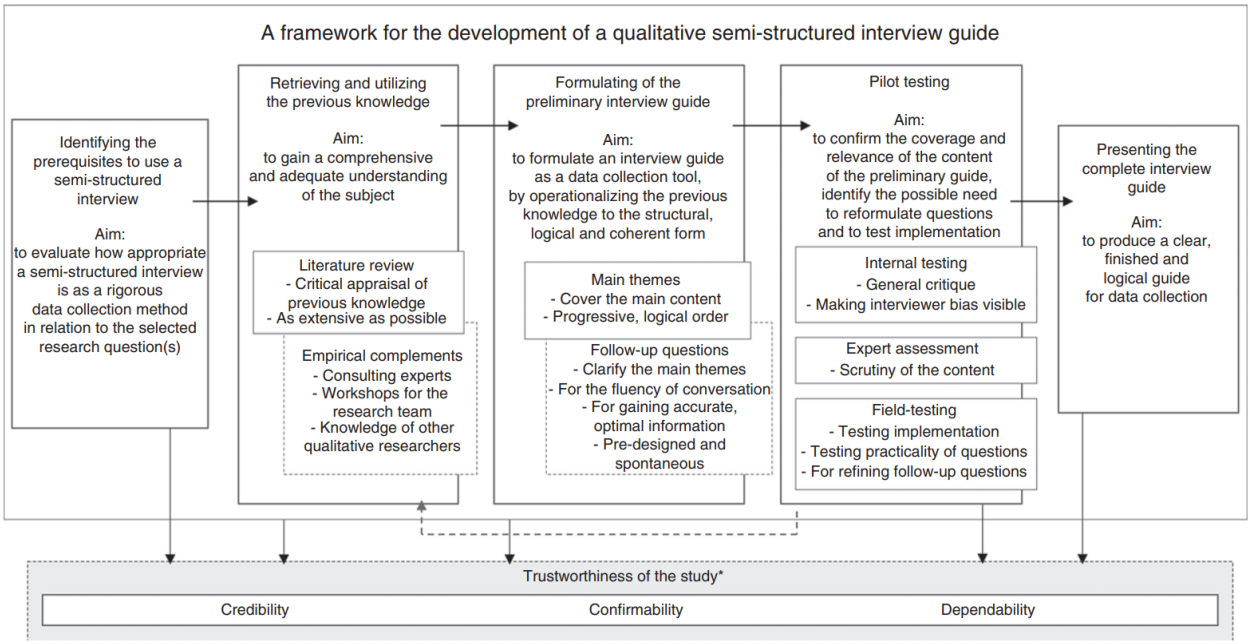

Figure 3.3: Kallio et al. (2016)'s framework for qualitative semi-structured interview guide

Following Kallio et al. (2016)'s framework shown in Figure 3.3, we were able to create an interview guide which is seen in Appendix A. In the framework, one of the main themes is that every phase is connected. Kallio et al. (2016, p. 8) describes the phases in the framework as *"The five phases were inter-related, as each phase contributed to the preparation and success of the next."* For example, the literature is gathered in the second phase to formulate the main themes and questions in the third phase. Likewise, choosing the semi-structured approach is deterministic for the literature collected. Consequently, every framework phase had to be treated in unison as they affected each other.

The first phase of the framework accounts for determining if the semi-structured interview approach is appropriate for the research. In this case, the information gathered in Chapter 2 aided in developing the research questions described in Section 3.2.1. Therefore, the applicability of the semi-structured approach can be based on whether or not the approach provides the needed answers to these questions. As discussed in Section 3.3, the approach is appropriate for the given research. For this research, the second phase of the framework is only realized through the literature review approach, which is thoroughly explored in Section 3.1, and through examining other semi-structured research as discussed in Section 3.2.1. The third phase of the framework was performed throughout the interview process, where the main themes were developed based on the information gathered in Chapter 2 and previous research conducted in earlier courses. Questions were created and changed based on responses in the earlier interviews, and the interview techniques were tweaked based on the interview subjects' responses.

Moreover, in the fourth phase of the framework, the main interview techniques and data collection

methods were tested and perfected during the initial interviews, with the reformulation of themes, questions, and perfection of interview techniques. The choice of not having specific testing interviews outside of the main interview processes came down to time constraints and the availability of interview subjects. The interview process was time-consuming, and scheduling pilot testing would impact the potential availability of the actual interview subjects. The final phase of the framework was realized through the development of the interview guide, which is seen in Appendix A.

3.3.2 Interview Transcriptions

As discussed in Section 3.2.1, the interviews were conducted in Norwegian or English. For the transcription processes, this means that depending on the interview subjects' preference, the transcription of the interviews was either in Norwegian or English. When quotes or data are used from the transcribed interviews, they are translated into English, depending on the original transcribed language.

3.4 Analysis

The analysis of the collected data is directly related to Kvale (2011)'s aforementioned *Analyzing*, *Verifying*, and *Reporting* steps. The *Verifying* stage is further discussed in Section 3.4.1. For the analysis of the data, Kvale (2011)'s *"Six Steps of Analysis"* is used. A short description of these six steps, along with their application of it in the research, is seen in Table 3.6.

Table 3.6: Application of Kvale (2011)'s *"Six steps of Analysis"* in the research

Steps	**Description**	**Application in Research**
Information about the subject	General information about the subject and their employer	Used to verify the experience and expertise of the interview subjects
Subject discovered relationships	Subjects identified relationships between different subjects during the interview. Where the subjects can describe similarities between two different subjects matters and elaborate on these subjects	Happened during the interview process, where the interview subject was able to elaborate on practices
The interviewee discovered relationships	Interviewees' identified relationships between different subjects during the interview. The interviewee is asked additional questions based on identified similarities in subject responses	Satisfied through the semi-structured nature of the interviews, where the interview guide was circumvented when interesting subjects were discovered that were not described there

Analyzation of recorded interview	Analyzation of the recorded interview by the interviewee. Further analysis of the transcribed interview and describing the interview subjects understanding of the subject matter	The recorded interviews were transcribed and analyzed. Through coding of themes and subsequent comparison between interview subjects. The insight that was absent during the interview was also brought forward by this process
Re-interview	Potential re-interview of the interview subject based on analysis of the recorded interview. Interview subjects can elaborate on their responses	Interview subjects were able to read through the transcribed interviews and change, elaborate, or remove any information they wanted
Subject action	Potential step, where subjects act on the insight that is found during the analysis process	Dependant on the interview subject. Outside of the scope of this research

The development of the different coding categories was done collectively by both researchers. The chosen categories were based on the interview questions in the interview guide (Appendix A) and their premeditated themes. Subsequently, the data was coded individually between the researchers, where each researcher coded half of the interviews. Therefore, to group concepts and ideas discussed by the interview subjects, the NVivo 12 tool was utilized. Moreover, this allowed for comprehensible extraction and analysis of the interview subject's responses. Four main categories were chosen to extract and code the different themes and ideas, representing the interview categories in the guide, as seen in Appendix A.

Additionally, one extra category labeled *"Miscellaneous"* was added to account for any added insight from interview subjects that did not fit into any of the other categories but still provided useful information. The coding categories are seen in Figure 3.4, categorizing the interviewed subjects' responses to simplify finding similarities and differences.

Name	Files	References
1 Background	0	0
1 General Information	10	10
2 Patching and vulnerability experience	10	12
3 Resources	9	10
4 Framework Necessity	8	8
2 Routines and processes	0	0
1 Internal documentation or routines	10	13
2 Information Sources	10	14
3 Vulnerability Prioritization	9	10
4 Testing	1	2
5 Misconfiguration	8	9
3 Challenges	0	0
1 Patch-related	9	9
2 Vulnerability-related	7	7
3 Patching delays + unmitigated	8	8
4 Framework	0	0
1 Framework Applicability	9	9
2 Change	9	15
3 Incomprehensible aspects	6	6
5 Miscellaneous	6	12

Figure 3.4: Coding categories in NVivo 12

The relationship between the interview subjects' responses was identified after gathering the individual themes in different NVivo categories. This process included finding similarities and differences in practice and their subsequent correlations or contrasts with the literature applied in the conceptual framework presented in Section 4.1, further developing the framework by incorporating the interview subjects' expertise with the established literature.

3.4.1 Verification

The primary method for verification of the gathered information is through the consensus of the different interview subjects' insights. This means verifying what the interview subjects report is done by comparing it to other sources between different organizations and within the same organization. This methodology ensures that what the interview subjects say is accurate compared with other interview subjects and increases validity through a general consensus.

3.5 Ethical Considerations

Interviewing subjects within patching and vulnerability management areas creates specific ethical factors that must be evaluated and accounted for. Moreover, information within this subject area is often confidential and can cause real financial damage to organizations if divulged. Therefore, there are limitations on what can be discussed within interviews and what information can be used within this research. As a result, special care has to be taken when asking about sensitive subjects, as discussed in Section 3.2.1. For example, asking about specific policies or procedures might reveal confidential information, while asking about the existence of said policies or procedures

does not. Therefore, the interview process requires a delicate balance between pursuing fascinating information and acting ethically toward the interview subject. Throughout the interview process, the practice of informed consent is also critical and has to be accounted for. The interview subjects will be informed of how data collected in the interview process is stored, analyzed, and used in the consent form in Appendix B. Ethically, this consent has to be verified throughout the interview process and is satisfied through the interview subjects' opportunity to change or remove anything from the transcribed interview before the analysis as described in Section 3.3.

4 | Findings

Following a semi-systematic literature review and semi-structured qualitative interviews, the findings first present the formulation of a conceptual model based on the gathered literature review and the empirical findings gathered through interviews. The findings present the most important empirical evidence and are divided into categories to depict differences and similarities between the interviewed professionals. Subsequently, the findings and the conceptual model are synthesized in developing the comprehensive model discussed in Chapter 5.

4.1 The Conceptual Framework

Ensuing a semi-systematic literature review and interpretations of the literature, the conceptual model is constructed regarding the over-arching processes of patching and vulnerability management. The conceptual model aims to depict the ideas and the understanding of the topic area with a theoretical background.

The conceptual framework, as depicted in Figure 4.1, is constructed with carefully assessed literature, combined with a theoretical understanding of the current processes within patch management and vulnerability management. Additionally, the model is rationalized into three primary levels: the strategic level, the tactical level, and the operational level. These three levels form the basis of our understanding of how each process works in each main branch based on the understanding of the assessed literature. As such, the proposed conceptual framework builds on certain prevalent pieces of literature defining processes and practices, such as Huang et al. (2012)'s *Patch Management Process Workflow* as shown in Figure 2.1, along with Li et al. (2019)'s and Dissanayake, Zahedi, et al. (2022)'s elucidation on the stages of patch management. Moreover, Li et al. (2019)'s *Sources used for discovering available updates* as shown in Figure 2.4 is utilized as inspiration for identifying the most common and viable information sources for gathering patching and vulnerability information. Lastly, Bautista (2018)'s *Cyber Intel Levels* as shown in Figure 2.5 is utilized as a foundation for dividing the responsibility areas into the strategic level, the tactical level, and the operational level.

Ultimately, the framework aims to depict the processes within an organization when performing patching and vulnerability management. Therefore, the framework itself is divided into specific main categories, with each category being in one way or another related to other key aspects of the processes. Additionally, the framework is divided into three over-arching areas of responsibility

that depict the vital processes performed in each area. The areas are constructed based on existing literature regarding the prevalent processes and a subjective understanding of each category's logical and efficient placement. The three levels of leadership are the strategic, tactical, and operational levels. Subsequently, different information sources utilized (as inspired from Li et al. (2019) and Dissanayake, Zahedi, et al. (2022)) work concerning how information is processed and utilized at the operational level.

