Speedrunning the Maze: Meeting Regulatory Patching Deadlines in a Large Enterprise Environment

Gerbrand ten Napel
Delft University of Technology
G.H.TenNapel@tudelft.nl

Michel van Eeten
Delft University of Technology
M.J.G.vanEeten@tudelft.nl

Simon Parkin
Delft University of Technology
S.E.Parkin@tudelft.nl

Abstract*—Many enterprises struggle to apply security patches in time to remove the risk of security breaches. Delays can be attributed to technical dependencies, outdated asset inventories, and issues of scale. Governments have started pursuing a strategy of mandating through regulation the patching of a highly selective set of severe vulnerabilities under very strict deadlines. We worked with a large organization to examine the patching timelines under these regulatory deadlines. We analyze patching ticket-system entries for 81 security advisories over seven years, covering 944 CVEs. We complement this with nine interviews with professionals involved in managing patches. We find that 40.2% of advisories required patching action, with a median completion time of 13.2 days; advisories that do not end in requiring a patch have a median of 1.4 days. Completing the patching process in 48 hours – a recommended industry best practice – is achieved in just 16.2% of the cases. For the deadline of one week, under the Dutch BIO regulation, patching is achieved in 32.4% of the cases, while the performance against the typical CISA KEV deadlines is a bit more hopeful: 56.8% is patched in two weeks and 62.2% in three weeks. We find that some variance in delays can be explained by coordination effort, as measured by the number of involved teams and people. Overall, the strategy of regulatory deadlines for a highly selective set of priority vulnerabilities is associated with much faster enterprise patching. The deadlines are routinely missed, yet they need to trade off realism versus exposure. The three-week KEV deadline is more feasible than the 48-hour one, yet it also leaves open a longer exposure window for exploitation.

1. Introduction

By now it is abundantly clear that many known security breaches in organizations could have been prevented by applying a security patch that existed at the time of the attack [1], [2], [3]. The effective remedy for organizations would therefore seem as straightforward as timely patching. Indeed, this has been a mainstay of security advice and frameworks for many years. Yet, enterprises continue to struggle with patching on short notice [4], [5], [6].

In response to these difficulties, regulatory requirements have emerged in recent years. Some, such as the EU's Network & Information Security Directive (NIS2) [7], require organizations to take "appropriate" measures to handle vulnerabilities, and impose liability for when such measures are insufficient. Others go one step further, and impose mandatory patching deadlines for the most critical vulnerabilities.

The leading example of this strategy is the U.S. Cybersecurity and Infrastructure Security Agency (CISA)'s Binding Operational Directive 22-01 [8]. It legally requires federal agencies to remediate each vulnerability that is added to the CISA-managed Known Exploited Vulnerabilities (KEV) catalog within a specified deadline. Vulnerabilities listed in the catalog are prioritized as they are actively being exploited in the wild. Prioritization is a key part of this strategy: KEV contains a lot fewer vulnerabilities than those rated as critical under CVSS (Common Vulnerability Scoring System). The catalog currently contains just over 1,000 vulnerabilities – roughly 1% of all published vulnerabilities since 2021, when it was launched. The remediation deadline assigned to a new KEV addition is typically three weeks.

While the CISA directive is leading, it is not unique. In the Netherlands, where our study is located, there is a similar binding security policy for government organizations (called "BIO") that requires agencies to patch within one week after receiving a top-priority advisory from the national CERT [9]. Like with KEV, the idea is to achieve much faster patching by radically prioritizing which vulnerabilities to focus efforts on. Regulations like BOD22-01 and BIO are not binding outside government, but they are a recommended best practice for private enterprises as well [10].

Does regulatory pressure lead to shorter patching timelines? Are the KEV or BIO deadlines achievable for enterprises? A deadline of three weeks might not sound that strict, but it is an order of magnitude faster than the time enterprises normally take to patch vulnerabilities, as reported by Kotzias et al. [11]. In 2018, those authors analyzed telemetry from 28,000 enterprises and found that it takes 9 months to patch 90% of all server-side vulnerable systems – with many sectors take up to two years. The study does not report patching speed for highly critical vulnerabilities specifically. And it does not analyse the effects of new regulations, since it precedes those.

Beyond [11], there is a remarkable lack of prior work measuring patching speed (Section 2). There are a number of internet measurement studies that looked at the patching

speed of hosts for a specific vulnerability [12], [13], but these do not look at enterprises, nor at the vulnerabilities that are targeted by CISA KEV and BIO. Other papers on enterprise patching are based on interviews or surveys among system administrators [3], [14], [15]. These papers provide insight into the self-reported behaviors of sysadmins. They lack any data on actual patching timelines, let alone patching under regulatory deadlines. In fact, no study directly mentions regulation as a prioritizing factor.

To overcome this gap in the prior work, we present the first study on enterprise patching timelines under regulatory deadlines (with method described in Section 3). Our partner organization is a large public infrastructure provider in the Netherlands with over 10,000 employees and over 1,000 IT staff. We measure patching timelines of top-priority vulnerabilities across seven years (2015–2022). The enterprise operates under the compulsory BIO deadline of patching top-priority advisories of the national CERT in one week, as well as its own internal policy to patch them within 48 hours. The latter is even more strict than the deadlines of KEV or BIO, though it is consistent with non-binding advice of CISA and others: "Organizations should patch vulnerable software and hardware systems within 24 to 48 hours from when a vulnerability is disclosed" [16].

Our goal is to bring evidence to bear on the effectiveness and feasibility of the strategy to mandate patching deadlines for a small set of prioritized vulnerabilities. We reconstruct the patching timelines via the ticketing system that is being used for tracking the progress. We track 399 top-priority patching (sub)tickets with over 3400 timestamped comments, covering 81 security advisories and 944 CVEs. We reconstruct patching timelines and compare them to the binding deadlines of BIO and CISA KEV, as well as against the industry advice (and the internal enterprise policy) of 48 hours. To support our interpretation of the ticketing data, we conducted nine semi-structured interviews with professionals involved in the patching process. Our goal is to answer the following question: *Is patching under regulatory deadlines associated with shorter patching timelines?*

Of course, ideally we would measure patching speed across a large number of enterprises. However, research on real-world enterprise security is dominated by case studies – [11] being a rare exception. This reflect the difficulties of obtaining access to such privileged data. To assist in the generalizability of our findings, in our Discussion in Section 5 we situate the partner organization within the reported patching timelines of Kotzias et al. for different sectors, as well as the results of a very recent industry report on KEV remediation [5]. Our contributions are as follows:

- We present the first scientific study of real-world enterprise patching timelines under regulatory deadlines using ticketing logs and comments to identify how long it took overall and how much time was consumed by various activities in the process.
- In our case study, IT staff concluded that actual patching actions were needed for 40.2% of all mandatory advisories from the national CERT. For advisories that do not end in patching, there is still a median of 1.4 days before the organization is able to conclude that no patching is needed. This demonstrates that the prior steps of identifying potentially vulnerable systems inside the organization take most, if not all, of the time within the deadline of 48 hours.
- When actual patching actions are needed, completing a ticket within 48 hours is achieved in just 16.2% of cases. Patching within the mandatory BIO deadline of one week is achieved in 32.4% of cases. When we compare the performance against the typical KEV deadline of three weeks, it looks more hopeful: 62.2% is completed in time. Overall, these timelines are almost an order of magnitude faster than the timelines reported by Kotzias et al. [11], the best available measurement of enterprise patching speed. In other words, our evidence supports the effectiveness of the policy to impose regulatory deadlines for a highly selective set of vulnerabilities.
- We find that some variance in patching time can be explained by the number of involved teams and people, as well as coordination efforts (including whether to take down a service for patching). There are clear correlations between the patching time and the number of involved teams ($\rho = 0.46$), people ($\rho = 0.48$) and comments in the ticket ($\rho = 0.45$).

