# Patch management planning - towards one-to-one policy

Veikko Markkanen and Tapio Frantti*
University of Jyväskylä, Jyväskylä, Finland
veikko.m.markkanen@student.jyu.fi, tapio.k.frantti@jyu.fi
*corresponding author

*Abstract*— Software security is a key component in protecting an enterprise's digital and operational infrastructure. Time from vulnerability disclosure to exploitation is decreasing and a large number of vulnerabilities are being exploited before security teams have implemented patches or other mitigation methods. To achieve a sufficient level of software security, timely patching is critical. However, failure to recognize patching as a necessary business cost is extremely common at many levels of organizations. The lack of proper patch management strategy is apparent. In order to help organizations to improve their patch management planning we surveyed the latest academic publications, vulnerability reports and the latest NIST special publication 800-40 on patch management planning and focused on enriching the provided recommendations for organizations, to improve their management of threats.

*Keywords- patch management; vulnerability management; patching policy*

## 1. INTRODUCTION

At any given time, information and operational systems will have vulnerabilities, especially if they use legacy systems or software. This is often the case with industrial actors and smart buildings with shared spaces that include number of different technologies and assets that have been integrated to each other. These vulnerabilities pose significant risk to enterprises and have been connected to, for example, over 60% of all data breaches according to a study by Ponemon Institute [1]. According to another study conducted by the Ponemon Institute, only 53% of the surveyed 660 IT security professionals responsible for managing their organization's endpoint security strategy said that their organizations have a formal patch management process in place [2].

One of the main reasons for vulnerabilities is that vendors release buggy products first and let the markets test them only to fix bugs later. This is possible because vendors face little liabilities for losses caused by security holes in their products [3].

Several successful attacks could have been prevented or limited if the correct security patches provided by the software producer would have been applied [4]. In many cases, organizations cannot effectively identify the vulnerabilities that are an actual threat if left unpatched [5]. They also state that only small portion of known vulnerabilities are exploited, so the optimal goal is to try and find the vulnerabilities that are most likely to be exploited or the ones that can cause the most damage. However, Condon et al. states that 56 % of the vulnerabilities were exploited within seven days of public disclosure [6]. They examined 50 vulnerabilities that set considerable risk to organizations of all sizes. In total, the report included 45 vulnerabilities that were exploited in 2022, of which 44 % arose from zero-day exploits. They also show that 87 % increase in first-week exploitation since 2020 [6].

In the past information systems were protected by numerous layers of physical and network security controls. While patching was generally recognized as a necessary step to improve the resilience of information systems, there were few willing to prioritize. In the operational technology the situation has been even worse. At the practical level, the code and systems were often considered ready at the time of purchase. Such neglect of proper patch management has since cost the global economy hundreds of billions (US billions, $10^9$ $) in repair, data, and credibility, among other organizational assets. Patching today has reached the level of mission criticality [7].

In today's environment it is widely recognized that the past software security measures have largely been rendered ineffective. This is due to most technologies' direct exposure to the Internet and as such, the past perimeter of non-connectivity no longer exists. It is a fact that the systems today are at significantly greater risk of compromise. This has created the need of zero-trust perimeter, that assumes that no user, device, or network should be trusted by default, even if they are inside the organization's perimeter.

Even so, enterprise patch management is lacking, deficient or it obeys an obsolete policy. The culture of productivity and usability over-security still holds strong in today's world. On many occasions the interest of security and business departments clash when it comes to security investments. For someone focused on creating continuous profit, causing any disruption for the production, or ramping up costs for anything but certain needs, may seem unnecessary [8]. Even if the long-term goal of business resilience is included in strategic planning, many are willing to tolerate the risk of non-patched systems. While at time this was also a valid strategy, in most cases it is not any more. The leadership should reconsider the priority of patch management in light of today's risks [7]. On the contrary to popular belief, it is more than likely that unpatched systems, especially popular ones will be targeted and exploited. Therefore, patching should be considered a standard cost of doing business. If an organization needs a certain technology to conduct operations, it needs to maintain that technology throughout its life cycle – and that includes evaluation of risks, vulnerabilities, and impacts of the realised risks, security measures and effective patching policy.

2767-6684/23/$31.00 ©2023 IEEE
DOI 10.1109/DSA59317.2023.00018

Enterprise patch management planning and policy are critical aspects of reducing the significant risk of unpatched software vulnerabilities. Leadership at all levels of organizations and supply chains should combine their efforts to create a strategy that simplifies and operationalizes patching while also improving its reduction of risk. Doing so will increase organizations resilience in today's active threats and minimize the impacts for business and improve business contingency.

## 2. LITERATURE REVIEW

Dissanayake et al. formulate, using [9], [10], [11], and [12], that software security patch management refers to the process of applying patches to the security vulnerabilities present in the software products and systems deployed in an organizations IT environment. The process consists of identifying existing vulnerabilities in managed software systems, acquiring, testing, installing, and verifying software security patches [13]. Dissanayake et al. also note that a major part of patch management studies focus on preventive actions such as vulnerability scanning, risk assessment and prioritization whereas only a small portion of studies consider on the actual patches and the developing, testing and deploying them [13].

Baiardi and Tonelli [14] define patches as countermeasures that remove vulnerabilities by deploying new code to replace flawed one. They continue that patches are among the cheapest countermeasures, but no effective solution exists to select the patches to deploy and to schedule their deployment and the average time to patch a vulnerability is steadily increasing and it currently ranges from 60 to 150 days but with a very long tail.