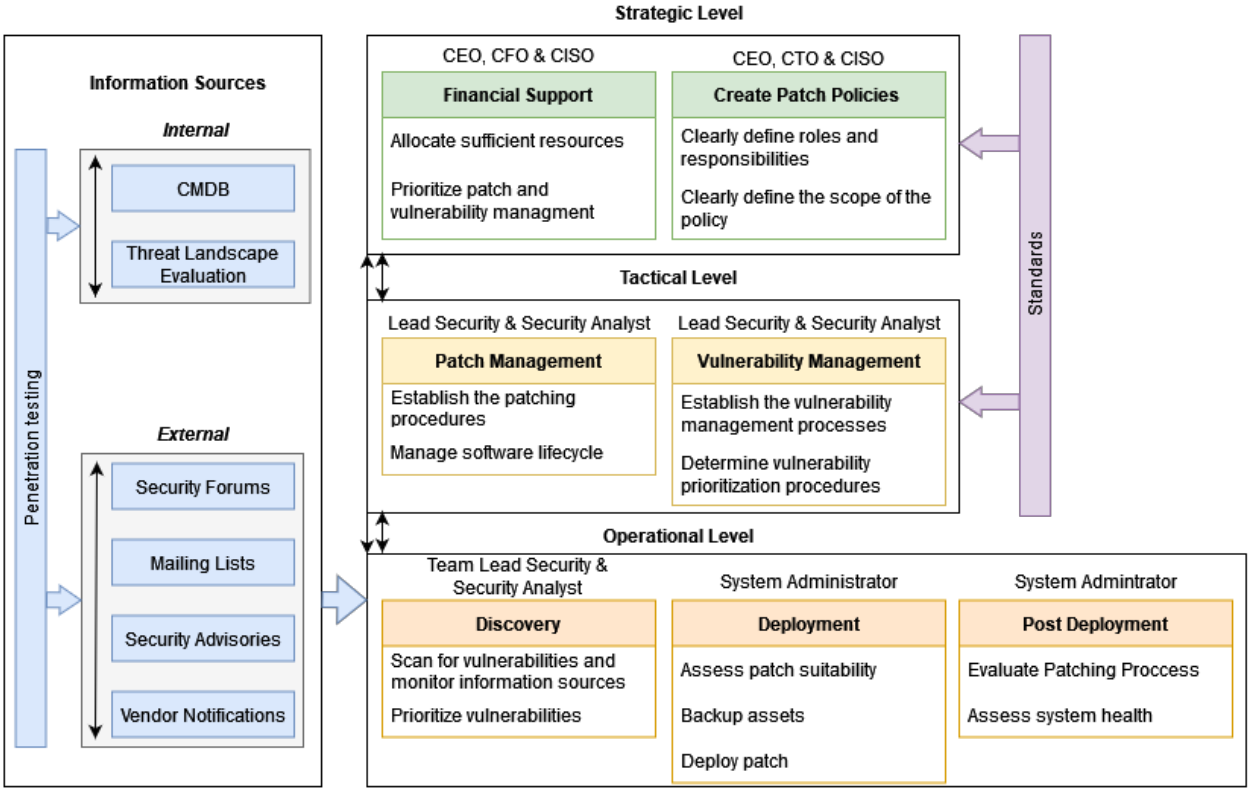

Figure 4.1: A Conceptual Model for Patching and Vulnerability Management

The model comprises several aspects that, in unity, form the patching and vulnerability management processes. The main parts of the model, depicted in green, yellow, and orange, depict the levels of leadership as described by Bautista (2018) and function as an over-arching guide on what tasks each area is responsible for. These three levels depict the difference in technical levels, top-down, where the top indicates less technical and becomes gradually more technical as one moves down. The black arrows between each responsibility category represent information flow in the form of patch policy creation, patch procedure establishment, approval of procedures, vulnerability management information, and patch process evaluation. The arrows are bi-directional, meaning information flows up and down the chain for continuous communication and approval across all responsibility areas. Furthermore, as depicted in the purple box, standards heavily influence the actions of the patching and vulnerability management processes exercised at the strategic and tactical levels. Thus, standards such as ISO 27001 (patch and vulnerability management), NIST, and laws regarding the GDPR influence the rigorous descriptions of the patching and vulnerability management processes.

Moreover, the blue segment depicts the information sources influencing and aiding the practitioners in identifying relevant patches and threat information. The external and internal information sources are gathered in correlation with Li et al. (2019)'s results when interviewing system administrators for their primary information sources. These sources collectively build on manual information streams (e.g., security forums, threat landscape evaluation) and automatic/semi-automatic information (e.g., vendor notifications, system scanners). The black communication arrows bind these sources and signify that information traverses through different information mediums and influences one another. Additionally, penetration testing is the most invasive and intrusive potential for information gathering as it may expose security misconfigurations and security flaws within the company's systems. It acts as a resource for external and internal information sources as it can be performed by the company's team or hired externally. The information sources initially influence the operational level, as indicated by the blue arrow, where information from these sources is fed to aid the *Discovery* within the operational level in identifying and prioritizing vulnerabilities.

The strategic level that comprises the C-level management is a significant contributor in allocating sufficient funds and resources towards the tactical and operational level and is responsible for creating relevant patch policies with clearly defined scope, roles, and responsibilities. Moreover, the tactical level works closely with the strategic level, where information flows bidirectionally, as the black communication arrows signify.

4.2 Empirical Findings

The empirical findings utilized in the report are substantiated by the findings made in the interviews conducted. Subsequently, the findings are categorized into different areas of interest and mapped accordingly for precise usage in developing the finished framework.

4.2.1 Information Sources

Collecting information is a significant part of patching and vulnerability management, and it is here that system administrators can get an overview of what needs to be patched and how prevalent specific threats are in the emerging threat landscape. The interviewees each practiced and undertook gathering information from information sources differently, where the majority of objects utilized the sources specified in the conceptual model.

From the interviews, several interview objects utilized the same information sources and information-gathering techniques to ensure their systems could be deployed with the right patch. Consequently, the employees working in larger companies were more likely to get patching and vulnerability information from a dedicated department within the organization as the business had sufficient resources to allocate. Therefore, the patch information was not necessarily gathered by the employees on the operational level who implemented the patches. Additionally, the interviewees working at larger organizations utilized tailored patching systems, contributing to patch recommendations and deployment alternatives. As *Infra_Engineer_2* describes:

> *"For example, it's very rare that we will go to Microsoft themselves and download the patches directly, but we can do it if we want to, but we generally don't. Everything comes in from [HCL] BigFix. [...] We only pull our sources in, the actual patches themselves, from one source via Big Fix, the patching tool. They take all the patches from Microsoft, package them, and then supply them to us or to Big Fix."* – Infra_Engineer_2

Additionally, *Lead_Application* and *Infra_Engineer_1* collaborate on the same team to implement patches in the same company (Company F) as *Infra_Engineer_2*, where they further describe the collection of information from their security department:

> *"We have the operational responsibility for the services we run, and the security team has the responsibility of uncovering the security flaws and informing us about them. The principle is that they detect and inform vulnerabilities towards us and other customers. The information is received several different ways, but the principle is that we are just receivers of the information."* – Lead_Application

Consequently, the patching and vulnerability information is mainly collected through automated means correlated to a patching system or through a dedicated security department that feeds relevant information to the operational level. *Infra_Engineer_1* continues:

> *"The way I do the patching itself is that I read relevant information before I implement a patch, in forums online and such, from people who push this out the day it arrives. Many people do that. I read about the experiences and discoveries they have made before we tackle our environment. I also utilize a source I know many others also use, which is /sysadmin on Reddit. They refer to Microsoft and often link their explanation towards them [Microsoft]"* – Infra_Engineer_1

The patching practitioners all undertake information collection differently and mainly receive information from their security department. *SOC_Analyst* works as a security analyst and collects and handles threat information which is subsequently sent to the patching department. *SOC_Analyst* describes the different information sources utilized and emphasizes that the company utilizes both manual information gathering, in addition to automatic systems like Qualys and Security Scorecard to gain information about threats in their network:

> *"Almost all of our customers use Microsoft, so when a patch Tuesday arrives, we collect and review the most critical vulnerabilities and assess if they should be patched instantly or during the workday. [...] We also follow CISA's list of vulnerabilities. They continuously follow the most exploited vulnerabilities at any given time. Additionally, we have our tools, Qualys and Security Scorecard, which are systems that scan the network internally and externally for vulnerabilities."* – SOC_Analyst

Contrarily, *Lead_Security*, which works for Company B, states that larger companies tend not to have time to perform large-scale manual information gathering unless there are sufficient resources allocated in the security department to perform such tasks:

> *"Unless the organization is very large with suitable investment in the security department, they don't have time to be going all over the place. They will rather find tools that collate that information into feeds; quite often, some of that information comes through their vulnerability management solution, ServiceNow, or other tools."* – Lead_Security

Sec_Engineer_1 from Company D validates the responses from interviewees from Company F, where they utilize several vulnerability scanners that perform threat analysis on the network. *Sec_Engineer_1* also states that they collect patching and vulnerability information from a dedicated security department that provides an overview of the current threat landscape. They also utilize certain newsletters and mailing lists to ensure they are updated on the latest vulnerabilities and threats relevant to their systems. *Sec_Engineer_1* explains:

> *"We use vulnerability scanners both internally to scan the internal systems we use, and vulnerability scanners externally to scan the systems we have exposed to the Internet. Our security department utilizes lots of external resources from which they get information. This department reports to our department when they get notifications of relevant vulnerabilities requiring further action to follow the normal routines. The system we have chosen for our vulnerability scans, both externally and internally, is Tenable. Additionally, I also know that the security department gets information from the national CERTS. In our case, we get daily newsletters from Telenor, Mnemonic, and SANS."* – Sec_Engineer_1

Back_Developer at Company A works closely with the security of the systems and code-related tasks and emphasizes that their approach to dealing with security differs from *"traditional"* and larger enterprises as they work at a startup with limited routines regarding security and patching. Additionally, *Back_Developer* states that the usage of web-based forums for gathering information is a valuable resource; however, they do not follow recommendations from Microsoft's patch Tuesday:

> *"We use forums, but not necessarily patch Tuesday from Microsoft. For example, you have a /sysadmin subreddit which can be valuable. However, that is mostly valuable when things go down on a global scale. [...] We do not necessarily follow the recommendations from NSM and such since they will be mostly targeted to more traditional enterprises. [...] as a startup, will my department, the development department, focus on our infrastructure, our laptops, and that our network is secure."* – Back_Developer

4.2.2 Internal documentation and policies

Internal documentation and policy creation is an area that aids in clearly describing and outlining the different responsibilities and necessities correlated to a given task area. In addition, the policies typically influence the procedures, which again influence the creation and usage of documentation, where each company may perform it differently based on their policies. Accordingly, internal documentation and policy creation affect the strategic, tactical, and operational level as it is all interdependent.

Several companies utilize patching tools to aid in patching and vulnerability prioritization, while others additionally perform manual labor to ensure all critical vulnerabilities are covered. *Infra_Engineer_2* explains that having thorough documentation for how each process is executed and undertaken is hugely time-consuming, and the majority of the documentation and processes are covered through the unique patching system. As of now, the company only has high-level documentation that describes the strategy and general approach; however, there is a lack of documentation regarding how to utilize the systems and undertake the processes:

> *"It has been an ongoing task, but we have a high-level document in how we are set up. It is a broad idea, like a design document on how patching is done, how we do it, and why we do it. But as far as I know, we don't have a document with BigFix for exactly how we do it. It is a lot of work, and we just never have time to get to it. We have documents that say that these services are patched at this time. We just don't have: "click this button to release patch one." We haven't written that yet because it is a lot of writing."* – Infra_Engineer_2

When asked if there is a need for documentation and clearly stated routines, *Infra_Engineer_2* substantiates:

> *"Yes, there is. If I'm not here, no one knows how to patch. Now you will need the documentation. The last six weeks, we have tried to describe and write documentation because for me to train someone, I need to give them something to work with."* – Infra_Engineer_2

Additionally, *Lead_Security* in Company B validates that, generally, businesses might have documents and routines that elaborate on the over-arching nature of the specific area and that there should be allocated time to generate high-quality documentation which is made for the specific system:

> *"Businesses do often have this documented. I think one thing that causes the most significant issue is that these documents are often very generalized, and ultimately you need to have a more specific patch/vulnerability mitigation documentation around individual systems of interest. Often they need to come up with their own adjusted prioritization and scheduling. This just does not work quite often, and many IT support desks have been burnt by applying day-one patches, so this often leads to a one/two weeks delay."* – Lead_Security

Businesses all have different needs when it comes to having clearly stated documentation and routines. Operational-critical tasks might increase the chances of needing documentation to ensure all tasks are performed according to the policy and procedure, while some tasks might be performed by utilizing existing knowledge from the employees in the enterprise. *Sec_Officer* explains that in Company E, some developed routines explain the overall processes, which are built according to the existing policies in the enterprise. However, documentation is scarce as the IT department is busy, and calls for knowledge from the employees to fill in the gaps of the missing routines and

procedures:

> *"It is somewhat divided. We possess some routines that pertain more to cybersecurity, which I might personally document to some extent. Additionally, we have a number of policies atop these routines, which define the requirements at a minimum. The IT department is frequently overburdened and engaged in putting out fires rather than documenting sound processes or procedures for handling such situations. This is a matter we are progressively addressing; we identify a problem, elevate it, and then document a corresponding process or routine. It is not always in place, and at times we must fill gaps with our own knowledge."* – Sec_Officer

Contrary to Company E, which has some written documentation developed by the IT department and *Sec_Officer*, Company A's *Back_Developer* does not currently enforce documentation or routines as there are not enough employees undertaking these tasks yet. However, a few documents were created highlighting tasks that are rarely done compared to routine operational tasks. When asked if the enterprise has any internal documents, routines, and processes, *Back_Developer* substantiated:

> *"Not really. There are not enough employees dedicated to the infrastructure, as there is only me at the moment. It is up to me to ensure things are done when handling patching on the service-related part, and it is up to me to ensure there are routines that are written. Currently, I think there is a maximum of 10 documents describing routines and matters that describe "to do" lists concerning security and patching specifically."* – Back_Developer

There seems to be an apparent difference between smaller and larger enterprises, where smaller enterprises tend to have fewer resources available in the security department, thus resulting in it being less developed than larger enterprises with sufficient resources. Additionally, larger companies tend to have separate security departments correlating to patching and vulnerability management. *Lead_IT* is the manager of the patching department in Company D and substantiates that they inhabit documentation that both explains the *"what"* and *"when"* regarding patching, in addition to that there is a distinct operational documentation which describes how a task should be technically performed:

> *"We do not have routines on how, but on what and when, which state that patching should be performed within the given requirements. We have internal processes which state that they must be followed. How something is to be done technically lies more in our operational documentation. Those are not official documents, only how to appropriately do it."* – Lead_IT

Contrarily, *Infra_Advisor*, which works with development and operationalized tasks in Company C, has no specific documentation or routines for patching. However, their mentality is to shorten the time it takes a code change to reach their production environment to ensure the systems are continuously updated. Contrary to the other companies, there is complete trust towards the developers of the services that they perform in the best manner, even if that results in vulnerabilities

being made locally:

> *"We have had focus groups around NSM fundamental principles, but we focus more on that it should be swift from code change to production. We commit directly to main, we don't have much pull request regime, so it is more that we trust the employees that they do the right things and dare to have enough testing around it. If we create a vulnerability ourselves, a logical code error, we trust the developers to do the best they can and ask if they are unsure."* – Infra_Advisor

Subsequently, *Infra_Advisor* believes that having less documentation results in the enablement of more understanding of a system by trial and error:

> *"We are trying to learn the techniques and technologies, so if anyone needs documentation, they write it for their own sake. People, especially graduates, would expect there to be more documentation, but it is also a good way of getting to know others in the team and getting to know the solutions by not having documentation so that they have to ask questions."* – Infra_Advisor

4.2.3 Patch-related Challenges

The patch-related challenges findings revolve around the findings within specific areas of patching and challenges related to the patching operation. Additionally, it substantiates the patch testing formed by the various interview subjects.