2. Related work

There are two main bodies of work on measuring and understanding patching timelines in enterprises: (1) tracking vulnerable systems over time via network scans; and (2) using surveys and interviews to capture system administrator perspectives on patching within enterprise environments.

The first is based on internet-wide scans that longitudinally observe vulnerable hosts – e.g., [4], [11], [12], [13], [17], [18]. These studies quantify the time delay of patching at the host level, but they do not distinguish hosts in enterprise environments. Neither do they report on vulnerabilities under regulatory patching deadlines.

A rare exception is [11]. It presents a wide range of measurements of patching speed across different enterprises in different sectors. While the authors do mention including vulnerabilities with critical CVSS ratings, they do not report patching timelines specifically for these vulnerabilities. Also, the set of vulnerabilities that CVSS rates as critical is about two orders of magnitude larger than the set in KEV or BIO; it is thus much less selective, and hence less representative to evaluate the strategy adopted by recent regulations. Finally, this study is from 2018, before any of these regulations were adopted.

The second body of related work is based on interviews and surveys on the self-reported behavior of sysadmins and other IT professionals involved in enterprise patching. This has revealed common phases of the patching process together with a variety of challenges administrators face within these phases [3], [15], [19], [20]. A recent study [14]

reported that few sysadmins identified an internal policy as setting priorities for patching. No studies directly explored regulation as a factor. More importantly, none of these studies contain empirical estimates of patching timelines.

The closest work in this direction is Dissanayake et al. [21], which combines interview data with meeting minutes. From the minutes they identify tasks, which are assumed to have a certain duration ("standard time frames"). They then combine tasks with assumed durations to produce "patching delays", but these are not empirical observations.

Our study is the first to conduct a measurement of patching timelines under mandatory policies. The only other analysis that we recently encountered is a concurrent industry publication from May 2024 [5]. It claims to track patching of KEV vulnerabilities across industries, but does not provide any transparency about its data collection methods, other than "scanning". We complement our measurements of the timelines with a small sample of interviews, to support the interpretation of the ticketing data and to explore some factors associated with the variation of timelines across different vulnerabilities.

3. Methodology

3.1. Organization Context

The organization where we conducted our study is a large government agency that operates critical infrastructure. It employs over 10,000 people, including more than 1,000 IT staff. It has a Security Operations Center (SOC), complemented by a general 24/7 operational incident response department that covers all domains, not just IT. In case of an urgent threat in IT, the SOC aims to contact the affected teams within the hundreds of IT teams throughout the organization.

The organization receives security advisories from the national CERT. The advisory informs recipients about important new vulnerabilities or threats. Some of these advisories are labeled by the CERT as "high impact, high probability", the highest severity level – colloquially referred to by security professionals as "high-high". The organizational policy is that advisories need to be taken care of within 48 hours. Beyond this internal policy, the organization is mandated under the BIO regulation [9] to meet the deadline of one week.

In order to investigate how these advisories are handled by the organization, and whether the deadlines of 48 hours or one week are met, we analyze data from the organization's main ticketing system. For each high-high advisory received by the organization, the SOC staff does an initial triaging to assess whether the organization potentially has the software or hardware that the advisories pertains to. Once the SOC decides that the organization has, or might have, assets with the vulnerability, they open a 'main' ticket to initiate the patching process. This ticket registration is compulsory and it allows us to track all patching activities based on the national CERT high-high advisories that passed initial triaging by the SOC.

After creating a main ticket, the next step for the SOC analyst is to be more specific as to where in the organization vulnerable assets might be located and whom should be addressed for patching them. This is executed by creating sub-tickets and assigning those to so-called 'solution groups'. These groups represent IT teams that maintain specific kinds of software, and the solution group members notify possibly affected teams. Within these IT teams, there are administrators that eventually patch vulnerable systems. The team might also conclude that the advisory is sufficiently dealt with without actually deploying a patch – for example, the software version they run is not vulnerable or some other mitigations are taken instead of patching. They can propose to close the sub ticket, citing their reasons for doing so. The SOC members have a final say in 'accepting' the solution and affirming a ticket close.

3.2. Ticketing Data

3.2.1. Advisory data. The national CERT issues advisories and then sends emails to subscribed organizations. Recipient organizations can choose to receive medium-priority advisories or only high-priority ("high-high") advisories. Our organization has a closely monitored mailbox for the latter. Advisories can also be retrieved at the national CERT website [22].

We log all available features from the advisories. They contain the description, risk level, possible consequences, and possible solutions to the vulnerability. For some advisories, a list of CVEs is included, and for some, a list of affected platforms is provided as well. Each advisory has an issue date and an identifier that enables us to link the advisory data to tickets. An incoming advisory might either be completely new or an update of a previous version, as indicated by a version number.

3.2.2. Ticket data. Our analysis relies on the ticket data to reflect actual patching activities, and especially timelines, with sufficient accuracy. Opening, evaluating and closing tickets is a dedicated task for the SOC, rather than by the teams that patch. The SOC has every incentive to track patching activity accurately, and their oversight role also adds pressure for practitioners to evidence their activities in the ticket as promptly as possible.

When a high-high advisory email comes in, it is analyzed by SOC. When deemed applicable, a ticket is created. All these tickets were exported by an employee specialized in managing the ticketing system. All individual names were removed from the exported data. The export contained 399 tickets that were both main and sub-tickets, which have a parent-child relationship. We removed one main ticket with its four sub-tickets as it concerned a test run. Also, we removed two main tickets with their eight sub-tickets, as they were still in progress at the time of export. Five sub-tickets could not be related to main tickets, so we left them out. Two main tickets included insufficient information to relate them to the relevant advisory, so they were removed

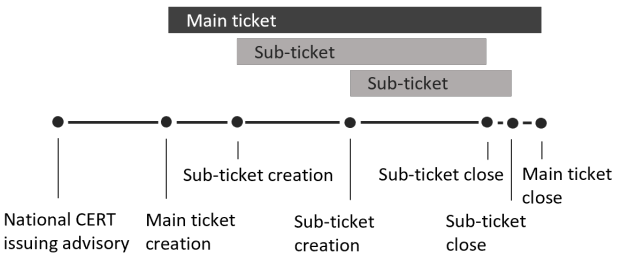


Figure 1: Measurable process timelines.

with their 2 sub-tickets. This left us with 373 tickets for analyses, being 92 main and 281 sub-tickets.