Authors in [13] state that there is no commonly accepted definition of software security patch management. Given that, several academic studies have formed their own operational definitions for patch management, based on their understanding and studies regarding the subject. Therefore, the general model for patch management program is often described like '*patch management is the process for identifying, acquiring, and verifying patches for products and systems*'. However, it does not consider patch assessment as a whole, *i.e.*, vulnerability identification, analysis, evaluation, and treatment process. We emphasize that the *patch management is about bringing the entire system up to an acceptable state*. This requires understanding and identification exactly your asset. The lack of visibility or lack of awareness of vulnerable spots leave weakly managed assets completely vulnerable to attacks. One vulnerable connected device in a system is enough to make the entire system vulnerable.

In another paper Dissanayke et al. state that many cybersecurity attacks with devastating consequences can be traced back to a delay in applying a security patch.

They continue that despite the criticality of timely patch application, not much is known about why and how delays occur when applying security patches in practice. The authors present an illustrative overview of the causes of delays in software security patch management [15].

Authors in [16] note that automated patching tool does not take into consideration exploitation probability of the vulnerabilities. The definition of the exploitation probabilities for vulnerabilities is a difficult problem and in many case it is more an artistic than scientific operation.

August et al. point out that a patch deployment often involves a significant setup cost, such as costs associated with system configuration checking, patch searching and documentation, and patch testing and installation [17], [18]. Furthermore, unplanned patching activities are bound to cause some business disruption, interrupting the normal system workflow and inflicting downtimes on critical business functions [19]. Beres and Griffin keep that as the primary reason why organizations postpone applying available patches.

Dey et al. [4] argue that organizations must consider both the setup and business disruption costs and weigh them against the potential exploitation cost, and decide when and how often to patch an enterprise system/application or its subsystems/components.

Dey et al. also analyze and compare different patching policies [4]. **One-for-one policy** refers to the practice of patching immediately after a patch becomes publicly available. NIST SP [9] notes that ideally organization would deploy every new patch immediately to minimize the time that systems are vulnerable". However, it also keeps such a policy as impractical and notes that it makes more sense for organizations to "balance their security needs with their needs for usability and availability" [4], [9].

In **time-based policy** an organization performs its patching operation at a predetermined time interval [4]. Beres and Griffin [19] note that the policy reduces the amount of device downtime due to patching as administrators are able to batch more patches together. Dey et al. [4] write that another major advantage of such a policy is that "all activities that need to happen prior to patch deployment can be preplanned and necessary resources can be allocated accordingly".

In **patch-based policy** an option is to patch when a predetermined number of patches become available [4]. Patches arrive at random intervals and waiting a batch of patches may expose systems to risks.

In **total-control policy** patching is done at moments the cumulative security risk reaches a predetermined threshold. The cumulative risk is the sum of the severity levels of all patches that have arrived but not been deployed [4]. This necessitates that organizations should identify and assign risk rankings to all vulnerabilities. National Vulnerability Database (NVD) and Common Vulnerability Scoring System (CVSS) can be used to get severity evaluations of the reported vulnerabilities.

**Emergency-control policy** refers to the practice of patching upon the arrival of a patch with a severity level that exceeds a predetermined threshold [4].

The policies represented from [4] are operationalized using a single metric, such as the patching interval, number of patches, or patch severity level. These policies can also be used together as a hybrid policy that combines two or more metrics. Dey et al. analyze and compare various policy classes with an aim to make a specific recommendation about what policy works the best in any given situation [4].

Patch management is commonly recognized as a critical need for organizations to mitigate the significant threat of exploitable, patchable software flaws. However, it is also a cause of internal contest within organisations. There are several challenges that complicate patch management efforts. As described by Dissanayake et al. [13] and Souppaya and Scarfone [9], some of the most impactful challenges include the lack of resources and time, the gap in knowledge of technical and business context, lack of support for dynamic environment context, the possible side effects, and the complexity of the patching process. While these issues have proven long standing, many solutions exist to help organizations overcome the challenge. One of the most well-known solutions to address these issues is patch management planning.

Patch management planning is a somewhat contentious term. It is widely recognized as a vital part of the vulnerability management life cycle, but the emphasis varies. According to Souppaya and Scarfone [7] and Tom et al. [20] patch management planning is the context of all software vulnerability management and serves the crucial role of minimizing the time and resources spend, while still achieving the required level of software integrity within an organization. These approaches are clearly motivated by the need of a mediator, between the business and security departments within an organization. According to this view, patch management should be considered an organization wide issue that can and needs to be solved in coordination with most of the leadership. In many other views patch management planning, despite its critical role, is not defined as an individual part of the patch management life cycle nor a specific focus of the provided patch management best practises [21], [22]. The reason for this is that these publications approach patch management from the security perspective. The patch management process is aligned with the security operations centre (SOC), and therefore it is assumed that the context and strategy are clear and accounted for within the budgets on the organizational level.

Shakir et al. presents the main security issues associated with cloud computing in their survey paper [24]. They classified papers into three categories: Security Issues in Cloud Computing, Authentication models in Cloud Computing, and Security Framework in Cloud Computing. The paper did not recognize patch management at all. However, organizations are now using more than before Software as a Service (SaaS) that may simplify usage and update of software but it may also enforce effects of new software vulnerabilities to the assets due to information exchange over the Internet. Identification of the assets that organization actually uses and which software versions are safe to use with the assets is then more complicated to share with service providers, SOCs, and organization itself. Kaplan (2008) notes that some businesses have reportedly been concerned about giving control and responsibilities to the SaaS providers [23]. They continue that the SaaS is based on the same source code and managing patches in larger quantities is a lot easier when compared to local software. It also means that if the SaaS provider is the target of an attack or their own systems or software contains vulnerabilities, the possible consequences are more significant and often harder to estimate.