Given the framework's aimed contribution of effectivizing the patching process, it is essential to look at the challenges the interview subjects could identify regarding the patching process, thereby ensuring that the identified challenges are accounted for in the comprehensive framework. Moreover, as the patching process involves many different moving parts, diverging responsibilities, and sentiments, the interview subjects identified several challenges involved with the patching processes.

One challenge that *Lead_Security* mentioned was the difficulty in controlling assets, precisely the configuration of assets. In other words, ensuring that the configuration management database (CMDB) is up to date and has the needed information for accurately identifying what assets should be patched. *Lead_Security* pointed out that inaccurate information might mislead the patching process, affecting what assets get patched and, more importantly, what assets do not. This demonstrates the importance of having control over both the assets present in the organization and the configuration of those assets:

> *"One thing I can say on patch issues is that CMDB being up to date very often an issue, or the CMDB not containing sufficient information to ensure that patch deployment is following the proper prioritization. You will often find that organizations say they are fully patched on OpenSSL, for example, and when they miss some services, it is simply because they were not properly added to the CMDB correctly."* – Lead_Security

In the same way, when discussing the challenge of finding out who owns an asset, *Sec_Officer* identified the solution as controlling asset management, describing the need for control over the organization's assets and understanding how it is organized. This reaffirms the challenge pointed out by *Lead_Security*:

> *"First and foremost, establishing asset management with what assets we actually have along with ownership of those assets. The next step is to get control of how the assets are connected and how they communicate."* – Sec_Officer

Furthermore, *Sec_Officer* also identifies the challenge of communicating in the patching procedure and ensuring that responsibility is assigned and understood between different groups of employees. Pointing out the observation that in patching, where different employees might be in charge of identifying the need for a patch, scheduling the patching itself, and patching the asset, there is the potential for a failure in communication. Therefore, there is a misunderstanding of where the different responsibilities lie and a tendency to slow down or stop the patching processes:

> *"It goes back to something I mentioned earlier: If an asset owner expects IT to handle it [patching] and IT expects that the asset owner requests the update, then it ends up being a loop where nothing gets done. [...] I know both from my own earlier experience and from talking with coworkers who have started working for Company E that this is a recurring challenge."* – Sec_Officer

Moreover, *Sec_Officer* points out that the solution to this challenge would be better communication and someone being in charge of coordinating the patching effort, ensuring that someone is keeping track of the patching processes for the assets and facilitating communication between the different parties. Furthermore, having someone tasked with coordinating the different groupings of employees ensures that everything is accounted for and that operations flow as smoothly as possible.

There are also similarities in the challenges affecting different parts of patching. For example, patching assets such as servers and infrastructure encounters similar challenges to patching workstations or provided work phones in the sense that the challenge impedes the user's time. The affected party is the users, whether infrastructure or workstations are getting patched, and whether there are interruptions to delivery or the time needed for patching. When discussing the need to manage people's expectations regarding patching, *Lead_IT* elaborated on the challenge of how there might be conflicting interests regarding the patching schedule. This was mainly regarding the time slot for patching, as some slots might work better for some parts of the organization, and conversely, some other slots might work better for other parts:

> *"Somebody wants something done at 11:00, and somebody else wants something at 1:00, and somebody wants something at 1:00 AM, you do realize that this is not like a Microsoft support center here, it's just a couple of people working on this."* – Infra_Engineer_2

Similarly, *Lead_IT* also mentioned people's general disposition on patching being a hassle, where nobody wants to restart their computer as it interrupts other tasks. Given the importance of

patching, it might be impossible to satisfy all users:

> *"On a human level, people do not like to patch because they do not like to restart their computer."* – Lead_IT

Moreover, *Lead_IT* also elaborates on the challenge of patching scheduling with employees that work in places where patching is not easily done, such as on ships or other remote locations. In those cases, there might be no service or limited bandwidth, requiring an even stricter policy and employees to be reliable and perform patching when available. Ensuring that the employees are appropriately educated about the risks associated with not installing new patches ultimately influences the prioritization of patching among employees. *Lead_Security* also points out that patching workstations is often the biggest challenge for smaller organizations and that, given their size, they have to rely on employees for updating workstations, pointing out the recent *LastPass* incident and how lacking patches on private workstations was exploited. This confirms that patching on workstations, which should be a straightforward affair, can potentially lead to security incidents:

> *"When you look at small companies that don't have central IT solutions available, their biggest challenge is ensuring that workstations are patched. So they are reliant on their own employees to ensure that this is done. You can even look at the case of "LastPass" recently, where it was even a developer's personal machine where he had used credentials, and the lack of that system being patched was actually one of the entry points."* – Lead_Security

Lead_Security also elaborates on why patch deployment delays may happen in enterprises and substantiates that having proper testing and sufficient knowledge aids in ensuring patches are timely deployed. *Lead_Security* also mentions that CMDBs are vital for system administrators as they allow a centralized platform with an overview of the organization's configurations on their hardware and software. This again may lead to a more efficient patching process and aid in decreasing the chances of patching delays occurring:

> *"Every single case is generally down to numerous factors. One can just be the staff's capability to perform the operations, depending on the size of the organization. A factor can be just the lack of availability to test a patch. Not all patches can be smoothly reversed. Quite a common item is systems just getting missed, which is down to a lack of sufficient knowledge of all the systems in the environment, making CMDBs very critical. CMDB really needs to be implemented as automated as possible."* – Lead_Security

When asked about what an organization should do if there are patches that should be implemented but can not be deployed due to specific reasons, *Lead_Security* addresses that the system should be shut down as it ties closely with a statement to top management, signalizing a better business awareness environment. Additionally, it ties in with risk management, as running an unpatched system poses a security risk for an enterprise:

> *"I would walk up to the owner of the system and tell them that we are shutting it down.*

> *It is not always the nicest, but it generally makes the owner scream and shout, and it feeds up to the top level; there is generally business awareness. When we look at this from the business process's ultimate point of view, it is back to risk management. And when you are looking at that, it is only about the business impact; it is about where could this system from being compromised, what would be exposed, or what could the attackers pivot to?"* – Lead_Security

Contrarily, *Lead_Application* in Company F substantiates that they do not perform specific measures if there is an unavailable patch for a system, nor do they have any requirements from the company to respond to such incidents:

> *"We wait until we get a patch; that is what's practically happening. We do not take any other measures to protect it [the system] in the period they are vulnerable. We also don't have any requirements either to treat those types of incidents you are addressing."* – Lead_Application

4.2.4 Patch Testing

Albeit patch testing is not originally a part of the conceptual framework, additional feedback from the interviews (presented in Section 4.2.7) and literature search substantiated the urgency for patch testing as a part of the comprehensive framework. Patch testing relates to testing patches on similar systems before they are deployed to limit downtime and potential security vulnerabilities when the patch is deployed. *Lead_Application* in Company F substantiates the importance of thoroughly testing patches to ensure the systems respond satisfyingly to the implemented patches. Consequently, not testing may cause unwanted consequences and unknown behavior in the system:

> *"We cannot just push out patches and hope that it works. It will have consequences, especially in regard to downtime on the system, and suddenly you might get unintentional side effects. You have to do it within the test environment first. Some vulnerabilities are harder to account for than others, and this is something we pay attention to."* – Lead_Application

Infra_Engineer_2, which works with implementing patches for a customer on behalf of Company F, elaborates on the lack of a proper patch testing environment to test patches. *Infra_Engineer_2* mentions that the environment is inadequate because the testing environment provided differs vastly from the actual system environment in which the patches are being installed and deployed. This creates a factor of not knowing how patches will influence the original system as it is not tested to handle the production system's load and programs:

> *"There are various ways we do this, but the test environment is very limited, and I mean VERY limited. We have maybe 70 servers that are in the test environment, but it is a very lightweight reproduction of the main product, and it's not really usable in any way. We roll out of that, and we check to make sure everything went OK. Nine times out of 10, there is never a problem. It always works. However, it does not always work in prod*

[production]. We do have testing; it's just not great." – Infra_Engineer_2

Lead_Security explains that testing becomes increasingly essential to do on patches implemented in a system as the environment is growing. Additionally, *Lead_Security* states that a lack of a suitable environment is one of the reasons why, in general, companies may find difficulties with testing their patches before deploying them. The interviewee introduces a new and modern strategy named *"don't patch, redeploy"* [DR tests], which revolves around deploying a newly patched system instead of updating the old one:

> *"The higher the demand on the system, the more critical there is that there is time available for testing patches, which unfortunately is not possible for absolutely every case. It is because they just do not have a suitable environment to test that patch to see if it impacts anything. By doing DR tests, nobody is faced with it. They just deploy, and it is tested. They have a scheduled occurrence to just redeploy the entire solution."* – Lead_Security

When asked if *Lead_Security* would encourage this type of patching approach, they answered that while being effective, it is not the right approach for every business. Moreover, *Lead_Security* substantiates that it is the CISO's responsibility to allocate sufficient funds for patch testing to be prioritized:

> *"I definitely would, but I would say that while this is a very interesting route to take, it is not a fit for every business, but it definitely comes with a lot of benefits. Every organization should consider it. [...] It is the CISO that is responsible for ensuring to get suitable funds, but it needs to be prioritized. It is actually the system owners' responsibility to provide test environments."* – Lead_Security

4.2.5 Vulnerability Prioritization

The vulnerability prioritization section describes findings about vulnerability prioritization and how it relates to vulnerability management. By focusing on how the different organizations prioritize vulnerability handling and mitigation, enterprises may gain insight into their own processes.

As there is a finite amount of available resources for patching, both financially and in terms of working hours, the vulnerability prioritization process is an important step in the overall patching effort of an organization, as there is not enough time to follow a *"patch everything"* approach. Additionally, when an organization grows in size, there is always going to be an increased risk:

> *"The reality is that a business only grows by accepting a certain level of risk. The higher the risk, the higher the potential development and growth."* – Lead_Security

Lead_Security describes risk acceptance as a byproduct of growth, primarily exemplified in the vulnerability management process, as more assets generally mean a bigger attack surface. This, in turn, means more potentially vulnerable assets. The organizations interviewed in this research each followed some vulnerability prioritization methodology to mitigate vulnerabilities as appropriate.

This proves the need for its inclusion in the vulnerability management box in Figure 4.1.

There were methodological differences in how vulnerability prioritization was handled in the different organizations interviewed. The majority of organizations, such as Companies B, C, D, E, and F, used specific tools in the prioritization processes. For example, when asked what kind of framework was used in Company B to prioritize, *Lead_Security* responded that they did not use any *"off-the-shelves"* framework. Instead, they used their vulnerability management within the ServiceNow solution along with their configuration management database:

> *"Generally, we have not utilized any "off-the-shelves" frameworks for this. What has been used quite often is a vulnerability management solution within ServiceNow, but for that, it requires the CMDB to be maintained correctly, which has been the most cohesive solution when you are able to produce reports in a tool that provides the CVSS rating along with the prioritization per technical item. [...] When it comes to general frameworks, they are all good starting points, but all of them need to be taken as a draft, and it needs to be developed with the business."* – Lead_Security

Additionally, *Lead_Security* also pointed out the limitation of these frameworks, looking at them as starting points rather than end-all solutions. Company C also uses tools to manage prioritization. *Infra_Advisor* stated that they used a tool called *"dependabot"* to understand the severity of vulnerabilities:

> *""Dependabot" gives us some hints into how severe it [vulnerability] is, and that affects how we approach it."* – Infra_Advisor

Sec_Engineer_2 also mentioned that Company D uses internal scoring within their *"Tenable"* tools. Similarly, *Lead_Application* at Company F, which inhabits a more operational role in the organization, stated that classification of vulnerabilities with scoring was done automatically in their *"BigFix"* tool. Moreover, *Lead_Application* also mentioned that while this tool sorted the vulnerabilities, they did not prioritize based on this:

> *"They [Vulnerabilities] are classified in BigFix. They are sorted, but we do not have a fixed order as we have a window for when we patch, and of course, testing on the testing platform has to be done first. So it's not like something gets handled one day and other things on another day. Security patching is fixed, day or night."* – Lead_Application

Besides, *Infra_Engineer_2*, which works in Company F in a different department, mentioned that certain patches take precedence. They highlighted the service stacking updates as critical, given their role in making the overall patching processes easier:

> *"[...] service stacking patches is a number one priority because they actually influence how the patches will actually install and how smooth it is."* – Infra_Engineer_2

Furthermore, as well as the operational patching perspective provided by *Lead_Application* and *Infra_Engineer_2*, *Sec_Officer* provided a more security and vulnerability management viewpoint

for Company E. Comparably as with the operational part of the organization, the security department also used a tool for vulnerability prioritization. Conversely, they did not use the same tool but instead used *"Qualys"*, which provides risk scores:

> *"Qualys, which automatically discovers vulnerabilities, has an in-built risk score for vulnerabilities and assets to help customers prioritize. You might have a low criticality vulnerability in CVSS, which according to Qualys is, in reality, a higher risk."* – Sec_Officer