These main tickets represent the overall organizational process, whereas the sub-tickets represent organizational sub-sections handling the advisory. Main tickets can only be closed after all their sub-tickets are closed. This ticket hierarchy is illustrated in Figure 1 with an example of a main ticket with 2 sub-tickets and the accompanying timestamps we use to calculate timelines. Amongst other attributes, main and sub-tickets have creation and close timestamps, an ID, and a short and long description that often contains the entire advisory text, including its advisory ID, allowing us to relate tickets to advisories. Another ticket field contains all comments with a time stamp and an anonymized username. These comments often concern people reporting, asking for status updates, and sharing information on the vulnerability, assets, owners, or reasons why a ticket can be closed.

3.2.3. Labeling ticket outcomes. When analyzing the tickets and sub-tickets, an important first finding was that tickets do not necessarily lead to patching. Therefore, we used the ticket comments to categorize tickets by their outcome. In order to create categories, a researcher with a background of many years in both patching and using ticketing systems analyzed all tickets. The categories were developed by analyzing the comments where employees often explicitly discussed why the ticket could be closed. Some cases require more extensive analyses of comments. One set of tickets did not contain information to confidently categorize the outcome; those tickets were put in the category 'unknown'.

The initial set of categories was developed inductively by the same researcher who analyzed the tickets, following a coding reliability style of Thematic Analysis (TA) [23], where categories had been derived from regularly-seen outcomes during the ticketing data analysis. Initial categories were discussed at regular intervals among the whole author team. This led to an initial set of nine ticket outcome categories. In the next step, two researchers categorized 100 random (sub)tickets, involving the researcher who analyzed the tickets and a researcher with a social sciences background who was familiar with qualitative research and cybersecurity, but was not directly involved in this project. In 74% of the cases, the categorization was the same. For the divergent cases, various iterations were made, with input from the two other authors. This resulted in consensus on a set of seven categories: two categories of 'duplicate' and 'transferred' were consolidated into 'Taken care of elsewhere'; and 'partly patched' and 'completely patched', which were consolidated into 'Patched'. We apply these to both main tickets and sub-tickets. The definitions of the categories are as follows.

Patched. We label a sub-ticket as 'patched' if there is any reporting of patching of the vulnerability, regardless of the number of patched systems. Mitigating measures might have been applied prior to actual patching. If a main ticket is reporting on patching activities, or contains sub-tickets that are labeled as patched, we label the main ticket as 'patched'.

Was already patched. We label a sub-ticket as already patched when the ticket comments tell us that the vulnerability mentioned in the advisory was already patched and no additional patching was done. The main ticket is labeled 'was already patched' if this is reported in the main ticket comments or if its sub tickets are labeled 'already patched' with no sub tickets that involved patching.

Will be patched later. We label a sub-ticket as 'will be patched later' if the ticket reports that patching will be done at some point in the future, after the close of the ticket. In case a main ticket reports on later patching, or has any sub-ticket that is labeled as 'will be patched later' with no sub-tickets in the above categories, it would be labeled as 'will be patched later'.

Mitigated. We label a sub-ticket as 'mitigated' if the ticket shows reporting of mitigating measures and none of the activities fit in the above categories, so no patching actions were taken. If a main ticket reports mitigating measures or has any sub-ticket that is labeled as 'mitigated' and no sub-tickets in the above categories, it is labeled 'mitigated'.

Not applicable. We label a sub-ticket as 'not applicable' when the comments report that an advisory refers to software or a software version that is not present within the responsibility of the ticket assignees. We label a main ticket as 'not applicable' when it concludes that the vulnerability did not affect the organization and that none of the above activities were reported in the main or its sub-tickets.

Taken care of elsewhere. We label a sub-ticket as 'taken care of elsewhere' when the ticket comments describe that the advisory is taken care of in another sub-ticket or when it is dealt with by a third party. Main tickets are labeled 'taken care of elsewhere' when this is concluded in the comments or when all sub tickets are 'taken care of elsewhere.'

Unknown. We label a sub-ticket as 'unknown' when we cannot confidently assign it to any of the above categories based on the ticket comments. Main tickets are labeled 'unknown' if neither the comments nor its sub-tickets conclude on any of the above categories. For 'unknown' tickets, the timings are clear, and the ticket was resolved. However, the activity that occurred during the resolution of the ticket is not clear enough to place it in a specific category as above.

3.2.4. Patching deadlines. We compare patching timelines to the organization's internal policy of 48 hours, to the binding one-week requirement of BIO [9], and to the typical CISA KEV deadline of three weeks [10]. The comparison

with the KEV deadline is performed for a subset of 49 tickets (53% of all tickets) since the associated CVEs were present in the KEV catalog as well as in the set of "high high" advisories that the organization opened a ticket for.

The start date of the 48-hour policy was not fully clear. Respondents mentioned that it was related to BIO. BIO came into force on January 1, 2019. If we assume that the 48-hour policy came into force at the same time, then we should be careful when evaluating pre-2019 cases against that deadline. It seems unfair to hold the organization to that standard. That being said, we have two reasons to still make that comparison. First, we are not passing judgment or formally assessing compliance. Instead, we want to understand how the imposed deadlines relate to real-world efforts to patch top-priority vulnerabilities. Second, patch times did not visibly improve after BIO came into force. We looked at how many tickets meet the 48-hour deadline before and after January 1, 2019. Before 37.7% of the tickets met the deadline, after 30.4%. So we are not biasing the assessment against the organization by including the cases that predate BIO. The small difference makes sense as the high-high advisories had a special status that gave them absolute priority, even before the BIO imposed formal deadlines. This is reflected in the fact that almost all tickets were assigned "prio 2", both before and after January 1, 2019.

Next to BIO, we look at BOD 22-01. Of course, this policy was never directly applicable to our partner organization, as is not based in the U.S. That said, other organizations, also internationally, use KEV as a way to prioritize their patching strategy [5], so the comparison is meaningful.

Since the internal policy is the most strict, the organization is trying to complete the process as fast as possible, even after 48 hours. This means all cases can also provide valuable evidence of patching speed in relation to BIO and KEV. Contrary to the internal policy and the BIO deadline, KEV assigns a specific deadline to each vulnerability that is added to the catalog. CISA published the KEV catalog in November 2021. It started with 287 vulnerabilities and it is now a repository of over one thousand CVEs. To get the KEV timeframes, we used its catalog at [10]. There is some variability in KEV deadlines. The initial batch of 287 was assigned a longer deadline, since it concerned a lot of vulnerabilities that were added to the catalog at the same time. After the bootstrapping phase of the catalog, most new additions to KEV are assigned a deadline of two or three weeks. For simplicity, we report on the results for both deadlines in our comparison.

3.3. Semi-Structured Interviews

As we neared completion of the ticket analysis and had developed some understanding of the phenomena observable in the tickets, we conducted nine interviews with practitioners involved in various ways with top-priority advisories. The interviews were conducted with a focus on the ticket data, and improving our understanding of phenomena we saw from the ticket analysis.