In summary there are at least two major approaches towards patch management planning and patch management in general. The patch management planning as a mission critical need for the whole organization, and the more technical approach of patch management as a responsibility of the security team. These will be called the *general* and *technical* approaches to patch management.

Chronologically, the more security team aligned approaches have been dominating the scene of patch management for quite a while. Since the early 2000's there have been numerous publications that acknowledge the lack of knowledge and resources to establish or even consider applying some sort of software vulnerability management programs [25], [26], [20]. Since then, the information has spread and the academic approach has shifted towards patch management technologies, best practises and how to apply them [27], [13], [21], [9], [20]. It can be concluded that at this point the security management had the necessary knowledge available to establish a proper vulnerability management program. Still, the cyberattacks resulting from unpatched software vulnerabilities have grown in frequency and destructive potential [28]. The concern is reflected by the latest change in the academic literature surrounding the issue. The enterprises have what they need to create a proper patch management system at their respective levels. What they may not have, is the collective understanding of the necessary investments, in the scope of information security. The lack of concern for the modern-day threats has started to birth publications specifically focusing on the more general, organization wide approach to cyber security, which includes software risk and vulnerability identification as a responsibility of all levels within an organization [7].

Some research has diverged from the general line of development towards specific issues, such as the persistent existence of known exploited vulnerabilities [29].

## 3. NIST SP 800-40R4

The NIST SP 800-40 fourth revision is a state-of-the-art advocate for the general, more comprehensive approach to patch management planning. Developed by the National Institute of Standards and Technology (NIST), under the authority of U.S Department of Commerce. The purpose of this publication is to help improve enterprise patch management planning for organisations to strengthen their management of assets and vulnerabilities and planning the risks responses. The core value of the publication is the comprehensive understanding of patch management as a preventative maintenance for an organization's technology.

NIST SP 800-40r4 is written under the assumption that within the overall scope of enterprise patch management, organisations would benefit more from rethinking their patch management planning than their patch management technologies. The assumption is based on the availability of resources focusing on software vulnerability management for enterprises. It is expected that security management has the theoretical capacity of establishing proper patching regime, at their respective level, in light of years of warnings and best

practises published by numerous credible sources such as NIST themselves. The incentive for this publication is therefore in realizing this potential. The main challenge, for such approach is overcoming the differences between the business and security requirements. The publication is written in light of the changes within the threat landscape. There are more vulnerabilities, more threat actors, and the cost of time and resources is fast approaching uncontrollable. The landscape of the cyber security threats is evolving towards the point that threats that were once considered unlikely are occurring with regularity. This ongoing trend can be attributed to higher maturity of attack tools and methods, increased exposure, and increased motivation of attackers. This will also force us Basic software vulnerability life cycle to become better at protecting our assets and devising creative solutions to mitigate risks and threats.

Patching has become increasingly important, but simultaneously, the leadership is still reluctant to invest in this activity, risking the software integrity and therefore the operative capacity of their enterprises. NIST SP 800-40r4 tries to level the differences regarding patch management between the leadership and security management, by providing the information necessary to allow organizations to simplify and effectively produce a patch management plan.

The SP 800-40 defines three ways of how following the guidance presented within the publication should help organizations:

- Security and technology management and leadership at all levels of the organization gain a new understanding of the role of patching in enterprise risk management.
- The security and business personnel of organizations will be able to communicate with each other more effectively regarding patch management and reach consensus on planning.
- Security and business personnel of the organization will be prepared to revamp their enterprise patching strategy throughout the entire patch management life cycle.

### 3.1. Risk response approaches for software vulnerabilities - context

Once assets and vulnerabilities have been identified, the next phase is planning the risks responses and prioritize them. The framework of NIST SP 800-40r4 is the risk response model. It centers around assessing the risk and impact of each vulnerability identified, selecting a risk response, and determining how to best remediate the risk. Vulnerabilities with high impact need to be prioritized and repair whenever the patch is available. Less critical vulnerabilities may wait for a while, but it is recommended that patches should be deployed as quickly as possible. This requires risk and vulnerability management process with periodic update of the risks and related impacts.

The risk response model consists of four types of risk responses; *Accept, Mitigate, Transfer* and *Avoid*. The main purpose of the approach is to address the fact that there is more to software vulnerability management than patching. According to the publication the risk responses present the available choices for each individual situation.

Building on the risk responses, 800-40r4 also describes the basic software vulnerability management life cycle, applicable to all risk response approaches. Summary of the said life cycle is presented in Figure 1.

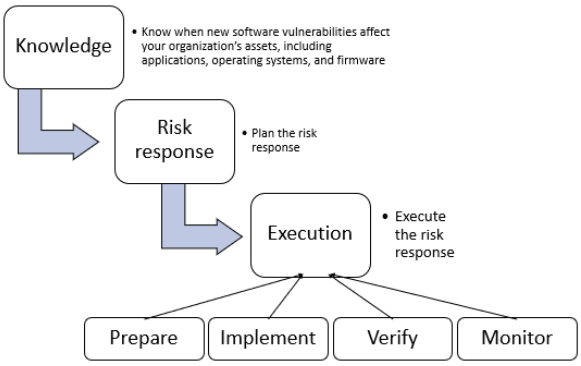


Figure 1. Basic software vulnerability life cycle

In the context of patching NIST SP 800-40 places emphasis on the risk response execution. It addresses the crucial step of preparation, implementation, validation, and monitoring for patch deployment. Preparation encompasses preparatory activities, which include prioritizing, scheduling, acquiring, and testing the patch, followed by deploying the patch. The deployment phase varies from the environment, but it commonly includes implementation, validation, installation, configuration, and resolving issues. Implementation examples include distributing and installing the patch, deploying additional security controls, and changing asset configurations and state. Validation involves ensuring that the implementation has been completed successfully and vulnerable assets were decommissioned or replaced.