Alternatively, some organizations interviewed, like companies D and E, use CVSS scores as part of their vulnerability prioritization. As presented earlier, Company D, uses internal *"Tenable"* vulnerability scores for their *"Tenable"* tools, but *Sec_Engineer_2* states that their main prioritization comes from CVSS scores:

> *"It [vulnerability prioritization] is not very thoroughly described in any processes yet; we have not come that far. But we mainly use CVSS [...]"* – Sec_Engineer_2

Similarly, *Sec_Officer* specifies that their main prioritization comes through the usage of CVSS and the vulnerability's potential criticality. *Sec_Officer* points out the important aspects that affect criticality, like the importance of the asset and specific details about the vulnerability:

> *"Mainly it is based on criticality and CVSS which gives a foundation. [...] When it comes to internal assets, where there are numerous servers and workstations, it will vary based on how important the system is and details about the vulnerability."* – Sec_Officer

Elaborating on their usage of CVSS, *Sec_Engineer_2* specifies that while CVSS is used, it is not used as a standalone solution. Instead, the inclusion of a technical description of the vulnerability with the CVSS score is used for better prioritization:

> *"Mostly yes, but there are some things that are not accounted for in CVSS which we often looked at anyway. CVSS comes with a technical description of how the vulnerability can be exploited. [...] CVSS in itself works fair enough, but the technical explanation of how the vulnerability works are important."* – Sec_Engineer_2

Similarly, *Lead_Security* also points out the limitations of the CVSS scoring system, thus demonstrating that while a vulnerability might be rated highly using CVSS, assessing the potential impact of the vulnerability is also down to where the vulnerability is present:

> *"Yes, really, the prioritization that is given to a patch is great for providing the sensitivity importance to weight behind the specific patch, and organizations that follow that blindly actually tend to not do as well. The largest factor needs to be added, which is that you have to have your own prioritization assigned per system. The prioritization for a CVSS of 9 for a system that is air-gapped and the patches are nine should not necessarily be prioritized over a web server that is not suitably protected. So, CMDB needs to be applied against this."* – Lead_Security

Conversely, Company A went for a more manual approach compared to how all the other organizations facilitated vulnerability prioritization. This may be based on the fact that their infrastructure is smaller as they are a smaller organization:

> *"It [prioritization] is a little bit manual for us, simply because our infrastructure is defined through ".YAML" files mainly. So the work goes into identifying parts which need updating and roll out in a secure manner, in a testable environment."* – Back_Developer

In their responses, *Lead_Security* provided valuable insight into how vulnerabilities should be prioritized. Moreover, *Lead_Security* was also adamant about the importance of tailoring prioritization to the specific organization and pointed out how different systems and assets across organizations and within the same organization have different requirements. Furthermore, *Lead_Security*'s responses show that the actual scope and available patching slots are an essential decider on how and what patching is done within an organization.

> *"It [Tailoring] has to be. [...] Vulnerability management is exactly the same; for example, deploying patches cannot avoid impacting production; however, deploying in an environment where a business is entirely reliant on 24/7 operation means that it is a cost to the business. Whereas, a business that is predominantly 9 to 5, general hours, you have this beautiful window every day, and normal weekends, where you can basically impact those systems. If you want to take them down for 3 hours and if there is an issue that extends it to 6 hours, that can be done without impacting the business. But that is not the case if the business is reliant 24/7."* – Lead_Security

A common trend across the interviewed organizations is using tools or frameworks in vulnerability prioritization. Most organizations used some automated tool that handled other parts of the patch and vulnerability management processes to manage prioritization. Another commonality was using CVSS as a framework and ingraining it into other tools. One outlier is Company A, which handled prioritization more manually with some self-developed tools. Common among all the organizations that used some framework or tool is their relative size being the larger size, with only Company C being the exception. Conversely, the only organization to perform mainly manual assessment is also the smallest one, Company A.

4.2.6 Vulnerability-related Challenges

For all organizations, there are bound to be challenges associated with vulnerability management. Vulnerability management responds to the inert challenges faced when keeping IT assets secure. However, new vulnerabilities and ways to exploit them will always be discovered, and the people working with vulnerability management need to adapt to this. Therefore, understanding the challenges faced in the vulnerability management processes is paramount to learning from the challenges the interview subjects have experienced to improve the conceptual framework.

As new vulnerabilities will always be discovered, one of the challenges of vulnerability management is being able to account for every discovered vulnerability. As with all other aspects of operation

in an organization, a finite amount of resources are available for the vulnerability management effort. This means that the people working on vulnerability management must justify the allocated resources and work within this budget. Therefore, the usage of automatic tools and calibrations of processes is essential in vulnerability management:

> *"Mainly, it [the challenge with vulnerability management] is about being able to handle everything we find. I am trying to build something that reports vulnerabilities more automatically, and one thing I have discovered is that is a lack of a good place to find reliable data. [...] so we could work with this data and lookup every time we find a vulnerability and then report it to the correct team."* – Sec_Engineer_2

Another aspect of vulnerability management that many organizations fail to account for is its close relationship to their risk management effort. Including risk management helps justify using resources in vulnerability management as it provides context on how potentially dangerous vulnerabilities that are not accounted for can harm the organization. Additionally, motivating resource usage to ensure a continuous reassessment of vulnerability management efforts to higher management in the organization is easier to encourage if there are data to present that backs up the need for the resources. *Lead_Security* identifies this as a major challenge that organizations have in their vulnerability management efforts:

> *"Organizations generally do not ensure their suitable linking of this [vulnerability management] with risk management because this is what it ultimately is. It puts figures behind the risk involved and helps businesses ensure they put the correct focus there. [...] It needs to be done with a risk management view and understand that vulnerability management is not just patching. It is not just risk; it is mitigation. So you need to ensure that your vulnerability management program is following and feeding through this correctly with regular reviews."* – Lead_Security

Furthermore, the vulnerability management challenge exists when different asset owners are within the organization. When there is a divergence between the employees responsible for vulnerability management and the system owner of the affected asset, it might cause a discrepancy regarding who is responsible for the vulnerability being mitigated. This means there is the possibility for a vulnerability to either be stuck between responsibilities and subsequently take longer to mitigate or be overlooked, as both the vulnerability management team and the asset owner believe the other team is responsible for mitigation. When talking about the challenge of the ownership of assets in vulnerability management, *Sec_Officer* substantiated the following:

> *"If we start with ownership, for example, some of the vulnerabilities, which are really important to mitigate, have ended up in a finger-pointing situation. [...] This requires more time and follow-up to actually mitigate the vulnerabilities and understand the risks. There is always a big difference in how much time I have to use when following up on vulnerabilities. My task is mainly to notify the asset owner and follow up, not actually mitigate."* – Sec_Officer

Alternatively, *Infra_Advisor*, which works for Company C in IT consulting, identifies a challenge as keeping track of all of their solutions' vulnerabilities. Working in consulting and creating solutions for customers means that, at some point, there is a transfer of responsibility where the customer might acquire ownership of the solution and, by proxy, also the job of mitigating vulnerabilities. *Infra_Advisor* also mentioned the challenge of identifying vulnerabilities and communicating to the customer, but the vulnerability is disregarded:

> *"Really, the full picture of our solutions can sometimes be missing. We have created a lot of solutions through the years, some of which we may not have ownership of anymore, but we know that they are still functional and running at the customer, and we report vulnerabilities. [...] It [vulnerabilities] is not taken seriously, and nothing happens, which we think is quite unfortunate."* – Infra_Advisor

4.2.7 Framework-specific Input

The interviews contain questions regarding the processes and experiences of the interviewees, along with their understanding and input on the conceptual model to improve it in conjunction with the feedback. To prevent the interviewees from having to assess the model on short notice, all the interviewees received the model prior to the interview to individually assess the concepts and processes within the model. Consequently, the interviews were constructed so that there was allocated time between the interviewer and interviewee to discuss the framework and identify areas of improvement and applicability. This process is necessary to ensure the framework can be improved and to gain valuable insights from people who practice this methodology frequently. As such, the interviews collected numerous points of interest and several points of improvement.

In the interview with *Lead_Security*, there were several great contributions on aspects that should be added. The interviewee substantiated that risk management is an over-arching area that influences the behavior and actions of the security team. Consequently, ensuring risk management is included is important as it is ingrained in the business:

> *"I think it is quite reasonable [the conceptual framework]. What really is missing is at your tactical level, there should be governance, and that is where you bring in your risk. Ultimately, what you have in your tactical layer is governance monitoring; it is that oversight. Your operational is the "Do". Security can really struggle to provide anything other than what eventually gets perceived as scaremongering. You need the risk side of the business, coming in as a partner on that tactical level. They are ensuring not only that the security team is doing what it should be doing but also that they are monitoring the actual financial risk factors behind it. They help bridge the gap with the business, and especially with the C-level."*
> – Lead_Security

Additionally, when addressing the explicit roles presented in the framework, *Lead_Security* states that the operational level should include *application owner* to ensure there is an employee who is

responsible for the risk of a certain application:

> *"Obviously, you can go to very fine grains role level, but something that I would put in here is: your operational level works for the large part, but you could put application owner in here. Many businesses have their own systems, so what we generally see is that you actually move the ownership to ensure vulnerabilities are patched or mitigated, the risk acceptance, it has an owner who is taking responsibility. He has actually put on that system or application owner. It doesn't mean they necessarily do it, it may still be system administered, but it is about who owns the risk."* – Lead_Security

As *Lead_Application* works closely with implementing patches and ensuring timely deployment, they state that there should be a distinct management of *software life cycle*, which influences patch management as a whole, and should be run as an individual process:

> *"Here [in patch management] it also says "manage software life cycle", which I would extract to its own box there. Meaning you have "patch management" and "software life cycle" as they are two different processes. [...] The only thing I see now is that I would have "software life cycle management" as its own box, its own process."* – Lead_Application

As the conceptual framework depicts, the policies are created at the strategic level. However, as *Sec_Officer* states, the policies should not necessarily be created at the top level (strategic) but should be approved at the top level. When asked if the policies should be created at the strategic level, they answered:

> *"They should at least be approved at the top; that is the most vital. There can't be any leaders lower in the system that are approving the policies; then, it is worth nothing. Policies are very important, and they should be placed at the strategic level with input from the ones actually enforcing them."* – Sec_Officer

Additionally, *Lead_Security* also substantiated that the top-level management does not necessarily define the policies; however, they approve them:

> *"I wouldn't say that they define the actual policy; they approve it. It is the tactical level that defines those policies."* – Lead_Security

Moreover, as *Lead_IT* explains, having penetration testing as an information source in the model significantly contributes to ensuring all systems' areas are tested for vulnerabilities and subsequently applied patches. They also think that in order for the people working with patching to get financial backing from the top-level management, it is easier to point to a finding in a penetration test than an arbitrary report:

> *"I think that your penetration testing segment is extremely vital. You can monitor all you want, but penetration testing is what really brings it forward. At least for us, when there has been a pentest, we always find something. It is then easier to earn "backing"*

> *from the top of the pyramid to do something than to point to something in a report."* – Lead_IT

Lead_IT also substantiates the need for a *one-pager* which distills the vital elements of the conceptual framework to make it more appealing and readable by top-level management:

> *"I think that you can have a "one-pager" because then you will not lose people's interest on the first slide. All the aspects should be present, but you should simplify the contents of each process."* – Lead_IT

Infra_Engineer_2, which works closely with patching and its deployment, substantiates the need for testing before deploying patching. Therefore, having a dedicated testing environment ensures that the implemented patches do not unnecessarily break the system or cause other harm, which could be prevented with proper testing. When asked if the model should include a testing category, they answered:

> *"Yes. Because what you could do was you could choose an off week, and you could test. You could do some proper tests to see if you can break things. Even for system recovery, you could do normal things like: "I am going to delete everything out of Active Directory. Let us see how easy it is to recover it."* – Infra_Engineer_2

Additionally, *Infra_Engineer_2* substantiated the need for a communication channel with customers as a customer usually provides a lot of input and requests regarding patching. As they work in a company providing customer patching services, this would only apply to certain businesses. However, customer input could also correlate with management input. Thus, implementing a communication channel within the framework could contribute to better information flow between peers:

> *"They have a lot of contact with the customer. There's probably a lot of contact with customers here that I don't see factored in anywhere. The customer is very important, and they're really sitting hands-on with a lot of things when it comes to patching. So that's a big influence."* – Infra_Engineer_2

The following Table 4.1 summarizes the key points of improvement and feedback gathered from the interview subjects from the framework-specific questions, which is further used in developing the final framework model. Consequently, the feedback is carefully assessed to apply to a general audience of companies, given that there are different-sized companies with different needs. The feedback is therefore used in a general manner. The table is only assessing the feedback from the last section of the interview (Framework), where the additional findings made in Chapter 4 are added and discussed in Chapter 5:

Table 4.1: Framework-related input distilled from Section 4.2.7

Framework-related Input
Add risk management as an over-arching category
Put *"application owner"* in the operational level
Extract software life cycle as its own category
Policy creation from the strategic level should be exchanged with policy approval
Add *"patch testing"* as its own category
Add communication/contact with the customer, and their influence on the patching
Develop a distilled framework based on the conceptual model

5 | Discussion

The discussion presented in this chapter elaborates on the reviewed literature from Chapter 2 and the findings from Chapter 4, bringing together the theory and the acquired insight from subject matter experts. Furthermore, we discuss how the findings corroborate or deviate from the applied theory in the conceptual framework and use this to improve the comprehensive framework, ensuring that the theory and findings are represented equitably.