The interviews were organized in two rounds, each with a slightly different protocol. In the first round, we interviewed four SOC team members and a SOC coordinator who are the first ones to be involved when a high-high advisory comes in. This interview started with a broader discussion of the process when an advisory comes in and a ticket is started. This was followed by more specific discussion of the factors that cost time within that process.

In the second round of interviews, we interviewed four people who are located in what are called 'solution groups', being two coordinators and two technical application managers. These groups are more closely involved in the process of patching and the business of operational IT. These interviews also opened with discussion of the process for incoming high-high advisories, and a part on the factors that take time in patching. This was complemented with focus on a few tickets that were addressed to their solution group, asking them if they recognized the ticket and if they could comment on what was happening and related timelines. The question sets can be found in the Appendix.

3.3.1. Recruitment. We recruited our participants for round one by asking an internal security advisor to address a cross-section of the SOC team, sending them an email and asking for participation. This resulted in the recruitment of two coordinators (P1, P2) and 3 analysts (P3, P4, and P5). The interviews took 60-90 minutes, except for one with an analyst who only had 20 minutes available (which also serves to highlight that these are difficult to reach participants, actively working to secure infrastructure).

The second round involved the coordinator contacting practitioners working with high-high advisories who are closer to operational IT. This resulted in the recruitment of two coordinators (P6, P9) and two technical application managers (P7, P8). Interviews here lasted 30-60 minutes.

3.3.2. Analysis of interviews. Regarding analysis of the interviews, a Thematic Analysis (TA) approach [24] was used, broadly exploring factors which may affect the timing of the patching process. Interviewees were asked about the patching process in an open manner, though around what the steps were in the patching process. This was informed by the analysis of the ticketing data, where the analysis of the interviews then helped us to better understand the phenomena we saw in the ticket analysis.

The protocol was driven by 'codebook' style TA, looking for examples that linked the dataset and explanations of the trends seen in the ticketing data. In that sense, we leverage the themes emerging from the coding to broadly explain trends in the dataset (where trends correlate broadly with the sub-subsection headers in Section 4). The same author who analyzed the ticketing data also conducted the interviews, and coded the anonymized transcripts. inter-rater reliability is not usually a direct measure of quality for the kind of 'codebook'-based analysis we conducted [23], where here there was highly regular discussions between the coder-author and other co-authors about the codes and emerging themes (which then formed the topics discussed in

our Results). The coding meetings were conducted between the authors at regular intervals (of every two weeks or less) to discuss notable themes and explanatory factors, refining the codebook.

3.4. Ethics

The ticketing data was anonymized and shared for analysis. When interviewees needed to be identified and contacted, this was done by an employee of the organization who was our liaison. We conducted interviews where people elaborated on their professional behavior and experience around patching within the organization. The interviews were approved by our Institutional Review Board. All interviewees were informed about the purpose of the research and signed an informed consent document that explained the use of the data, the associated risks, and the research purposes. All interviewees had the chance to review the paper that reflected their contribution and possible quotes. They were allowed to opt out of the research at any point.

3.5. Limitations

As with most work in enterprise security, we rely on a case study approach. The obvious limitation of any case study is the question of generalizability. We revisit this in the Discussion section, where we do a close comparison between our findings and other works based on higher-level analysis of many enterprises [5], [11].

A limitation with respect to the main data source is that our analysis relies on the ticketing system being used effectively. The high importance and deadline for high-high vulnerabilities give employees a further incentive to document rigorously and close tickets properly. Of the 399 main and sub-tickets we analyzed, 17 tickets were closed without any comments and categorized as 'unknown' – a small minority of tickets. Another limitation is that we were not present as tickets were being worked through (where e.g., some studies on SOCs have embedded a researcher in the team itself [25]). However, as part of the interviews we stepped through a few examples of previous patches.

It can be argued that the practitioners themselves would, ideally, be the ones to apply the researcher-derived ticket categories to tag tickets and determine their meaning relative to our ticket categories. We did discuss specific tickets with the interviewees, but were mindful of limiting the amount of their time we asked for, given that they are busy practitioners in a high-pressure role.

4. Results

Each National CERT's so-called "high-high" advisory, when deemed relevant by the SOC staff, leads to the opening of a main ticket. If needed, subsequent sub-tickets are created to engage relevant organization units in identifying and working with affected assets.We extract the patching time from the ticketing data: the time between creating and closing a ticket. There is also a prior delay, between advisory publication time and creation of a main ticket (with an initial assessment of whether the organization has assets that are affected by a vulnerability). We quantify and discuss this delay in our analysis, using the publication date and time for the advisory.We collected and analyzed 399 main and sub-tickets and their corresponding advisories. To answer our first research question, we quantify the amount of time it takes to close the ticket (Section 4.1). We then describe the patching process and time spent in each step (Section 4.2). We answer our second research question and investigate factors influencing the duration of patching (Section 4.3).

4.1. Patching time

The distribution of ticket resolution times is very skewed. Closing a main ticket took 23.5 days on average, with a median of 5.5 days. The big difference between median and average shows that the distribution is heavily right-skewed, indicating a number of outliers with unusually long closing times. Figure 2 shows the distribution of completion times over tickets. After 5.4 days, 50% of all main tickets are closed, while it takes 26.9 days to close 80%. At the level of sub-tickets, the average time was 12.5 days and a median of 2.8 days.

The fastest main ticket took 2 hours and 20 minutes to be closed. The vulnerability affected Debian Linux, FreeBSD and SUSE Linux, but since the organization was exclusively on RedHat at the time, they were quick to conclude that they were not affected. This is indicative of a ticket where the time costs are essentially all related to coordination – and yet it still took a few hours. The fastest sub-ticket took only 5 seconds as it was switched to 'ready', with a single comment was added 'this was already addressed this morning'.

The longest delay occurred for the infamous Spectre and Meltdown vulnerabilities where it was a sub-ticket that took 308 days to resolve. It makes sense that these tickets remained open for nearly a year. There was widespread uncertainty on how to mitigate these vulnerabilities, or whether to mitigate them at all. There were patches available, but these are better understood as mitigations, since they did not remove the underlying side-channel vulnerabilities, and they came with significant performance losses, especially in server settings [26]. At the same time, there was no evidence of real-world exploitation. This led many organizations to adopt a wait-and-see attitude before adopting mitigations. As such, the ticket for this vulnerability shows a lot of deliberation on multiple levels, particularly on performance loss and related costs. Patches are applied and tested for performance, but no further patches were activated at the time due to serious performance concerns. For some platforms, a strategy of moving to other hardware was considered.

Of the 92 main tickets, only 35.9% were solved within the 48-hour deadline specified in organizational policy. For tickets that required actual patching actions, a mere 16.2% was completed within the deadline.