Finally, as described by the special publication, the deployment is verified and entered into the state of constant monitoring.

### 3.2. Vulnerability management planning – best practises

The second half of the special publication focuses on patch management best practises. It offers both realistic situational awareness as well as a set of concrete steps for organizations to take to enhance and encourage patch management.

The publication recognizes patch management as a contentious issue, with different personnel having conflicting opinions. For example, the case of balancing trade-offs between earlier deployment and more testing. Deploying patches faster may reduce the opportunity of successful attack but increases the risk of operational disruption for the enterprise due to the lack of testing. Unfortunately, this in term increases the window of opportunity for the attacker. Testing can also drain resources, without a concrete payoff as not all vulnerabilities pose risk for the organization. According to the publication what has made enterprise patch management even more tough recently is how dynamic and disperse computing assets have become. The sheer amount of software to patch is oftentimes uncontrollable and disruptive.

Because of this change, trying to see profit in comparison to expended resources has become harder. Consequently, patching has often become reactive versus proactive and in combination with organizations not being able to keep up, the environment favours the attacker. According to SP 800-40r4 what needs to change is the perception of patching as a nuisance caused by the organization on itself. Disruptions from patching are controllable, which creates a clear difference on incidents caused by a third party. In addition, the disruption or "nuisance" is necessary for maintaining nearly all types of technology in order to avoid larger disruption from incidents. It is also evident that there are not any error free software and patching is necessary task for all of them. Therefore, the publication presents recommendations for organizations to implement, in order to improve their patch management planning, thereby minimizing the potential negatives of patching to operations.

To further concretize the patch management planning, the special publication first defines the main principles of security management:

- Problems are inevitable; be prepared for them
- Simplify the decision making
- Rely on automation
- Start improvements now.

Considering the defined principles of security management, the NIST SP 800-40r4 approach to patch management planning is as follows:

- Organizations should strive to reduce patching related disruptions. By which NIST aims to challenge the increased attack surface and lower the amount of patching needed for organizations. Possible methods are described as validating the software and service supply chain and decreasing the amount of software, especially those connected to the Internet.
- Organizations should establish and constantly maintain up-to-date software inventories for their physical and virtual computing assets. According to the special publication, outdated inventories cause increasingly inaccurate and incomplete information for patching efforts. The publication also emphasises the need for a robust system to include information on each computing asset's technical characteristics and mission/business characteristics. By doing so the organization is able to address the fact that each asset has those technical and mission/business characteristics, that provide context for the vulnerable software running on that asset. The publication also realizes that this is impossible manually, and strongly recommends automating the process. Vulnerability scanning of the systems is a widely used method to identify non-compliant, unpatched and vulnerable parts. It is a method largely used for continuous identification of the vulnerabilities. It should be included to the patch management plan as a periodic automatic process and regularly manually address only those computers where it is not working and correct the problem. Vulnerability scanning requires the use of multiple scanners for more complete vulnerability coverage.
- Organizations should also define the software vulnerability risk response scenarios necessary. The publication addresses the following examples: Routine patching, Emergency patching, Emergency mitigation, and an Unpatchable asset. Brief explanations are provided for each of the risk response scenarios.
- Organizations should assign each asset to a maintenance group. According to the publication, there should never be a case of an individual asset, with a management plan of its own. The publication presents a set of simplified examples of possible maintenance groups including mobile workforce laptops for standard end users, on premises datacentres and legacy OT assets. In addition, organizations should define a maintenance plan for each group, defining the protocol for when a risk scenario occurs.
- For the last two recommendations and the final messages for the document the NIST special publication presents the actionable enterprise-level patching metrics and the consideration for maintenance when procuring software. For metrics NIST specifies their importance in several roles of patch management and vulnerability management, also addressing the existence of many free and commercial sources for more information on the subject. According to NIST, most importantly, they enable organizations with vision into the effectiveness of their patch management programs. This information is crucial for the leadership to create the right decisions in response to existing risk.
- In the case of software maintenance procurement, NIST recommends that the organizations consider the software maintenance needs before procuring any piece of software. It also addresses the fact that providing methodologies for estimating maintenance costs of factoring software maintenance into procurement decisions is out-side of the scope of this publication. Even so, the document ends with a sample questionnaire of software maintenance needs that a new software may include.

While NIST emphasizes involving top management and rethinking patch management as an organizational asset, it lacks focus on the post-patching process. In contrast to the general principles, NIST does not provide actionable details on change management, backups, and testing, indicating these resource-intensive aspects to be discouraging for management. Table 1 presents a comparison of the NIST SP 800-40r4's recommendations on software security patch management planning with 13 commonly proposed best practices from academic literature.