5.1 A Comprehensive Framework for Patching and Vulnerability Management in Enterprises

The comprehensive framework, as showcased in Figure 5.1, summarizes and conceptualizes the theoretical findings from the literature review (Chapter 2), data from the interviews (Section 4.2), and our collective understanding of how patching and vulnerability management is performed and effectivized in enterprises. The framework is empirically validated through extensive interviews with relevant peers to closely depict an operational framework businesses can use to perform and effectivize their patching and vulnerability management process. Moreover, the framework builds on essential literature that builds the model's foundation and ensures that important areas within the target topics are covered. Bautista (2018)'s model depicting the hierarchy and over-arching roles of leadership within an enterprise (Figure 2.5) is prevalent within the model to depict a clear difference in the levels of performing patching and vulnerability management. In relation to the interviews, Li et al. (2019)'s empirically validated list (Figure 2.4) is utilized as inspiration for which information sources are relevant for each level, while Li et al. (2019)'s and Dissanayake, Zahedi, et al. (2022)'s elucidation on the patch management process (Section 2.1.1) is ingrained. Lastly, the framework is built on and gathered inspiration from Huang et al. (2012)'s *Patch Management Workflow Process* framework (Figure 2.1). Comparatively, the conceptual framework (Figure 4.1) and the comprehensive and empirically validated framework (Figure 5.1) share several similarities in terms of setup and foundation. In addition, further vital parts and processes are implemented that influence and contribute to effective patching and vulnerability management based on the interviews, additional literature, and our sense-making.

A significant change implemented in the comprehensive framework is the removal of roles within each major level of leadership. Our findings indicate that the roles correlating to each category within

each level did not closely relate to the roles assigned in the interviewed companies. Therefore, as the majority of contributions and input revolved around the roles not applying to the interviewee's business model, whether it was based on resources, company size, or maturity level, the roles were removed. Subsequently, when a business wishes to utilize the model, the roles should be applied individually based on the organization's given posture and business hierarchy to fit proportionally. Consequently, through sense-making, it is up to each business to construct its roles and responsibility areas as applicable. The idea that each business will utilize the framework differently based on capabilities and structure is why the model may not include vastly technical components but rather generalized processes to adhere to all levels of enterprises.

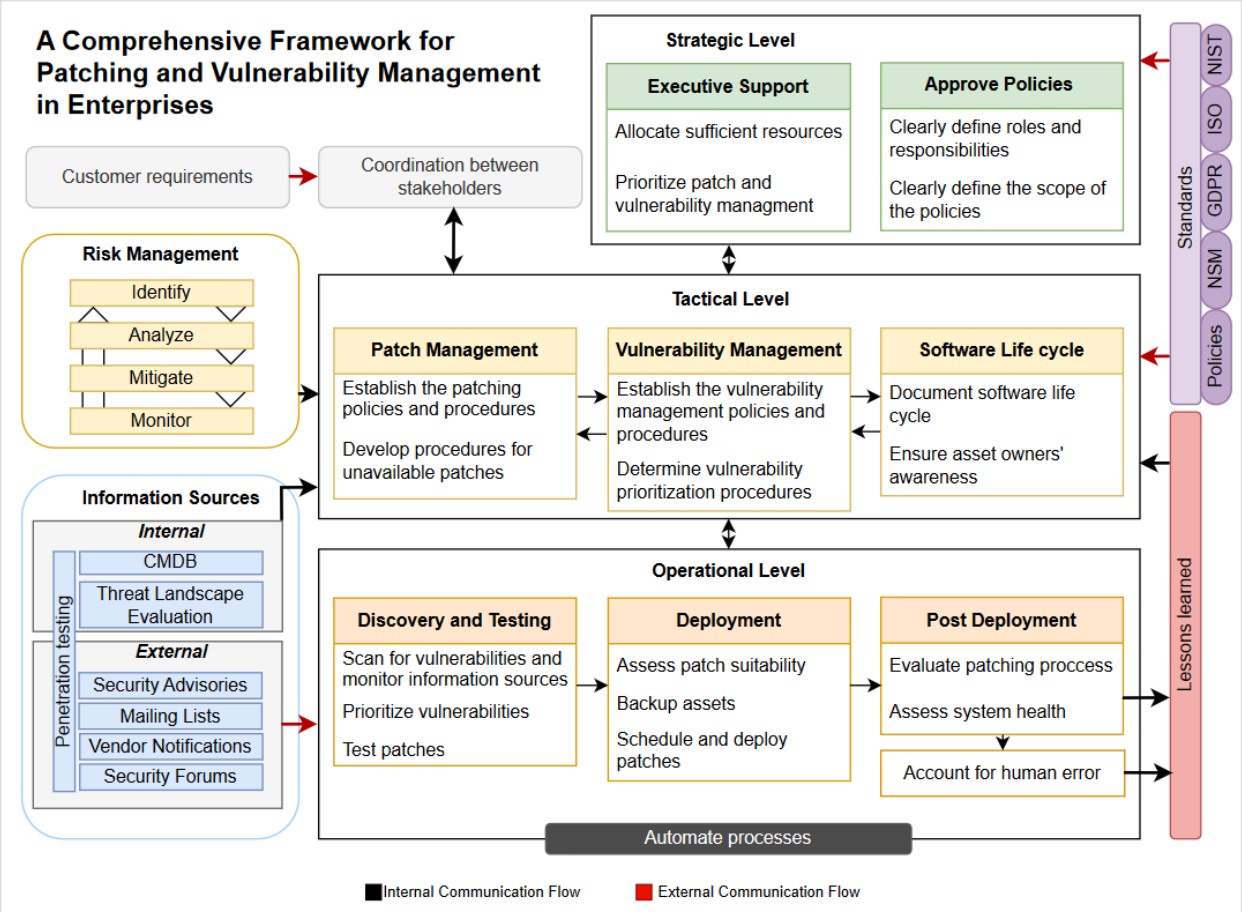

Figure 5.1: A Comprehensive Framework for Patching and Vulnerability Management in Enterprises

The framework also contains two different colors of arrows, which depict the type of information flow that is flowing throughout the enterprise when performing patching and vulnerability management. The black arrows signify *internal communication*, which is information gathered from internal sources and internal actors. This type of communication can be influenced by external information but is solely flowing internally. The red arrows signify *external communication*, which is the communication from outside of the business towards the business. In the model, this type of communication comes from the influence of standards on the strategic and tactical level, the external information sources

going to the operational level, and the external customer requirements influencing the coordination between stakeholders and the tactical level. Additionally, when comparing the conceptual framework and the comprehensive framework, the addition of internal communication between the internal information sources and the tactical level is prevalent. Through sense-making and input from the interviews, it was apparent that there should be a distinction between which levels of leadership receive the information from the information sources. Therefore, the internal sources point to the tactical level with a black arrow, while the external sources point to the operational level with a red arrow.

5.1.1 Strategic Level

The strategic level allocates sufficient resources for the patching and vulnerability management processes while ensuring the policies defined by the tactical level are appropriate and approved, as seen in Figure 5.2. A clear differentiation between the conceptual framework and the comprehensive framework is that there has been a change from *"Create Patch Policies"* to *"Approve Policies"*. Following the findings, extensive sense-making, and the gathered literature, strategic management does not necessarily *create* the policies, but they *approve* them mainly from the tactical level. Previously, the thought was solely that the strategic level created and approved the policies, but later it was updated to an approval process instead. As *Sec_Officer* elaborated on: *"They should at least be approved at the top, that is the most vital. Policies are fundamental and should be placed at the strategic level with input from the ones enforcing them."* The findings generally indicate a consensus that policies are approved at the top level but developed at the lower levels. Utilizing this as a guideline, the policies and procedures are developed within the tactical level due to them enforcing the actual policies and approved within the strategic level as they are responsible for the over-arching operations of the enterprise.

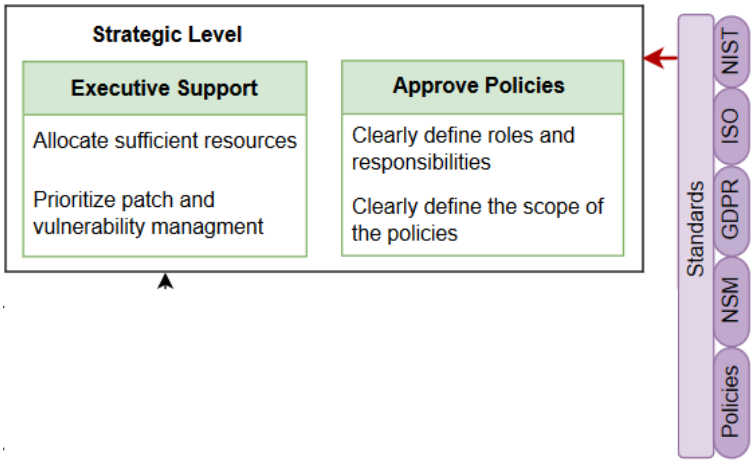

Figure 5.2: The Strategic Level of the Comprehensive Framework

Additionally, *"Financial Support"* has been substituted with *"Executive Support"* as the findings substantiated that support from the strategic level does not necessarily have to be in the form of

funds. The executive support aims at ensuring sufficient allocation of funds is performed to increase the value and capability of the patching and vulnerability management, as well as prioritize it as they are important areas for an enterprise to implement appropriately. As the findings suggest, support from the executive level is vital to allocate sufficient resources, as the majority of the interviewees agreed that there was currently insufficient allocation of resources to perform their desired work. The findings additionally point to the fact that the size of the company does not affect the insufficiency or sufficiency of allocation as, no matter the size, most interviewees agreed on the lack of allocation of resources.

The processes of the strategic level are selected regarding the findings, along with Souppaya and Scarfone (2022)'s recommendation of performing patch management planning in enterprises. Their research focuses on critical aspects to assess when implementing patch management, where creating and approving policies and procedures are vital. The findings ultimately confirmed the theory around the processes within the strategic level, which are primarily unaffected compared to the conceptual model (Figure 4.1).

Another addition to the model is the distinct types of standards, denoted at the side of the purple *Standards* box. The findings suggested there should be specific and relevant standards that are likely to influence the creation and approval of patching and vulnerability management, as well as influence the strategic level's prioritization on the topic areas. As the target companies are located in Norway, relevant standards and national advisories such as ISO, GDPR, and NSM are chosen as they apply to Norwegian and European businesses. However, NIST provides guidance applicable to American businesses (Souppaya & Scarfone, 2022), but the recommendations still apply to Norwegian enterprises. The findings also suggested that several interviewees followed the proposed standards and recommendations, including NIST. Additionally, as standards are seen on as *external communication*, the influence on the strategic level's processes is denoted with a red arrow.

5.1.2 Tactical Level

The tactical level of the framework represents the necessary planning that must be done before the practical patch and vulnerability management responsibilities, especially concerning the preemptive processes that must be satisfied to ensure that the operational effort is managed effectively and fulfills the laid-out strategic objectives. Compared to the conceptual model seen in Figure 4.1, the tactical level in the comprehensive framework seen in Figure 5.3 has undergone several adjustments. Mainly among the changes is the inclusion of *Risk Management*, which is further discussed later, and the inclusion of a *Customer requirements* and *Coordination* element. Incorporating a *Customer* element that realizes a customer's requirement on the framework directly relates to the *Coordination* element as an external communication factor. The addition of customer requirements as a factor in the framework is based on the findings, highlighting the importance of communication and cooperation with the customer. While customer feedback might apply to some organizations, specifically those offering patching and vulnerability management as a service, it certainly will not be relevant for all

organizations.

Nevertheless, its inclusion is justified because it is highly relevant for organizations working with customers. Contrarily, in an organization that does not have a customer in the same sense, it can be ignored as with its exclusion; the framework is still applicable. The inclusion of *Customer requirements* in the tactical level connected to *Coordination* is chosen as the assignment of handling customer relations falls under the category of a *"Stakeholder"* in the tactical work of a coordinator between the team working on the project and the customer.

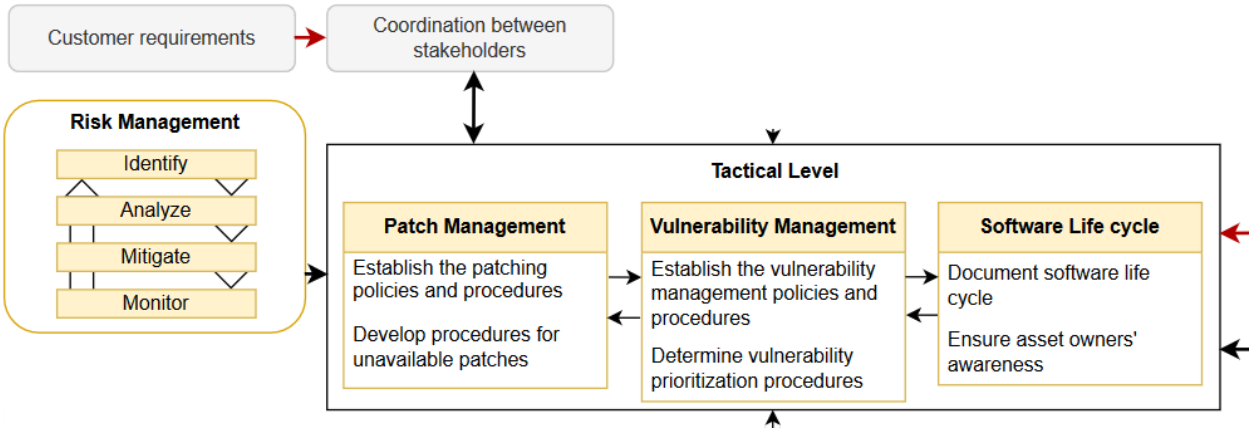

Figure 5.3: The Tactical Level of the Comprehensive Framework

Connected to the *Customer requirements* element in the comprehensive framework is the *Coordination between stakeholders*. This factor was added to the comprehensive framework both based on the communication mentioned above and cooperation with the customer identified by *Infra_Engineer_2* and through *Sec_Officer*'s insight on the importance of having someone being responsible for the coordination between the different parties involved with patching and vulnerability management. This *Coordination* aspect of the framework attempts to account for the required cooperation between potential customers and the patching and vulnerability management team and between different teams or departments in an organization. The framework presented by Huang et al. (2012), discussed in Section 2.1.2, displays the interconnectivity of different employees and teams in patch-related management and advocates through its complexity the need for a tactical effort in facilitating cooperation. Looking at the framework in Figure 2.1 with the different roles and communication flow, there is a definitive need for a coordination element. As discussed in Section 5.1, roles have been removed from the comprehensive framework, and therefore it will be up to the individual organizations to decide who should coordinate.