We also compared the main ticket completion times to the Baseline Information Security Government (BIO)

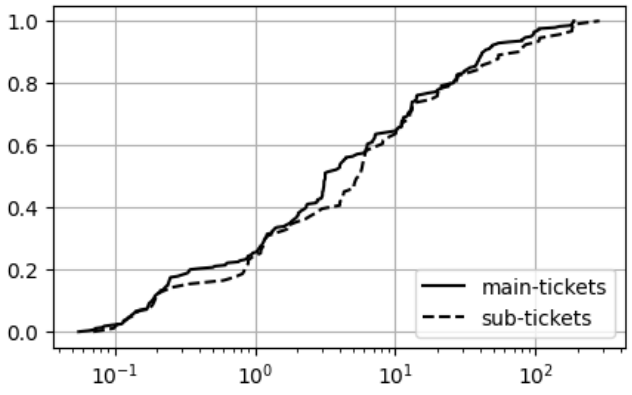


Figure 2: Cumulative Distribution Function (CDF) of main and sub-ticket duration in days for all categories.

version 1.04 [9], which has a deadline for National CERT high-high advisories of one week. Closing a ticket within a week is accomplished for 58.7% of main tickets. For tickets that included actual patching activities, this was 32.4%.

We found that the CVEs of 49 advisories were also included in the KEV catalog. When we compared the completion times for these matching tickets to the typical KEV deadlines, we found that 81.6% were solved in two weeks and 87.8% in three weeks. When we compare all tickets to the KEV deadlines – so not just tickets that overlap between KEV and our dataset – we can see that 73.9% are solved in two weeks and 78.3% in three weeks. For tickets that require actual patching activities, 56.8% are completed in two weeks and 62.2% in three weeks.

These results paint a picture of patching under regulatory pressure that is considerably faster – in fact, nearly an order of magnitude faster – than the main prior work [11] has measured for enterprise environments. Kotzias et al. found that patching 90% of all vulnerabilities took 23 months for enterprises in the transportation sector. We find that it took less than 2 months to deal with 90% of all high-high vulnerabilities (Figure 2). If we only look at the cases where actual patching actions were needed, then it took 5 months for 90% to be completed. In short: regulatory pressure to patch a very select group of priority vulnerabilities seems to be effective in getting much faster patching to take place. Yet, we also see that the deadlines are routinely missed. The more strict, the lower proportion of ticket cases that manage to meet it, especially if actual patch deployment is required. In the case of BIO, only one in three patching processes meet the regulatory deadline.

4.2. Patching process

Now that we have a high-level answer to the main research question of whether regulatory pressure is associated with faster patching, we can delve a bit deeper in the various stages of the patching process. Interviewees helped to clarify that three main activities take place, though not in a strict order. The first phase is to ingest the advisory and asses whether it potentially affects the organization, as basic triage. Next, the focus shifts to identifying and engaging relevant 'solution groups', teams, owners, application managers, and ultimately system administrators. The identification often involves more triaging, as each party needs to interpret the advisory and analyze the applicability to its own systems. Finally, there are activities concerning the analysis of software for the presence of the vulnerability. This can include taking mitigating measures or carrying out actual patching. Below, we elaborate on these three main phases: (1) advisory intake, (2) identifying and engaging relevant stakeholders, and (3) software-related activities.

4.2.1. Advisory intake. In our timeframe (August 2015 – November 2022), the national CERT issued 461 "high-high" advisories, which means an average of 5.3 advisories per month. These 461 include version updates of prior advisories, which the national CERT sends out as new advisories. If we do not count these updates as new advisories, there are 183 unique advisories. This results in 2.1 completely new incoming advisories per month, on average.

Usually, an advisory version update is added to the existing ticket that was created for the initial advisory. However, SOC members might also create a new ticket if, for example, they are unaware of that existing ticket. This happened in eight cases. In three cases, a duplicate ticket was created for the same advisory version. The 92 main tickets that were created thus reflect 81 unique advisories. So, if we consider only the unique advisories, 44% of the high-high advisories are deemed potentially applicable by the SOC and lead to opening a ticket.

When the national CERT publishes a high-high advisory, it notifies the organization via email; this is in contrast to the kinds of community-formed notification services identified by e.g., [14], where their participants appeared to not be driven as much by regulatory expectations. The email is directly addressed to the SOC and to the 24/7 operations team as well, since the operations team contacts the SOC outside office hours. The first step is that a SOC analyst estimates if there is any chance the affected software is present in the organization. In familiar cases, this is known by heart – if not by the SOC analyst themselves, then perhaps by a colleague. Consulting other SOC members, therefore, can be fruitful. In other cases, asset management tools like a CMDB (Configuration Management Database) are used. This is far from straightforward, however. The organization maintains six different CMDB-like databases, illustrating the unsolved problem of asset management in complex environments.

Sometimes, logging tools like the ticketing system or security information and event systems (SIEM) are queried to find mentions of affected software. As P5 puts it: "In an ideal situation, you have a central CMDB where everything can be found. The reality is that this is often more complex, so it is often a matter of sorting out between ticketing systems, CMDBs, and SIEMs where we aggregate event logging. See [...] whether we use that product." When still

in doubt, network scanning tools might be used to recognize software-specific network traffic.

Once the affected software is found, or its presence is deemed likely, a main ticket is created. The potentially affected departments are added to the ticket, although the search for additional affected departments might continue. A short description of the vulnerability is added to the tickets, and usually, the advisory information provided by the national CERT is added as well.

We calculated the delay between the advisory release time and the main ticket creation, to quantify the initial intake delay. The advisory intake duration has a mean of 4.0 hours (median: 2.6 hours, min: 0.8 hours, max: 23.5 hours). This intake does add some delay to the overall process, especially for the very short-lived tickets, even if it is not a major factor.

4.2.2. Identifying and engaging relevant stakeholders. After a main ticket is created, and the first possibly affected departments are identified, the next goal is to contact all affected IT teams within these departments. This is done in sub-tickets. When a sub-ticket is created, it needs to be assigned to the relevant specialized IT teams that maintain the affected software together with the software owner. However, there are about 850 such teams, so it is nearly impossible for the SOC to know which teams to contact.

To help coordinate activities, the organization has introduced so-called 'solution groups' which have specialized knowledge of the teams within the various departments. As expressed by P5: "That is one of the difficult things because we cannot, of course, identify all [850] teams, so how we have set this up now, is that we have umbrella solution groups ... so if we say, we want to ask department X, but we don't know exactly who and where, then we put it there, and then they distribute it further among department X." A solution group can, for example, represent 'operational Linux' and affect all teams involved in Linux hosting; another solution team is called 'workplace', representing all teams involved in laptops and PCs for personnel. The solution group continues the search for affected software, owners, and teams, although the SOC might also stay involved. The 92 main tickets mentioned above have 281 sub-tickets, so main tickets have an average of 2.6 sub-tickets with a median of 1.0 and a minimum of 0. The sub-tickets were assigned to 75 unique solution groups.