Table 1. Comparison of the NIST SP 800-40r4's recommendations on software security patch management planning with 13 commonly proposed best practices from academic literature.

| | BP 1 | BP 2 | BP 3 | BP 4 | BP 5 | BP 6 | BP 7 | BP 8 | BP 9 | BP 10 | BP 11 | BP 12 | BP 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NIST SP 800-40r4 | x | x | x | x | x | x | x | x | | x | | | x |
| Limiting Vulnerability Exposure through effective Patch Management: threat mitigation through vulnerability remediation [43] | x | x | | | x | | | x | x | x | x | x | x |
| Software security patch management - A systematic literature review of challenges, approaches, tools and practices [13] | | x | x | x | x | x | x | x | | x | x | x | |
| A Practical Methodology for Implementing a Patch management Process [42] | x | x | | | x | x | | | x | x | x | x | x |
| Survey on international standards and best practices for patch management of complex industrial control systems: the critical infrastructure of particle accelerators case study [12] | x | x | x | | x | | | | x | | x | x | x |
| Patch Management [8] | | x | x | x | x | x | x | | x | x | x | x | x |

In Table 1 BP1 = Reduce Patching Related Disruptions; BP2 = Inventory Your Software and Assets; BP3 = Define and distribute roles and responsibilities; BP4 = Establish effective plans to communicate with stakeholders; BP5 = Define Risk Response Scenarios; BP6 = Assign Each Asset to a Maintenance Group; BP7 = Define Maintenance Plans for Each Maintenance Group; BP8 = Choose Actionable Enterprise-Level Patching Metrics; BP9 = Consider change introduced by patches; BP10 = Automate, distribute, utilize technologies for effective deployment and installation; BP11 = Develop a backout plan; BP12 = Improve testing framework and environment; BP13 = Require validation, reporting and documentation throughout the patch management lifecycle.

## 4. Software as a service

The NIST SP 800-40r4 focus on patch management planning considers implicitly locally, on premises, installed software, but the landscape of the software use is changing towards SaaS. SaaS apps are typically accessed by a web browser and it is a common delivery model for many business applications. In early 2020, Check Point researchers discovered and reported critical vulnerabilities in the Microsoft Azure infrastructure [30]. The vulnerabilities received the highest CVSS score of 10.0, critical. Unlike for traditional or local software, identification of the vulnerabilities of the SaaS apps and installing of the updates is SaaS provider's responsibility. However, organizations should identify SaaS connections to their assets and prepare to prevent misuse of them in case of delayed patching. Each asset has those technical and business characteristics, that provide context for the vulnerable software running on that asset, but the approach reduces patching and software maintenance costs of the single organization using SaaS apps.

For SaaS apps risk response belongs to the service provider, but organizations using SaaS apps should plan risk management process. The responsibility of the risk response initiation may still belong to the organization but response execution, *i.e.*, actual patching belongs to the SaaS provider. However, organizations using SaaS apps should be aware of the vulnerabilities affects to their assets related to the use of apps. If they receive, *e.g.*, online information from vulnerability databases and know vulnerabilities of an app, they should consider abandoning of the vulnerable app until it is patched. Deployment of the patches are simplified in case of SaaS apps; software providers need to update software in their servers and end users have the updated version in use.

## 5. Guidelines for patch planning

The NIST SP 800-40 collection on software vulnerability management aka. patching, intends to be a complete guide on the subject, updating regularly and shifting focus on issues that at the time, appear most pressing. The fourth revision of this publication has put specific focus on patch management planning. The focus is unique in a sense, that very few publications specifically target patch management planning at all levels of leadership within an organization. In fact, it appears the to be the only publication solely focusing on driving said subject. That being said, comparable research exists. Referencing back to the literature review, in many cases, patch or vulnerability management planning is brought up as a crucial step of any vulnerability management program.

### 5.1. Enriching the NIST SP 800-40r4

The SP 800-40 intends to be a thorough guidance on patch management planning, targeting specifically the void of knowledge between the business and the security management. It considers the context of patch management, the changing environment and along that a new risk landscape. For its main takeaway it presents a robust set of recommendations for organizations. However, it is because of this thoroughness that the reader is left wondering about priority. The publication itself notes the issue of lacking

resources as one of the main challenges for effective patch management. Still, at the very level of decisions, answering the question of what we should do, the publication does not offer clear guidance on how to prioritise the recommendations, unlike it does for the case of a single vulnerability. For example, Ruppert [10], in his publication "Patch Management", dedicated a separate chapter for instructions on where to start in creating a successful patch management program. In their publication Recommended Practise for Patch Management of Control Systems, Tom et al. [14] addressed the issue by referring the reader towards another document: "Quarterly report on cyber vulnerabilities of potential risk to control systems" by US Department of Homeland Security. Similar approach of referring sources that help build understanding of vulnerability profile and can provide basis for prioritizing patch management efforts, could prove beneficial for the NIST SP 800-40r4 if focusing on strategic priority is not intended.

As for its main takeaway, the NIST SP 800-40r4 recommendations, they generally go through, or at least recognize most of the recommendations and best practises present in other significant publications on patch management planning. A comparison can be made between the SP 400-80r4 and Software Security Patch Management – A Systematic Literature Review of Challenges, Approaches, Tools and Practises by Dissanyake et al. [13] in which most of the recommendations meet at some level. Both of these however, fail to emphasise an important aspect of patch management best practises and therefore an element to consider in patch management planning, the backups. The Special Publication expects the patching process to succeed if performed according to the recommendations. This however is not always the case. Despite appropriate testing, the asset owner should maintain a current and functional archive, that provide a last "good" snapshot of functional system, as written by Tom et al. [14]. The plan should describe at least the frequency of the backup, the process, and functional requirements of creating the archive, verification process, retention period, and the physical storage. The backups must also be validated before storing them to avoid, for example, storing contents encrypted by malware for ransomware. Proper back up planning is a widely accepted as a crucial security practise. Nothing would be lost by including it in the special publication. Similar conclusion can be drawn regarding testing as part of proper change management.

Another improvement to consider for the NIST SP 800-40r4, is the lack of reasoning behind each recommendation. For the security management, the recommendations may not need further explanation, but according to Ruppert [10] for the leadership in business and other departments, the "why" is crucial. The special publication gives context on to why patch management in general is important and provides the tools for leadership to create a realistic strategy for patch management, but the recommendations themselves are simply that. Based on historical context the approach works only when the security personnel already have the resources needed. Unfortunately, this is rarely the case. Improvements could include translating recommendations to numbers [10] and citing existing case studies of failed patch management. According to CISA advisory [2] there should be numerous examples of known vulnerabilities exploited in the wild which can further support the agenda.