The findings substantiate making *Software Life Cycle* its own element on the tactical level of the framework instead of a component of *Patch Management*. This change sees the importance of *Software Life Cycle* being increased by being moved into its element and signifying that it affects patch and vulnerability management. Furthermore, while there are several different approaches and models for *Software Life Cycle* explicitly developed for software development, such as *Agile*

Model and *Waterfall Model* (Saravanan et al., 2020), which model is followed does not affect the framework's applicability. Therefore, the individual organization's approach to *Software Life Cycle* can be effortlessly applied to the comprehensive framework. Instead, the comprehensive framework points out the components of *Software Life Cycle* that directly affect the patching and vulnerability management effort. These are *Documentation* and *Awareness*, which directly relate to the findings' common themes regarding patching and vulnerability management. Specifically, the need for *Documentation* can be seen through *Infra_Engineer_2*'s response, where they advocate for the need for documentation as it is needed in the training processes for new employees. Additionally, *Awareness* is also required as it improves the control of assets, supported by *Sec_Officer*'s statement on asset control. Furthermore, it can help account for the problem with controlling assets identified by *Lead_Security*.

More minor changes were also made to the conceptual framework and adopted into the comprehensive framework based on the findings. Chief among these was the addition of *Develop procedures for unavailable patches*, which was added as it had relevance in patch management based on the findings, as there is decisively something that does happen and has to be accounted for. Given this finding, the element *Unavailable patches* was included in the framework. Another change made to the model was the inclusion of black arrows signifying the *internal communication* happening between the different elements at the tactical level, which was added to substantiate the importance of communication between the different aspects of patch and vulnerability management.

Risk Management

Perhaps, the most noteworthy change to the tactical level from the conceptual framework into the comprehensive framework was the inclusion of *Risk Management*. The main argument for its addition was *Lead_Security*'s description of how risk acceptance was necessary for any growth in an organization. Going further, the findings elaborated on the relationship between risk management and vulnerability management and how it can be used as motivation for getting the needed resources. In addition, the findings directly state that risk is the ultimate motive behind shutting down unpatchable systems. Overall, there was a consensus for the inclusion of *Risk Management*. Therefore, based on the consensus from the findings, *Risk Management* was included in the comprehensive framework. Given the constant discovery of new risks, we also decided to structure *Risk Management* cyclically. The findings substantiate that the work in risk management always starts with identifying a new risk and ends with monitoring and discovering a new risk. This ensures that the importance of continuous risk management is established.

Risk Management is added to the framework through the five most common steps, which are *Risk Management. Identify*, *Analyze*, *Mitigate* and *Monitor*. The *Identify* and *Analyze* steps deal with the discovery and classification of the risks and, therefore, the control of what should be done with the risk. The *Mitigate* step accounts for the actual actions taken to lessen or remove the risk, while *Monitor* is a continuation that takes place if the risk is not removed.

The inclusion of *Risk Management* in the framework ensures that an integral part of the security effort

of an organization is included. This is especially important as risk is also a very intrinsic element of both patch and vulnerability management, given the overall risk-based approach organizations take to security. Furthermore, its inclusion also ensures that organizations can easily see the relationship between *Risk Management*, patching, and vulnerability management. Consequently, organizations must understand that patching and vulnerability management is not a separate issue that can be handled in isolation but must be jointly considered as a part of the overall security effort.

5.1.3 Operational Level

The operational level accounts for the activities revolving around implementing patches, gathering information from various information sources, and executing operational work in relation to patching and vulnerability management. Compared to the conceptual model (Figure 4.1), the comprehensive framework allows for more communication and evaluation as *"Lessons learned"* has been added, as gathered as input from the findings. This process, as seen in Figure 5.4, allows for continuous improvement and learning as, at the end of each patch cycle, the operational level gathers valuable experience and knowledge collected from the given cycle. In turn, the lessons learned pose as a learning opportunity to improve the next patching cycle. Ultimately, having a process around learning outcomes may contribute to the practitioners gaining more knowledge after post-deployment ends to ensure inevitable mistakes are learned from. The addition of *"Account for human error"* is empirically validated through the interview process, as well as it is substantiated in Dietrich et al. (2018) (as discussed in Section 2.3.2). Accounting for human error, such as misconfigurations, ensures employees do not fear repercussions if such an incident arises. As Dietrich et al. (2018) elaborates, allowing for *blameless postmortems* is vital for employees and the organization to minimize the restraint of information.

Additionally, the need to add patch testing as a split category in *"Discovery"* was prevalent both in the empirical interviews and the literature review. Testing is substantiated through its prevalence in Li et al. (2019)'s and Dissanayake, Zahedi, et al. (2022)'s five stages of patch management (Section 2.1.1), as well as the findings supported that having testing in a business environment is essential as it ensures the implemented patches does not cause unnecessary harm to the target system. Ensuring a proper testing environment decreases downtime and less unwanted behavior on the system. However, contrary to the collected theory, which endorses testing patches thoroughly, the findings substantiated that testing only provides value if a proper test environment is prevalent. If the testing environment is too lightweight or does not merely inhabit the same system as the production system, testing will be of no value as the two systems are unaligned. Testing patches should thus preferably be performed with a suitable test environment, ideally a wholly mirrored system, to ensure the testing affects the live production implementation.

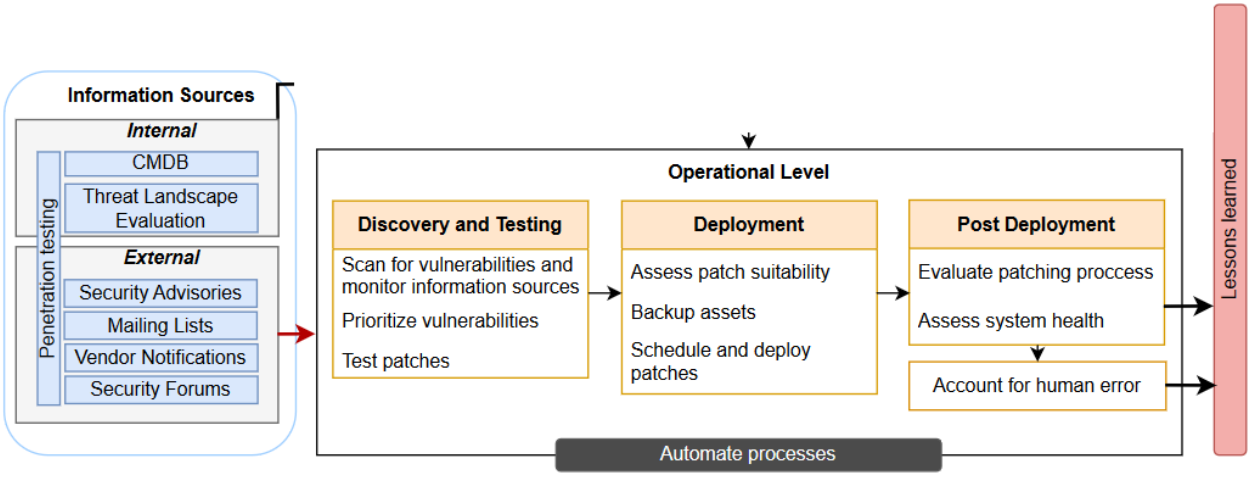

Figure 5.4: The Operational Level of the Comprehensive Framework

Moreover, in the findings and empirical collection from the literature, it is apparent that patching and vulnerability management automation should be enforced as much as possible, as it is a valuable tool to automate manual tasks. This, in turn, ensures time is saved on rigorous tasks, which can be further spent on tasks that require more human attention. Thus, the black *"Automate processes"* box is added to influence the entire operational level, including automation concerning internal and external vulnerability scans, patch testing, backup, patch scheduling and deployment, and post-deployment. Furthermore, automating tasks is reported in Dissanayake, Jayatilaka, et al. (2022), where interviewed practitioners substantiate that automation should be facilitated as much as possible in the patching process. Similarly, the empirical data from the interviews showed that several participants utilized automated tools to perform manual tasks to save time in their operations, such as Microsoft's patching tool SCCM (System Center Configuration Manager) and HCL BigFix.

Collectively, the operational level aims to perform the patching and vulnerability management task following best practices from the collected literature and the empirical validation and input provided by the practitioners within the field. The information sources proposed by Li et al. (2019) (Figure 2.4) are additionally confirmed through the empirical interviews as the majority of the interviewees utilized various information sources for their data collection. This relates to both the internal and external information sources and the usage of penetration testing to gather information. The findings suggest a vast usage of security advisories as well as online forums (such as the */sysadmin* subreddit) where several participants gathered their information. The external information sources are denoted with a red arrow as they are responsible for the external communication flow toward the operational level. Contrary, the internal information sources are denoted with a black arrow towards the tactical level, indicating the information comes from internal sources. Additionally, the *Discovery and Testing*, *Deployment*, and *Post Deployment* are all categories that are validated throughout the findings and the collected theory. Li et al. (2019) and Dissanayake, Zahedi, et al. (2022) substantiate the five phases of patch management in Section 2.1.1, where the three aforementioned categories

play a vital role.

Moreover, the interview findings also suggest a strong coherence and substantiation that these categories fit appropriately within the patching and vulnerability management process. Throughout the interviews, most interviewees recognized the areas as they currently implement them in their daily work, making them relevant to the framework. The processes within the three main categories are gathered both from the findings and from the literature search; however, the framework does not state any specifics on *how* to perform the patching and vulnerability management, only the most important process to include. A contributable factor is that businesses want to implement processes differently, whereas a method of work might fit one business more than another. It is, therefore, up to each enterprise how they want to implement each category. However, a trend identified was that smaller businesses tended to have more unclear responsibilities and did not perform all the proposed processes. Conversely, most of the interviewees working in larger businesses had distinct responsibilities, mostly accounting for the essential elements of the framework.

The black arrows within the operational level signify internal communication and a chronological approach to operational patching and vulnerability management. The arrows go from left to right as this is not an iterative process but a chronological process where each process is performed before moving on to the following process, as confirmed by the findings. There are additionally two arrows that go from *Post Deployment*; one to *Account for human error* and one to *Lessons learned*. If the current patch cycle did not include human errors, the communication goes directly to the lessons learned, but if there was an incident or an error to learn from, the information flow should be going through the *Account for human error* process to ensure it is accounted for in the next cycle.

5.2 Distilled Model

Contrary to the complete framework (Figure 5.1), the distilled model showcased in Figure 5.5 aims to depict the overarching processes of patching and vulnerability management. From the findings, there was raised concern that a comprehensive model may be too technical at first, so there should be a simplified one-pager. The distilled model consists of the same categories as the comprehensive framework, but it is heavily simplified to act as an intermediary model before a business implements the comprehensive framework. Additionally, a distilled model where most technical details are absent may be more applicable to C-level management as processes, and procedures may be more important to assess.

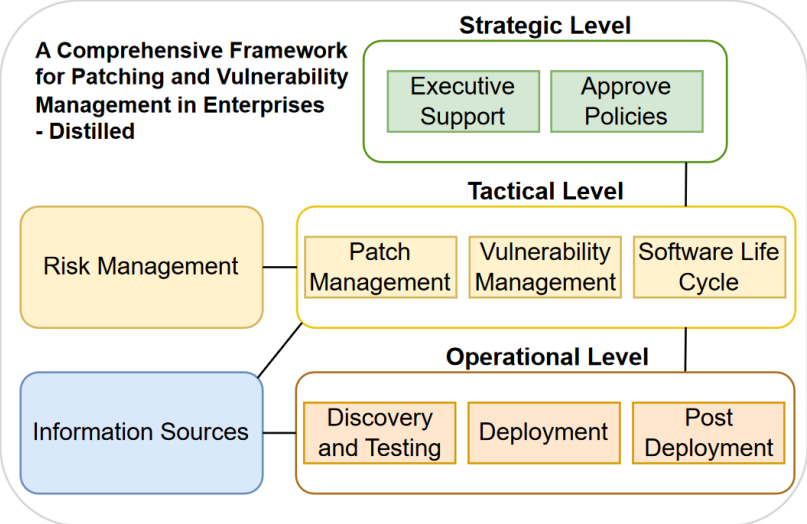

Figure 5.5: Distilled Model

Compared to Figure 5.1, the distilled model does not have specific tasks within each category, but it keeps the levels of leadership to adhere to a high-level depiction. As the model is only a process model, distinct processes such as standards, lessons learned, automation, and influence from customers and stakeholders are detached. This contributes to a transparent, over-arching model which only depicts the most vital processes for a business to assess.