The affected teams subsequently start their own analysis of the vulnerability and how it affects their systems. Sometimes, however, the teams were already aware of the vulnerability and busy dealing with it (where other work [2], [14] has noted proactive efforts instead to anticipate problems with patches, rather than vulnerabilities that need action). In some cases, they had even already completed the deployment of the patch. As P3 commented: "It is up to the administrator whether they take a proactive approach. You can really see that it differs per management team. One is very proactive about it, they open this security website every morning to see if there are any vulnerabilities in their systems. And others are waiting to be triggered ... and if you

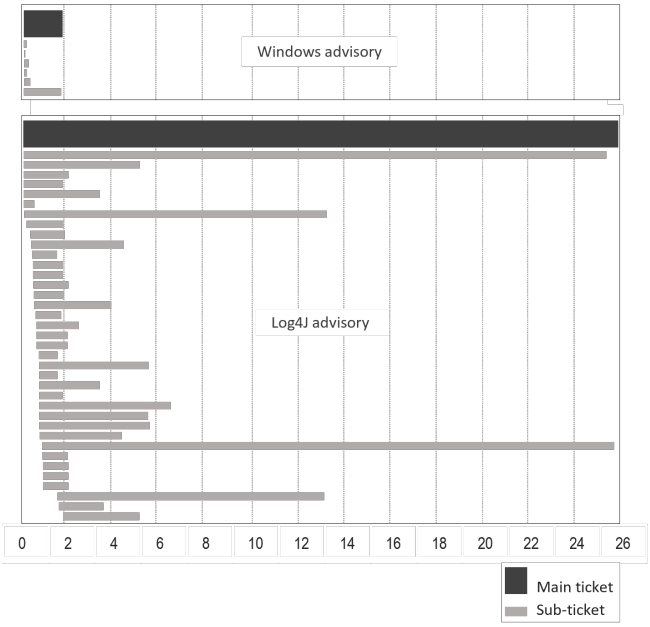


Figure 3: Examples of ticket duration in weeks.

are reactive, then it will take 1 or 2 days before it comes to you, and then you can get started".

We illustrate this process of finding who is involved with a particular patch with two ticket examples, as displayed in Figure 3. The top example is a main ticket of a typical Microsoft Windows advisory. The large black bar represents the main ticket duration, and six smaller bars represent sub-tickets. We see that these affected teams were found quickly as the sub-tickets started practically simultaneously. The second sub-ticket bar is nearly invisible as it was closed within 2 hours; the affected team was already applying the patches. The second example shows the tickets for the well-known Log4J vulnerability; note how sub-tickets were created as time progressed and new vulnerable software packages and instances were discovered. This differs from e.g., the sysadmins documented in [14], who appear to rely on patch notifications that refer to specific systems (i.e., the notification implies where the patch should happen).

These examples show that identifying the relevant solution group and teams can be a matter of hours, or potentially weeks. Our participating SOC analysts, coordinators, and application managers commented on the difficulties that can arise in connecting tickets to teams. For that reason, so-called 'process coordinators' can be involved, who "ensure that incident, problem, and change, even if this process falls apart, they still make it work" as stated by P2.

4.2.3. Software-related activities. Once sub-ticket assignees conclude their work on an advisory, they change the ticket status to 'ready'. This often comes with a comment that provides reasoning as to why the sub-ticket can be closed. The reasoning is typically more extensive when a decision is made *not* to patch. Subsequently, the registrants of the sub-ticket, i.e., SOC analysts, are notified automatically.

Category	No.	Perc.	Mean (days)	Med. (days)
Patched	37	40%	41.3	13.2
Was already patched	3	3%	2.4	1.1
Will be patched later	0	0%	empty	empty
Mitigated	4	4%	7.5	4.1
Not applicable	25	27%	8.6	1.4
Taken care of elsewhere	6	7%	1.4	0.8
Unknown	17	18%	22	4.4

TABLE 1: Main tickets categorized by the outcome and the mean and median duration of each category.

Category	No.	Perc.	Mean (days)	Med. (days)
Patched	70	25%	20.3	5.9
Was already patched	8	3%	0.2	0.0
Will be patched later	4	1%	25.5	22.3
Mitigated	20	7%	11.2	9.8
Not applicable	91	32%	7.1	1.9
Taken care of elsewhere	44	16%	8.2	2.9
Unknown	44	16%	16.2	1.1

TABLE 2: Sub-tickets categorized by outcome and the mean and median durations for each category.

They can then either approve and formally close the ticket or they can decide to reopen it, P3: "[...] because we are the registrants, we get the message 'it's closed, check if you agree'. And then we have had cases where we have said we disagree with your answer. We reopen it." Where other work – e.g., [14] – discusses formal risk assessment processes, we see here another side, of documenting one's patching actions, in part in anticipation of regulatory oversight, and that this in itself takes effort. This could constitute another 'constraint' on patching processes [27].

There can be a back-and-forth about whether patching is required, but the authority ultimately resides with the system administrators. As P3 describes it: "We report [the vulnerability] and we assign it. It is very stern advice and if I were to get mad, I think of it as compulsory advice. [...] If the administrator says 'yes, that patch, I get it, but it causes the functionality to collapse', then the administrator can argue 'OK, we understand the security risk, but the application will stop working if we patch'. Then the administrator can say "I'm going to patch it later' or 'we're going to test it first' or 'we're going to change something', et cetera." Where other work centres on sysadmins *deciding* how to patch [14], or a need to work in line with organisational policies [3], here we see where regulatory expectations are driving patching actions, reducing the room for sysadmin discretion.

4.2.4. Outcomes of the patching process. When all its sub-tickets are closed and agreed on by SOC members, the main ticket can be closed as well. For both main and sub-tickets, we analyzed the justification for ticket resolution, which allowed us to categorize ticket outcomes – as explained in Section 3.2.3. For main ticket outcomes, we used the main ticket conclusion, and when in doubt, sub-ticket outcomes were considered.

In Table 1, we classify the outcome of all main tickets across 7 categories. For 40.2% of main tickets, the outcome is that patching took place with a statement that there was no need for additional patching. For 27.2% of the main tickets, the outcome was a decision that patching was 'not applicable' – a conclusion was reached that no patching was needed. This can be due to different reasons. We observed several cases where the IT team concluded "we have a different version" than the version that has the vulnerability.

Some 17.6% of the main tickets are labeled 'unknown'. These tickets have either no comments or the conclusion of the ticket outcome is not clear. For 6.6% of the main tickets, the outcome was 'taken care of elsewhere'. This mostly reflects communication issues, such as the accidental duplicates mentioned before, when someone is unaware that a colleague has already created a ticket. For 3.3% of the main tickets, administrators were already done patching before the national CERT advisory reached them via the ticketing process. Those tickets are labeled 'already patched'. They reflect a proactive search for vulnerabilities.

Table 2 shows us the numbers of sub-tickets and solution times. Here, 'not applicable' is the largest category. The difference with main tickets, where 'patched' is the largest category, can be understood by the fact that a 'not applicable' sub-ticket can be part of a 'patched' main ticket. We find that 'Not applicable' has a median completion time of 1.4 days (8.6 day average). Compared to the 'Patched' duration, we can see that determining whether patching is needed takes up about one-tenth to one-fifth of the total patch time. Actual patch deployment still makes up the bulk.