It is possible to argue, that it is the security managements job to find reason behind each, individual recommendation, that is made into practise. However, even the security department has gaps in knowledge. Logically, and as done by various research [2, 4, 5, 15], providing explanations for each recommendation would improve the reception of the NIST SP 800-40r4 publication.

NIST maintains National Vulnerability Database (NVD) [31] of the scored vulnerabilities and specifies Common Vulnerability Scoring System (CVSS[1]) used for vulnerability scoring. In addition, Computer Emergency Response Team (CERT, Carnegie Mellon University) [32] publishes recent vulnerabilities also read regularly by hackers. This sets urgent timeline to patching process.

### 5.2. Patching policy for critical system

Rapid7's 2022 Vulnerability Intelligence Report states that today the time from vulnerability disclosure to exploitation is decreasing and a large number of vulnerabilities are being exploited before security teams have any time to implement patches or other mitigations [6]. 56% of the vulnerabilities in the report were exploited within seven days of public disclosure. In addition, 43% of the widespread threats Rapid7 researchers analysed in 2022 began with a zero-day exploit [6].

According to Edgescan [33], the average time taken to remediate internet-facing vulnerabilities was 57.5 days in 2022 whereas for critical severity vulnerabilities it was 65 days. The values can be considered as the reference values for the times that may follow the delayed patching. The US government's National Vulnerability Database (NVD) which is fed by the Common Vulnerabilities and Exposures (CVE) list currently has over 176 000 entries and more than 19 000 have a CVSS score of 9.0–10.0 meaning critical [33].

Immediate patching involves the risk of an unreliable patch economical expenses especially in plant environments, whereas late patching leaves a target system exposed to malicious attacks. When a new potential threat emerges, cyber security professionals make decision if the certain flaw is likely to be exploited and what is the impact of the exploitation. Exploitation probability is an evaluation at best and affected by the possible foreseeable reward that the malicious actors may get if the attack is successful. Possible impact of the exploitation includes direct financial costs, but also reputational damages. Both the exploitation probability and the impact are evaluations, not exact numbers. Shortened time windows between a disclosure of vulnerability and its' exploitation method should be considered in the exploitation probability evaluations.

[1] https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf

Maverick company challenges [34] the traditional security and risk management process. They recommend that security and risk management leaders should abandon risk management models in favour of threat exposure management and business impact analysis (BIA). By using BIAs, many, but not all, of the failings of risk assessments can be avoided. The impact of bias is lessened as business leaders have defined priorities for threat reduction [34]. From the patching point of view, the best solution is to install patches one-for-one policy based that makes ambiguous probability, impact, and priority estimations of the risks with patch obsolete.

For the secure use of the systems, whether information technology or operational technologies, timely patches are essential. We refer timely patches as one-for-one patching policy, *i.e.*, the practice of patching immediately after a patch becomes publicly available. As previously mentioned in Literature Review, NIST SP [9] notes that ideally organization would deploy every new patch immediately to minimize the time that systems are vulnerable", but it also keeps such a policy as impractical. They note that it makes more sense for organizations to "balance their security needs with their needs for usability and availability" [4], [9]. However, shortening time between disclosure or discovery of vulnerabilities and exploitation methods, guides us to deploy immediate patching despite of the inconveniences. Postponing patching may, in the worst case, lead to neglecting patching. Joint Cybersecurity Advisory report [35] provides details on the top 30 vulnerabilities routinely exploited by malicious cyber actors in 2020 and those being widely exploited thus far in 2021. Most of those vulnerabilities could have been mitigated by applying the available patches to their systems and implementing a centralized patch management system.

The SolarWinds hack in 2020, which involved the exploitation of a vulnerability in a third-party software component. SolarWinds unwittingly sent out software updates to its customers that included the hacked code that created a backdoor to customer's information technology systems. That highlighted the need for organizations to be vigilant in their management of third-party vulnerabilities [36]. Despite of the fact that the software patch included the malicious code, we still recommend deploying immediate patching but we also recommend detecting and testing patches carefully before it is sent out to customers. The current practice to let customers test half-made software is not acceptable and it is not understandable with security patches at all. The proper testing and validation of patches was pointed out in by Marinescu and Cadar as a critical aspect of patch management [37]. They emphasized the importance of testing and validation of patches before deployment to minimize risks and ensure successful deployment without negatively impacting system functionality.

## 6. Discussion

The basic concept of risk has stayed the same in regard to patch management. There are vulnerabilities in software, that have to be fixed in order to achieve a sufficient level of software security and resilience. Patch management is about bringing the entire system up to an acceptable state. Even so, much of the discussion has revolved around the fact that few of these vulnerabilities would ever get exploited by an adversary [2]. Therefore, investing in something that has no apparent profit, causes disturbance for daily operations and is unlikely to realize is not a good business decision. However, 56% of the vulnerabilities in the report of Rapid7 were exploited within seven days of public disclosure [6]. The average cost of a security incident caused by a software vulnerability has long passed millions [6]. The risk is great and growing. The NIST SP 800-40r4 emphasises the risk, created as a side product of increasing digitalization within an organization. The approach is well suited for the purpose of impacting the leadership, as these are things that are within their control. However, the risk created by an increase of the criminal capability is also worth discussing. A major reason for this is automation. Malicious organized groups, and individuals also follow reports of the identified vulnerabilities.