5.3 Contributions

The main contribution of our work is *"A Comprehensive Framework for Patching and Vulnerability Management in Enterprises"* seen in Figure 5.1, which summarizes the different elements of patching and vulnerability management, showing their interconnectivity. This framework applies to organizations and can be utilized to guide the development of patching and vulnerability management elements or identify elements already present within an organization. The elements within this framework are based on the description of the subject area in the literature, as well as valuable data from the findings. For example, in the patching element of the framework, Li et al. (2019)'s and Dissanayake, Zahedi, et al. (2022)'s elucidations are essential for the different tactical and operational components. Correspondingly, the different steps of vulnerability management described by Bautista (2018) were also instrumental in developing the vulnerability management components of the framework.

Moreover, Bautista (2018)'s layered strategic, tactical, and operational approach also was used to distribute different themes, thus defining the different responsibilities within patching and vulnerability management and their different related levels of leadership. The framework was extended through the interview processes, where the opinions of relevant interview subjects helped extend the framework through their experiences working with patching and vulnerability management. The comprehensive framework, therefore, represents the combination of the literature authors'

comprehension and the interview subjects' understanding and experiences. This contributes to a framework developed around fundamental themes from the literature with relevant input from people working with patching and vulnerability management. Moreover, complementary to the comprehensive framework is the distilled model, seen in Figure 5.5, which depicts the overarching processes of the comprehensive model. The distilled model, which serves as a simplified version of the comprehensive framework, is applicable in an organization as an explanatory to management, who might not have the needed knowledge or time to review the comprehensive model. Therefore, its function is as a motivation mechanism that simplifies the overarching themes of the framework.

5.4 Limitations and Future Work

Several limitations affect the results gathered in this research. Firstly, perhaps the most significant limitation of the research is the given period allocated for the entire master thesis. As the master thesis is scheduled to be performed over a university semester, a finite amount of time can be used for every part. Research into the patching and vulnerability management practices of organizations could potentially, depending upon willing interview subjects, amount to considerably more interviews than those conducted in this research. This limitation is somewhat limited through the scope discussed in Section 3.2.1, but having more interviews would give even more weight to any identified characteristics.

Another limitation of the research is the sensitivity of the subjects discussed with the interview subjects. The sensitive manner of the subject area for the interviewed organizations means that extra caution has to be taken to ensure that no potentially harmful information is made public. With this as an overlaying danger, the interviews themselves have the potential to err on the side of caution, in the sense that details that might have been valuable for the research are left out as a precautionary measure. Therefore, it is essential that both the *"right"* questions are asked and that the interview subject is made to feel comfortable. The *"right"* questions here mean avoiding questions that lead to potentially sensitive information.

Moreover, there is also a risk of the researcher's bias and difference of opinion when analyzing the interviews. Morse (2015) defines two types of researcher bias in qualitative inquiry: to find what is predicted and to interpret data with predefined values. In other words, the first type of bias is through the creation of findings before they are found, while the latter type of bias is more so regarding the researchers' values and how they affect the findings. In essence, the primary way to avoid these biases is through taking a neutral attitude toward the research by avoiding overly expecting or looking for correlation but still being open to it (Morse, 2015), which is an approach that has been substantiated by the researchers in the thesis. Correspondingly, the two researchers' subconscious values and viewpoints are also different. When researching qualitative data, there are bound to be differences in opinions when multiple researchers are involved. Notably, how the researchers interpret the data, given the lack of objective truths in qualitative research, is especially prevalent for the coding of interview responses collected, as discussed in Section 3.4, considering the

many different ways an interview response could be inferred. To oppose researchers' bias, Morse (2015, p. 9) recommends strictly that one adheres to the research principles, which were utilized in this research.

One of the main limitations of the contribution from the work done within this research is the lack of testing of the framework. The only indications of the framework's applicability and effectiveness are from the discussion with the interview subjects. In this process, the interview subjects were given access to the framework ahead of the interview, but none applied the model to their organizations. Therefore, the measurement of applicability and effectiveness is only gathered from the feedback given by the interview subjects and not from actual real-scenario testing of the framework. This is definitively a limitation of the developed framework, and testing in organizations is an obvious point of future research. Future research could include some benchmarks where an organization's patch and vulnerability management effort is investigated both before and after the implementation of the framework. The criteria of these investigations could be the employee experience of patching and vulnerability management before and after applying the framework, displaying the potential improvement or decline in the effectiveness of the patching and vulnerability management effort.

6 | Conclusion

The main objectives of the research were to identify how enterprises perform their patching and vulnerability management locally and how interviews and discussions with relevant interview subjects could contribute to a collectively enhanced patching and vulnerability management process. Consequently, to address the problem statement and research questions and to address the gap in existing literature, we have developed a comprehensive framework that aims at depicting the process of how patching and vulnerability management are integrated with an enterprise and how feedback from interviewees can aid in ensuring best practices are followed for a more effectivized and enhanced process. The comprehensive framework is further developed from the conceptual model as interviewees' feedback is added, in addition to supplementary research. The framework aims to depict the communication flow between leadership levels (strategic, tactical, operational) and stakeholders relevant to an enterprise and how these are interrelated in a patching and vulnerability management process.

Through an iterative process reflecting the findings from an extensive literature search, our sense-making, and validation from interviews, the extended model aims to clearly showcase how enterprises can utilize such a framework to integrate patching and vulnerability management into their enterprise infrastructure. The three main levels of leadership ultimately lay the foundation of the processes from the policy creation to the actual implementation of a patch.

Research Question 1: The semi-structured interviews' findings suggest that organizations perform patching and vulnerability management rather differently based on factors regarding existing knowledge and available resources. Interviewees working in larger companies tended to have more resources available when performing patching and more automated processes, ultimately saving time and resources. Consequently, as more resources were available, the larger companies had separate departments which performed patching and vulnerability management, where typically the patching department (operational) received threat information and patch information from the vulnerability management department (tactical). Additionally, patch information regarding severity and prioritization was pre-configured within the integrated patch systems – making the process more streamlined. However, it requires more planning and coordination compared to if a small team is responsible. Contrarily, the smaller enterprises had allocated fewer resources resulting in fewer employees being assigned the patching and vulnerability management role. As a result, gathering threat information and performing patch-related work was generally performed in one department.

Patching and vulnerability management were generally divided into three main levels of leadership: strategic, tactical, and operational. These levels each inhabit crucial processes validated in the interviews, which build up the processes the interviewees utilize currently, in addition to the theory. The strategic level approves patching and vulnerability management policies and provides financial and strategic support for the topic areas. The tactical level aims to create policies and procedures while assessing strategic and operational input. Furthermore, the tactical level assesses the internal configuration setup (CMDB) and the general threat landscape to develop relevant policies and procedures. Lastly, the operational level is responsible for performing system scanning, testing relevant patches, deployment of the patches, and handling post-deployment issues that may arise. Generally, none of the interview subjects followed any specific framework for how patching and vulnerability management is supposed to be performed in the enterprise, nor was there any literature that depicted the relation between patch management and vulnerability management as a unity. Consequently, the framework is developed as an addition to the literature and patching and vulnerability management for enterprises to integrate and improve the processes that are currently present.

Research Question 2: The insight from the interview subjects in relation to the relevant theory around patching and vulnerability management contributed towards enhancing the comprehensive framework. As most interview subjects worked closely with the topic areas, the insight gave us relevant and current information that could be translated into enhancements within the framework. As a result, several suggestions and key feedback points were added within the comprehensive framework and the theory that substantiated and validated the data. Subsequently, we integrated the insight and the additional theory within the extended framework following the feedback from the conceptual framework. As the theory proposed several suggestions regarding current implementations and practices, the semi-systematic literature review aided us in gathering relevant articles while still assessing input from the interviewees.

The SSLR, combined with the inclusion and exclusion criteria, ensured high-quality reports which provided insights and knowledge applicable to the framework and discussion. Utilizing a semi-structured qualitative approach also aided us in continuously developing follow-up questions for the interviewees to gain further insights into the practices and thoughts, thus gaining additional information to enhance the framework. Following a qualitative method, contrary to a quantitative method, discussion-based questions are more prevalent, which is crucial in our research for extracting key points from the interviews. Consequently, the in-depth gathering of theory allowed us to critically evaluate and confirm the empirical data from the interviews and identify similarities and differences.

Conclusively, the thesis utilizes a carefully selected SSLR in relation to semi-structured qualitative interviews to understand how enterprises perform patching and vulnerability management and gain insights to improve the processes and answer the research questions appropriately. Enterprises perform these areas differently depending on the allocation of resources and other contributing factors; however, the collection of theory and empirical data resulted in a comprehensive framework that enterprises hopefully will utilize to improve their patching and vulnerability management

processes. The comprehensive framework was built in relation to the conceptual framework, with our understanding and the interviewees' experiences playing a significant role. As enterprises have not yet implemented the framework, its effectiveness and substantiation of validation are not yet apparent; however, the comprehensive framework builds on important experiences and practices identified in relevant and high-quality literature and from skilled professionals in their field. As such, its integration, even without current validation, could contribute to enhanced patching and vulnerability management processes within an enterprise.

Appendices

A Interview Guide 79
B Information letter and Consent Form 81

A Interview Guide

Problem Statement: *How do organizations implement patching and vulnerability management?*

RQ 1: *How are organizations facilitating patching and vulnerability management?*

RQ 2: *How can the insight from appropriate subjects enhance the patching and vulnerability management process?*

Themes:

- Development of patching and vulnerability management in organizations and at customers
- Interview subject experience with the theme internally and externally
- Security routines and general patching, and vulnerability management

Interview Techniques:

- Introduce ourselves informally to start a dialogue and make the conversation comfortable for the interview subject
- Mention the anonymization process and how they can remove unwanted information
- Allow for silence to achieve in-depth answers
- Informed consent

Del 1: Bakgrunnsinformasjon

1. Hvor jobber du og hva jobber du med?
2. Hvor lenge har du jobbet innen ditt felt og hva slags felt innenfor dette har du jobbet med/interesserer deg for?
    a. I forhold til patching og sårbarhetshåndtering, hva slags oppgaver utøver du om dette?
3. Er patching og sårbarhetshåndtering noe dere utdøver selv eller er det noen som bistår dere ekstern?
4. Er patching og sårbarhetshåndtering en prioritert del av IT-sikkerheten i deres bedrift?
    a. Er det nok ressurser tilgjengelig til å tilstrekkelig håndtere dette?

Del 2: Rutiner og Prosesser

1. Har dere noen interne dokumenter/rutiner/prosesser som beskriver hvordan patching skal utføres I bedriften?

    a. Er det et behov for bedriften å ha et rammeverk som beskriver rutinene for patching?
2. Føler du at de med tilstrekkelig kunnskap lager rutinene eller er de utformet av mindre kvalifisert personell? Hvem utarbeider rutinene og hvem godkjenner de?
    a. For eksempel, er det ledelsen som har utarbeidet rutinene eller er det de som faktisk jobber med det?
3. Bruker dere noen offentlige/lukkede kilder for å innhente og dele relevant trusselinformasjon og informasjon om patching?
4. Hvordan prioriterer dere hvilke sårbarheter som skal prioriteres?
    a. Bruker dere noe rammeverk for å klassifisere sårbarheter og prioritere de mest kritiske?
    b. Hvorfor brukere dere dette rammeverket? Og er det modifisert på noen måte eller bruker dere en standard løsning?
    c. Syntes du den nåværende løsningen er tilstrekkelig eller er det noe du syntes kan forbedres?
5. Litteratur viser at ansatte er redde for å være åpne når det gjelder feilkonfigurasjon. Har dere hatt noen hendelser hvor feilkonfigurasjon av tjenester/systemer har ført til en sikkerhetshendelse? Hvordan ble det håndtert?
    a. Hvis ikke, hvordan ville dere håndtert dette? Har dere noen spesielle rutiner på det?

Del 3: Utfordringer ved patching og sårbarhetshåndtering

1. Er det noen utfordringer ved patching du har erfart?
    a. Hvordan vil du se for deg løsningene til disse problemer kan være?
2. Er det noen utfordringer ved sårbarhetshåndtering du har erfart?
    a. Hvordan vil du se for deg løsningene til disse problemer kan være?
3. Hvordan har dere/ville dere håndtert forsinkelser i utrulling av patcher?
4. Hvordan håndterer dere sårbarheter som ikke kan mitigeres?
    a. Forklar

Del 4: Rammeverk

Forklarer rammeverket vi har utarbeidet.

1. Gir det presenterte rammeverket ett godt bilde på hvordan patching og sårbarhetshåndtering utøves i bedriften din?
    a. Eventuelt hva ville du har endret på?
    b. Trenger modellen at det vises hvilke ansatte som utfører de forskjellige rollene eller det forstårlig uten det?
        i. Eventuelt, er det riktig type ansatt på riktig plass?
    c. Er det noe som ikke er lett forstårlig i modellen?
    d. Trenger modellen noe som sier hva firmaer skal prioritere?
    e. Skal risk management inkluderes i modellen?

B Information letter and Consent Form

Vil du delta i forskningsprosjektet

Masteroppgave: Patching og sårbarhetshåndtering?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å samle informasjon og erfaringer angående rutiner for patching og sårbarhetshåndtering. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Formålet med prosjektet er å lage et rammeverk på patching og sårbarhetshåndtering som bedrifter kan bruke for å forbedre sin tilnærming og sine rutiner angående disse temaene. I forbindelse med dette vil vi gjerne utføre intervjuer med relevante ansatte i bedriften for å få en grundig forståelse av hvordan dette utføres og hvordan rutinene rundt er oppbygd. Dette vil også bli brukt i sammenheng med samlet litteratur.