4.3. Factors affecting patching time

In the prior subsections, we established that the regulatory pressure to patch a highly selective set of vulnerabilities leads to faster patching. We also explored the dynamics in the various stages of the patching process and their impact on the timeline. This answers our main research question. However, during the interviews that we conducted to interpret the ticketing data, we also asked our interviews about potential factors that might cause patching to go faster or slower. We coded, categorized, and then thematized the answers on time-consuming factors, which revealed ten themes that are described below.

4.3.1. Advisory characteristics. When an advisory comes in, the very first task for the SOC is to read it and find out if it applies to the organization. Whether this is easy depends on the available information on how to ascertain the vulnerability, either in the advisory, from suppliers, or in other sources such as Internet communities.

A recurring example in interviews was the unclear information on how to find Log4J-affected software. As P2 elaborated: "Log4J is a good example here, it was not clear to what software this component belonged, and actually it was not documented". P3 adds: "For example Log4J, it was unclear for a long time who exactly uses Log4J. And then you are very dependent on the supplier who says 'yes, it's in

our system' because you can hardly find it yourself". When we look at Figure 3, we see that the Log4J creation time of sub-tickets varied but took up to 2 weeks, reflecting the struggle of finding affected software and teams.

Patch complexity can affect the speed of patch application as well, for example by how complex patches are more likely to disrupt systems, as P3 states: "and then, of course, it is also how complicated is the patch? Does it affect business operations and continuity?"

4.3.2. Circumstances on patch arrival. P3 described the ideal patch scenario that started with an advisory arriving at business hours, indicating that the time of advisory arrival is a factor in resolution time. We therefore analyzed ticket creation time for advisories arriving within office hours between 9:00 and 17:00 (59.3% of the tickets), which took 3.4 hours on average with a median of 2.4 hours. Tickets outside these office hours (40.7% of tickets) take 4.7 hours on average, with a median of 3.0 hours So we do see a slightly slower process outside office hours.

4.3.3. Clarity of responsibilities and finding owners. A recurring theme in the interviews is how complicated it can be to find out who owns affected software. As P3 stated: "You see the ping-pong game. 'Yes that's your system', 'no that's not my system, it's theirs' and then it goes up and down for weeks and in the end, he says 'Yes I'm responsible for it' and then he gets on with it."

Software ownership boundaries can also become blurred when different departments are involved in the same technical infrastructure. This can complicate the search for affected software and its owner, as in P3's example: "You have a management team that provides virtual machines. But they say they only do the OS. And the application that runs on it belongs to the team that uses the [virtual machine]. If you have [Platform-as-a-service] or [Infrastructure-as-a-service], you also have those responsibilities with the cloud administrator, etcetera. [...] So the VM admin club says we are responsible for up to the OS system. The other team says we are only handling the application. But there is still something in between, who is responsible for that?"

Also, allocation delays can arise because of how IT responsibilities have been set up by specializations. According to P3: "The tricky thing is, traditionally, you had a server, and you put your application on it, and all that stuff is yours. With every cable that goes into it, you name it. So, if there was anything wrong with it, it was you. [...] Nowadays, and that is also a bit with the cloud, that is all more cut up, so you have more specialisms. So one is for the hardware, the other is for the OS, and the other for the application, for the network, you name it. But you're going to have grey areas in between. And those grey areas are a tough one to address in terms of responsibility." Various such 'dependencies' have been discussed elsewhere [27], where here dependencies cannot always be known in advance, or are clear-cut.

In general, we see our ticket data reflect that finding teams and owners is an ambiguous process, as 15.7% of the sub-tickets are being closed as 'taken care of elsewhere', which comes down to allocation issues with closing remarks like "it is already taken care of in sub-ticket X". On average, it takes 8.2 days for these tickets to close as opposed to sub-tickets that ended up being patched (which have an average duration of 20.3 days). It is not just top-down coordination, but also some horizontal 'negotiation' between teams.

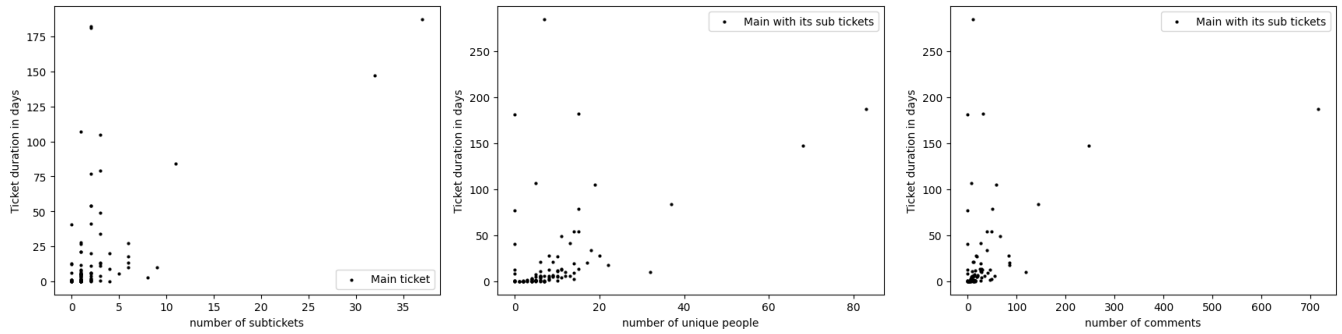
4.3.4. Organization size and coordination. The organization we investigated has over 1000 IT staff and a vast amount of IT components. We have already seen how guiding advisories to their destination can involve many organizational levels such as SOC, solution groups, coordinators, owners, application managers, IT teams, and administrators. It is sheer numbers that affect patch management time, according to P5: "Patch management takes more time with the number of teams, number of applications, number of servers, number of non-standards."

Furthermore, P3 explains how the size factor plays out in communication, as it affects the possibilities of approaching administrators personally: "If you know an administrator well, you can forward it to him, call him immediately, and it will be arranged in no time. But you can't do that with 850 different teams".

Previous research [27] found that coordination efforts are one of the main challenges in the patching process, so we can expect them to affect ticket duration. The number of involved teams amplifies the need for coordination; as P5 stated, "you spend so much more time on coordination if you have more teams". Since a sub-ticket is generated when a new system or system-owning team must be involved in the patching process, we regard the number of teams involved in an advisory as being reflected in the number of sub-tickets. Figure 4a shows how the main ticket duration depends on the number of sub-tickets; we removed one outlier for graphical clarity. This relationship has a moderate Pearson's correlation of 0.46.

As another proxy for coordination efforts, we counted the number of comments on the main tickets together with their sub-tickets. Figure 4c displays how the ticket duration relates to the number of comments. The Pearson's correlation is moderate with 0.45. Not all coordination is reflected in comments and subtickets; we also looked at the number of people directly involved in main tickets together with their sub-tickets. We see in Figure 4b, how this number of unique people involved in an advisory-based ticket correlates with solution time and a moderate Pearson's correlation of 0.48.