The NIST SP 800-40r4 emphasises automation as an effective means of security management, but automation is also increasingly leveraged by criminals in their efforts to find and exploit vulnerabilities. Exploit Kits are tools designed for this purpose. According to O'Kane [9] they are software applications that scan and identify software vulnerabilities on client machines, with the intention of exploiting vulnerabilities to upload malware on the victim's computer. These are a key part of the modern attack infrastructure. Because of this, the threat landscape has changed. Negligent organizations are increasingly at disadvantage. Frameworks, such as Diamond Model [38], MITRE ATT&CK® Framework [39], Lockheed Martin Cyber Kill Chain® (CKC) [40] or the Unified Kill Chain (UKC) [41], illustrate phases of the malware attacks.

Publications like the NIST SP 800-40, the collection [8, 12, 13], are of great importance. They target the challenges, give context, and provide concrete recommendations for organizations at all levels of leadership. As discussed in numerous times, the basic concept of risk is the same, however, the likelihood of such risk has increased rapidly. It is crucial that all levels of leadership gain insight into the changing threat landscape, and how to compete against it. One of the remaining issues is the reliable estimation of risk probability.

Change is constant. In patch management, it is fast, too. It requires speed to react to new vulnerabilities, especially those of high risk. It also requires speed to provide new recommendations and insight in the face of the fast-changing environment. The NIST SP 800-40, has updated 4 times, in 20 years. It is worth to research whether or not such frequency is enough. For example, the SP 800-40 focus on planning comes with a cost. Albeit the older versions can familiarize the reader

with patch management best practises and technologies, these concepts are live. The publication on technologies from 2013 is simply outdated. In addition, software is used more often SaaS than local on premise installation. SaaS apps risk management and software patching differs from the on premise installed software.

## 7. Conclusions

As a conclusion, there exists credible research on software security patch management, on challenges, approaches, tools, and practises. Despite this, the monetary losses to unpatched vulnerabilities continue to ramp up. The patch management caused disturbance is still a cause for internal contest between the different departments in organisations. Considering this, the focus of NIST SP 800-40r4 is unique and necessary. Even so, the landscape changes, and as the organisations become more invested in security matters, all the other aspects of patch management become timelier than ever. The best solution is to install patches on one-for-one policy because a large number of vulnerabilities are being exploited before security teams have any time to implement patches or other mitigations.

## Acknowledgment

University of Jyväskylä is acknowledged for financial support of the research.

## References

[1] Ponemon Institute, "Cost and consequences of gaps in vulnerability response," 2019. [Online]. Available: https://media.bitpipe.com/io_15x/io_152272/item_2184126/ponemon-state-of-vulnerability-response-.pdf [Accessed 30 March 2023].

[2] Ponemon Institute, "State of endpoint security," 2018. [Online]. Available: https://dsimg.ubm-us.net/envelope/402173/580023/state-of-endpoint-security-2018.pdf [Accessed 30 March 2023].

[3] BC. Kim, P-Y Chen, and T. Mukhopadhyay, "An Economic Analysis of the Software Market with a Risk-sharing mechanims," *International Journal of Electronic Commerce,* vol. 14, no. 2, pp. 7-39, 2010.

[4] D. Dey, A. Lahiri, and G. Zhang, "Optimal Policies for Security Patch Management," *Informs Journal on Computing,* vol. 27, no. 3, pp. 462-477, 2015.

[5] J. Jacobs, S. Romanosky, I. Adjerid, and W. Baker, "Improving vulnerability remediation through better exploit prediction," *Journal of Cybersecurity,* vol. 6, no. 1, 2020.

[6] C. Condon, R. Bowes, E. Galinkin, "2022 Vulnerability Intelligence Report," Rapid7, 2023.

[7] M. Soppaya and K. Scarfone, "NIST Special Publication 800-40 Revision 4 Guide to Enterprise Path Management Planning: Preventive Maintenance for Technology.," National Institute of Standards and Technology., 2022.

[8] B. Ruppert, "Patch Management. SANS Institute.," 2008. [Online]. Available: https://www.sans.org/white-papers/2064/ [Accessed 20 April 2023].

[9] M. Souppaya and K. Scarfone, "Guide to enterprise patch management technologies," NIST Special Publication 800 (2013) 40, 2013.

[10] F. Li; L. Rogers, A. Mathur, N. Malkin, and M. Chetty, "Keepers of the machines: Examining how system administrators manage software updates for multiple machines," in *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, Santa Clara, 2019.

[11] C. Tiefenau, M. H¨aring, K. Krombholz, and E. von Zezschwitz, "Security, availability, and multiple information sources: Exploring update behavior of system administrators," in *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, Boston, 2020.

[12] U. Gentile and L. Serio, "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,* vol. 9, no. 1-2, pp. 115-132, 2019.

[13] N. Dissanayake, A. Jayatilaka, M. Zahedi, and M. Ali Babar, "Software security patch management - A systematic literature review of challenges, approaches, tools and practices," *Information and Software Technology,* vol. 144, 2022.

[14] F. Baiardi and F. Tonelli, "Twin Based Continuous Patching To Minimize Cyber Risk," *European Journal for Security Research,* vol. 6, p. 211–227, 2021.

[15] N. Dissanayake, M. Zahedi, A. Jayatilaka, and M. E. Babar, "Why, How and Where of Delays in Software Security Patch Management: An Empirical Investigation in the Healthcare Sector," in *Proceedings of the ACM on Human-computer Interaction*, 2022.

[16] T. F. Costa and M. Tymburibá, "Challenges on prioritizing software patching," in *15th International Conference on Security of Information and Networks (SIN)*, Sousse, 2022.