Motivasjonen bak dette prosjektet er å samle hvordan patching og sårbarhetshåndtering burde håndteres basert på teori, og binde det sammen med hvordan det faktisk håndteres ved intervjuer med relevante bedrifter som dere. Målet med dette er å etablere en mer informert forståelse av patching og sårbarhetshåndtering for å kunne lage et relevant rammeverk som forhåpentligvis dere og andre bedrifter er interesserte i og som kan styrke denne delen av sikkerhet.

Problemstillingen for prosjektet er: *Hvordan har patching og sårbarhetshåndtering blitt foreslått og implementert internt i bedrifter?*

Prosjektet blir gjennomført i forbindelse med masteroppgave i Cybersikkerhetsledelse ved Universitetet i Agder.

Hvem er ansvarlig for forskningsprosjektet?

Universitet i Agder er ansvarlig for prosjektet.

Hvorfor får du spørsmål om å delta?

Vi vil gjerne snakke med bedrifter som har relevant kompetanse og vi har identifisert dere som en god kandidat. Utvalget er trukket basert på deres rolle med sikkerhetsmiljøet nasjonalt.

Kriteriene vi har som grunnlag er at bedriften må ha et sterkt miljø innen IT-sikkerhet og ha erfaringer innen patching og sårbarhetsanalyse slik at det kan brukes i utvikling av et rammeverk.

Andre bedrifter som vi har vurdert som relevante etter disse kriteriene har også blitt spurt om å delta.

Hva innebærer det for deg å delta?

Hvis du velger å delta i prosjektet, innebærer det at du deltar i et intervju, enten eller fysisk eller

digitalt. Det vil ta deg ca. 30-60 minutter. Vi vil gjerne ha informasjon om hvor lenge du har vært i stillingen din og relevant erfaring. Intervjuet vil inneholde spørsmål om temaene:

- Utvikling av patching og sårbarhetshåndtering i bedriften (og hos kunder dersom dette er relevant)
- Erfaringer med patching og sårbarhetshåndtering internt og eksternt
- Sikkerhetsrutiner og generell håndtering

Intervjuet vil bli tatt opp og blir registrert elektronisk hos Universitetet i Agder, og anonymisert.

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. Etter intervjuet er transkribert vil du motta et dokument hvor du har mulighet til å endre, redigere, og fjerne informasjon du ser relevant.

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.

I tillegg til studentene som utfører prosjektet (Gustav Martin Kvilhaug Magnussen og Mathias Pettersen), vil også prosjektveileder (Marko Ilmari Niemimaa) ha tilgang for å kunne bistå. Personopplysningene som lagres vil lagres på en kryptert skyserver tilhørende Universitet i Agder, i tillegg til at det lagres separat på et digitalt hvelv som krever ekstra autentisering for å nås. Navnet og kontaktopplysningene dine erstattes med en kode som lagres på egen navneliste adskilt fra øvrige data.

All publisert data vil bli anonymisert, slik som personopplysninger, stilling, og sensitiv informasjon om bedriften. Anonymisering utføres ved koding av informasjon. Deltakere, samt bedriften, vil bli gitt en anonym tittel som brukes for klassifisering av data.

Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?

Prosjektet vil etter planen avsluttes 3. Juni 2023. Etter prosjektslutt vil datamaterialet med dine personopplysninger slettes.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Universitetet i Agder har Sikt – Kunnskapssektorens tjenesteleverandørs 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:

- Marko Ilmari Niemimaa (marko.niemimaa@uia.no, +47 38 14 18 42) ved Universitetet i Agder
- Vårt personvernombud: Trond Hauso, personvernombud@uia.no

Hvis du har spørsmål knyttet til vurderingen av prosjektet som er gjort av Sikts personverntjenester ta kontakt på:

- Epost: personverntjenester@sikt.no, eller telefon: 53 21 15 00.

Med vennlig hilsen

Marko Ilmari Niemimaa (Forsker/veileder)	Gustav Martin Kvilhaug Magnussen & Mathias Pettersen (Studenter)

Samtykkeerklæring

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

- ☐ å delta i intervju
- ☐ at anonymisert data blir publisert i prosjektoppgave

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

---

(Signert av prosjektdeltaker, dato)

Bibliography

Ahmad, A., Maynard, S. B., & Park, S. (2014). Information security strategies: Towards an organizational multi-strategy perspective. *Journal of Intelligent Manufacturing*, *25*(2), 357–370. https://doi.org/10.1007/s10845-012-0683-0

Alsaawi, A. (2014). A critical review of qualitative interviews. *European Journal of Business and Social Sciences*, *3*(4), 149–156. https://doi.org/10.1093/applin/amq043

Avison, D., Lau, F., Myers, M., & Nielsen, P. A. (1999). Action research. *Communications of the ACM*, *42*(1), 94–97. https://dl.acm.org/doi/fullHtml/10.1145/291469.291479

Baiardi, F., & Tonelli, F. (2021). Twin Based Continuous Patching To Minimize Cyber Risk. *European Journal for Security Research*, *6*, 1–17. https://doi.org/10.1007/s41125-022-00079-7

Bautista, W. J. (2018). *Practical Cyber Intelligence: How action-based intelligence can be an effective response to incidents* (1st ed.). Packt Publishing.

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). Investigating System Operators' Perspective on Security Misconfigurations. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 1272–1289. https://doi.org/10.1145/3243734.3243794

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

Dissanayake, N., Jayatilaka, A., Zahedi, M., & Babar, M. A. (2023). An Empirical Study of Automation in Software Security Patch Management. *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. https://doi.org/10.1145/3551349.3556969

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. *Proc. ACM Hum.-Comput. Interact.*, *6*(CSCW2). https://doi.org/10.1145/3555087

ENISA. (2021). ENISA Threat Landscape 2021. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021

Farris, K. A., Shah, A., Cybenko, G., Ganesan, R., & Jajodia, S. (2018). VULCON: A system for vulnerability prioritization, mitigation, and management. *ACM Transactions on Privacy and Security*, *21*(4). https://doi.org/10.1145/3196884

FIRST.org. (2021). Common Vulnerability Scoring System version 3.1: Specification Document. *2021*(22.12.2021), 1–24. https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf

Fujs, D., Mihelič, A., & Vrhovec, S. L. (2019). The power of interpretation: Qualitative methods in cybersecurity research. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3339252.3341479

Grime, M. M., & Wright, G. (2016). Delphi Method. In *Wiley statsref: Statistics reference online* (pp. 1–6). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118445112.stat07879

Hammersley, M. (2013). *What is Qualitative Research?* Bloomsbruy Academic. https://doi.org/10.5040/9781849666084

Hollstein, B. (2011). Qualitative Approaches. In J. Scott & P. J. Carrington (Eds.), *The sage handbook of social network analysis* (1st ed., pp. 404–416). SAGE Publications Ltd. https://citeseerx.ist.psu.edu/document?repid=rep1%7B%5C&%7Dtype=pdf%7B%5C&%7Ddoi=7e6b15cc612e630e25a8d8a647d6f46eb1c904a8

Hore, S., Shah, A., & Bastian, N. D. (2023). Deep VULMAN: A deep reinforcement learning-enabled cyber vulnerability management framework. *Expert Systems with Applications*, *221*(November 2022). https://doi.org/10.1016/j.eswa.2023.119734

Howland, H. (2023). CVSS: Ubiquitous and Broken. *Digital Threats: Research and Practice*, *4*(1), 1–12. https://doi.org/10.1145/3491263

Huang, H., Baset, S., Tang, C., Gupta, A., Sudhan, K. N. M., Feroze, F., Garg, R., & Ravichandran, S. (2012). Patch management automation for enterprise cloud. *2012 IEEE Network Operations and Management Symposium*, 691–705. https://doi.org/10.1109/NOMS.2012.6211988

Jacobs, J., Romanosky, S., Edwards, B., Adjerid, I., & Roytman, M. (2021). Exploit Prediction Scoring System (EPSS). *Digital Threats: Research and Practice*, *2*(3). https://doi.org/10.1145/3436242

Jenkins, A., Kalligeros, P., Vaniea, K., & Wolters, M. K. (2020). "Anyone Else Seeing this Error?": Community, System Administrators, and Patch Information. *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, 105–119. https://doi.org/10.1109/EuroSP48549.2020.00015

Kallio, H., Pietilä, A. M., Johnson, M., & Kangasniemi, M. (2016). Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing*, *72*(12), 2954–2965. https://doi.org/10.1111/jan.13031

Kamolson, S. (2007). Fundamentals of quantitative research Suphat Sukamolson, 20. https://www.brlaboratory.org/wp-content/uploads/2022/11/SuphatSukamolson.pdf

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *2*.

Kvale, S. (2011). *Doing Interviews*. SAGE Publications, Ltd.

Langley, A., & Meziani, N. (2020). Making Interviews Meaningful. *Journal of Applied Behavioral Science*, *56*(3), 370–391. https://doi.org/10.1177/0021886320937818

Li, F., Rogers, L., Mathur, A., Malkin, N., & Chetty, M. (2019). Keepers of the Machines: Examining How System Administrators Manage Software Updates. *Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security*, 273–288.

Mell, P., Dugal, D., Casotto, F., Nordwall, P., & Sommerfeld, D. (2022). Measuring the Common Vulnerability Scoring System Base Score Equation. https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8409.pdf

Morse, J. M. (2015). Critical Analysis of Strategies for Determining Rigor in Qualitative Inquiry. *Qualitative Health Research*, *25*(9), 1212–1222. https://doi.org/10.1177/1049732315588501

Nappa, A., Johnson, R., Bilge, L., Caballero, J., & Dumitraş, T. (2015). The attack of the clones: A study of the impact of shared code on vulnerability patching. *Proceedings - IEEE Symposium on Security and Privacy*, *2015-July*, 692–708. https://doi.org/10.1109/SP.2015.48

NSM. (2020). NSMs Grunnprinsipper for IKT-sikkerhet. https://nsm.no/getfile.php/133735-1592917067/NSM/Filer/Dokumenter/Veiledere/nsms-grunnprinsipper-for-ikt-sikkerhet-v2.0.pdf

NSM. (2022). Nasjonalt digitalt risikobilde 2022. https://nsm.no/getfile.php/1312007-1667980738/NSM/Filer/Dokumenter/Rapporter/NDIG2022%7B%5C_%7Donline.pdf

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, *372*. https://doi.org/10.1136/bmj.n71

PCI. (2022). Payment Card Industry Data Security Standard Verions Requirements and Testing procedures Version 4.0. https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Standard/PCI-DSS-v4%7B%5C_%7D0.pdf

Qu, S. Q., & Dumay, J. (2011). The qualitative research interview. *Qualitative Research in Accounting and Management*, *8*(3), 238–264. https://doi.org/10.1108/11766091111162070

Rabiee, F. (2004). Focus-group interview and data analysis. *Proceedings of the Nutrition Society*, *63*(4), 655–660. https://doi.org/10.1079/pns2004399

Saravanan, T., Jha, S., Sabharwal, G., & Narayan, S. (2020). Comparative Analysis of Software Life Cycle Models. *Proceedings - IEEE 2020 2nd International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2020*, 906–909. https://doi.org/10.1109/ICACCCN51052.2020.9362931

Schulze, M. (2020). Cyber in War: Assessing the Strategic, Tactical, and Operational Utility of Military Cyber Operations. *International Conference on Cyber Conflict, CYCON*, *2020-May*, 183–197. https://doi.org/10.23919/CyCon49761.2020.9131733

Serio, L., & Gentile, U. (2019). Survey on international standards and best practices for patch management of complex industrial control systems: the critical infrastructure of particle

accelerators case study. *International Journal of Critical Computer-Based Systems*, *9*, 115. https://doi.org/10.1504/IJCCBS.2019.10020044

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, *104*, 333–339. https://doi.org/10.1016/j.jbusres.2019.07.039

Souppaya, M., & Scarfone, K. (2022). *Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology* (tech. rep. Special Publication (SP) 800-40 Rev. 4). U.S. Department of Commerce. Washington, D.C. https://doi.org/10.6028/NIST.SP.800-40r4

Suri, H. (2011). Purposeful sampling in qualitative research synthesis. *Qualitative Research Journal*, *11*(2), 63–75. https://doi.org/10.3316/QRJ1102063

Tiefenau, C., Häring, M., Krombholz, K., & Von Zezschwitz, E. (2020). Security, Availability, and Multiple Information Sources: Exploring Update Behavior of System Administrators. *Proceedings of the Sixteenth USENIX Conference on Usable Privacy and Security*.

Wang, B., Li, X., de Aguiar, L. P., Menasche, D. S., & Shafiq, Z. (2017). Characterizing and Modeling Patching Practices of Industrial Control Systems. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, *1*(1). https://doi.org/10.1145/3084455

Watson, R. T., & Webster, J. (2002). Analysing the past to prepare for the future: Writing a Literature Review. *MIS Quarterly*, *26*(2), 2005–2008. http://www.springerlink.com/index/R777101802276537.pdf

Xiao, Y., & Watson, M. (2019). Guidance on Conducting a Systematic Literature Review. *Journal of Planning Education and Research*, *39*(1), 93–112. https://doi.org/10.1177/0739456X17723971

Xu, C., Chen, B., Lu, C., Huang, K., Peng, X., & Liu, Y. (2022). Tracking Patches for Open Source Software Vulnerabilities. *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 860–871. https://doi.org/10.1145/3540250.3549125