[17] T. August and T. I. Tunca;, "Who should be responsible for software security? A comparative analysis of liability policies in network environments," *Management Science,* vol. 57, no. 5, pp. 934-959, 2011.

[18] T. August T, M. F. Niculescu, and H. Shin, "Cloud implications on software network structure and security risks," *Inform. Systems Research,* vol. 25, no. 3, pp. 489-510, 2014.

[19] Y. Beres and J. Griffin, "Optimizing network patching policy decisions.," in *Information Security and Privacy Research*, New York, Springer, 2012, pp. 424-442.

[20] S. Tom, D. Christiansen, and D. Berrett, "Recommended practice for patch management of control systems (No. INL/EXT-08-14740). Idaho National Lab.(INL), Idaho Falls, ID (United States)," 2008. [Online]. Available: https://inldigitallibrary.inl.gov/sites/sti/sti/4121152.pdf [Accessed 20 April 2023].

[21] H. C. T. Gerace, "The critical elements of the patch management process," *Communications of the ACM,* vol. 52, no. 8, pp. 117-121, 2009.

[22] R. Yasin, "Patch Management Best Practices. Federal Computer Week 17.41.," 2003. [Online]. Available: https://www.proquest.com/trade-journals/patch-management-best-practices/docview/218874227/se-2 [Accessed 20 April 2023].

[23] J. Kaplan, "Saas: Friend or foe?," *Business Communications Review,* vol. 37, no. 6, 2008.

[24] M. Shakir, M. Hammoond, and A. K. Muttar, "Literature review of security issues in saas for public cloud computing: a metaanalysis," *International Journal of Engineering & Technology,* vol. 7, no. 3, pp. 1161-1171, 2018.

[25] P. Mell, D. Henning and T. Bergeron, "NIST Special Publication 800-40 Revision 2 Creating a Patch and Vulnerability Management Program," National Institute of Standards and Technology, 2005. [Online]. Available: https://csrc.nist.rip/library/alt-SP800-40v2.pdf [Accessed 20 April 2023].

[26] H. M. Sihvonen and M. Jäntti, "Improving release and patch management processes: An empirical case study on process challenges," in *In 2010 Fifth International Conference on Software Engineering Advances (pp. 232-237)*, Nice, France, 2010.

[27] G. Adams, "Patch Management: Change, Configuration and Release or Something More. Fox IT. https://www.itilnews.com/uploaded_files/Patch_Management_-_Article_for_itSMF_Conference.pdf," March 2007. [Online]. Available: https://www.itilnews.com/uploaded_files/Patch_Management_-_Article_for_itSMF_Conference.pdf [Accessed 20 April 2023].

[28] Ivanti, "Ransomware Spotlight Year End 2021 Report," Ivanti, 2021. [Online]. Available: https://www.ivanti.com/lp/security/reports/ransomware-spotlight-year-end-2021-report [Accessed 21 April 2023].

[29] US Cybersecurity and Infrastructure Security Agency, "Binding Operational Directive 22-01-Reducing the Significant Risk of Known Exploited Vulnerabilities," CISA, 2021. [Online]. Available: https://www.cisa.gov/binding-operational-directive-22-01 [Accessed 21 April 2023].

[30] Check Point, "Check Point Vulnerabilities Found," 30 January 2020. [Online]. Available: https://research.checkpoint.com/2020/remote-cloud-execution-critical-vulnerabilities-in-azure-cloud-infrastructure-part-ii/ [Accessed 8 May 2023].

[31] NIST, "National Vulnerability Database", 2023. [Online]. Available: https://nvd.nist.gov/vuln [Accessed 11 May 2023].

[32] "Carnegie Mellon University, Software Engineering Institute," 2022. [Online]. Available: https://www.sei.cmu.edu/about/divisions/cert/index.cfm#CERTRecentlyPublishedVulnerabilityNotes [Accessed 8 May 2023].

[33] "CVE Details," [Online]. Available: https://www.cvedetails.com/ [Accessed 9 May 2023].

[34] A. Walls, L. McMullen, J. Heiser, and D. Gopal, "Maverick Research: Risk Management Produces Bad Cybersecurity," Gartner, 2023.

[35] U.S. Cybersecurity and Infrastructure Security Agency (CISA), "Alert (AA21- 209A): Top Routinely Exploited Vulnerabilities," 20 August 2021. [Online]. Available: https://uscert.cisa.gov/ncas/alerts/aa21209a [Accessed 8 May 2023].

[36] I. Jibilian and K. Canales, "The US is readying sanctions against Russia over the SolarWinds cyber attack. Here's a simple explanation of how the massive hack happened and why it's such a big deal.," 15 April 2021.

[37] P. Marinescu and C. Cadar, "KATCH: High-coverage testing of software patches," in *ESEC/FSE 2013: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, Saint Petersburg, Russia, 2013.

[38] S. Caltagirone, A. Pendergast, and C. Betz, "The diamond model of intrusion analysis," US Department of Defense, 2013.

[39] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK®: Design and philosophy (Revised)," The MITRE Corporation, 2018.

[40] E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains.," *Leading Issues in Information Warfare & Security Research, 1,* 2011.

[41] P. Pols, "The unified kill chain," 2017. [Online]. Available: https :// www.unifiedkillchain.com/assets/The-Unified-Kill-Chain.pdf [Accessed 28 November 2022].

[42] D. Voldal, "A Practical Methodology for Implementing a Patch management Process. SANS Institute.," 2003. Availbale: https://www.sans.org/white-papers/1206/ [Accessed 20 April 2023].

[43] D. White, " Limiting Vulnerability Exposure through effective Patch Management: threat mitigation through vulnerability remediation," Rhodes University., 2006