4.3.5. Patch impact on business continuity. In many ways, the interviewees mention the balancing between patch urgency and the risk of disrupting running business. Particularly for 'mission-critical systems', downtime is avoided leading to the postponing of patches. For certain applications, there are 'freezes', periods of time in which they are critical to business and should therefore remain untouched. Changing software is risky, but there is also pressure to avoid downtime because of reboots. For less critical systems it might be enough to just explain things well, as P2 said "suppose you implement that in the evening, then at least



(a) Number of sub-tickets. (b) Number of unique people. (c) Comments in the main and sub-tickets.

Figure 4: Main ticket times against proxies for coordination (one outlier removed from Figure 4a for graphical reasons).

you have not actually affected the office users. But that is often really just well-considered, and you can explain that."

4.3.6. Routine and preparedness. In several interviews, participants mentioned the example of Microsoft 'patch Tuesday' as a fixed patching time slot and teams trying to 'get into the Microsoft patch rhythm' (as has been documented elsewhere [2]).

Routine appears to be a way to increase patching speed on several levels. As P1 states: "Sometimes Windows maintenance is outsourced to another party, for which patching is core business; this can make a lot of difference in time and effort." In the same way, being prepared for patching was mentioned, like having a rollback in place and having a test procedure. P7 explained how his team recently has been able to avoided the struggle of recruiting employees to test a patch after an advisory comes in by pre-scheduling test time slots. This compares to the act of patch preparation noted by [14], though here there are questions as to whether this kind of preparation work can only be considered with specific platforms.

4.3.7. Urgency and pro-activeness. As all tickets reflect advisories that are labeled 'high-high' by the national CERT, they have the same initial priority. However, in some cases, a vulnerability is assessed by SOC staff as even more serious, and a 'CERT event' is proclaimed. This comes down to 'priority 1', which is the highest level. P2 illustrates the effect: "Are you going to put 1 ticket with fifty sub tickets underneath? We finally did that. [But only because] we had a CERT event. So, then you get increased management, [...] more resources, more attention". This phenomena, where patching deadlines activate more resources to realise a desired schedule, seems a key mechanism that allows regulatory pressure to result in faster patching. It stands in contrast with other works where sysadmins appear to have the authority to plan patching according to their own design, rather than external pressures and resourcing (e.g., [3], [14]).

For 7% of main tickets, so about once a year from 2015-2022, a priority 1 main ticket was created, which comes with a target time of 4 hours. The priority 1 tickets were closed with a median of 19 hours. In 83.7% of the advisory-based main tickets, a priority 2-level is assigned with a targeted solution time of 48 hours. On average, priority 2 tickets were closed within a median of 5.8 days from the moment the advisory was issued. The remaining 8.7% of the main tickets come with priority 3, which has no 48-hour due time, they have a median solution time of 0.6 days.

Another recurring theme is the pro-activeness of administrators, illustrated by P5: "you often already know about a vulnerability, because it takes time before it reaches the national CERT." Or as P7 states on patching: "Sometimes, before such a major incident has even started, I am already busy". This explains why 3.3% of the main tickets close with a conclusion of 'already patched'. With an average solution time of 2.4 days and a median of 1.1 days, these tickets are closed faster than tickets labeled as 'patched' with an average closing time of 41.3 days and a median of 13.2.

4.3.8. Software and system characteristics. Throughout the interviews, there were many examples of how system or software characteristics affect patching time. P7 explains how 'commercial-off-the-shelf' products generally have better supplier patching support. P8 told a different story, about a system that is very specific to the organization's domain, is less maintained by the supplier, and not ready for an underlying Windows upgrade, thereby blocking security updates. Some products are end-of-life and, although they are still running, are not being patched anymore.

Another time-affecting factor is the degree to which systems are customizable. As P2 explains: "Because software, that's like the old issue between Apple and Microsoft. Look at Microsoft, it's a semi-open ecosystem, and at Apple, it's closed. And that has advantages and disadvantages. So, the disadvantage with Microsoft is that you can customize it. So, if you then perform a certain update, it is never 100% clear what happens."

The national CERT attributes advisories to platforms, meaning a variety of specific OS versions, including network and mobile platforms. 64% of the relevant advisories have no platform information included. The remaining ones are characterized by 103 different platforms, with two large obvious categories: Windows and Linux/Unix-based operating systems. Advisories that affect multiple platforms were

Platform	Number of Main Tickets	Mean (days)	Median (days)	Min. (days)	Max. (days)
Windows only	14 (15.2%)	23.8	4.7	0.1	182.3
Linux/Unix-based only	10 (10.9%)	6.5	1.0	0.0	54.2
Network only	2 (2.2%)	32.5	32.5	11.0	54
Unknown or other	66 (71.7%)	25.6	5.8	0.0	284.5

TABLE 3: Main ticket solution time for platforms.

categorized as 'other'. Table 3 shows how these platforms vary in solution time. We can see that Windows patching takes significantly longer than Linux/Unix patching: 23.8 vs 6.5 days, on average; 4.7 vs 1.0 days at the median.

4.3.9. System expertise. Dealing with high-high advisories involves many roles and different expertise. Early in the process, there is a triage task for SOC that requires their expertise on security and the organization as a whole. In the end, it is application managers or administrators who have technical expertise in software (versions) that enable them to judge if the advisory applies and what the potential impacts are of patching. Ticket solution times depend on how fast the match between those experts is made. P5 explains how effectively assigning the patching issue depends on the technical knowledge of people "because that [speed] also depends on who you speak to in the organization, sometimes it is a super technical manager, and sometimes the application management is outsourced, and you are actually with someone who has more of a product owner role".

As it requires system expertise to conclude if the vulnerability in a ticket actually applies, there are inevitably tickets that end up being 'not applicable' after quite some effort: 32.4% of the sub-tickets (taking 7.1 days on average with a median of 1.9) and 27.5% of main tickets (taking 8.6 days on average with a median of 1.4). So we see that a significant part of the high-high advisory processes is in checking that a vulnerability is actually *not* applicable.

4.3.10. Dependencies and shocks. Previously, P5 described that enterprise environments involve couplings with other systems, which slows down patching speed. In addition to technical couplings, there are many human dependencies that affect patch time, like the number of users, as P2 states on downtime: "... for example, the Citrix environment. In principle, office automation is really important for some people, and that's true. But that's a very small population."

Also, dependencies exist in human resources, such as people being available for testing applications after a patch is applied in a test environment. As P9 states, "[...] the office is, of course, quite large, so he cannot do it alone; he still has a few people, to mobilize them all for testing in a few hours on a Wednesday afternoon, suddenly, that can be quite difficult... that could be a cause."

Dependencies can also be as practical, such as the process of ordering assets. P8 explains how an upgrade required the ordering of dedicated hardware, which took over two months. Another factor is supplier dependence, both in being available to answer queries, as well as in availability to help the organization to install new versions (at times, on-site). A dependency for swift analyses of advisories is on the tools to scan for vulnerabilities or assets like the Log4J jar, as mentioned by P3: "But then we also depend on the community, on the national CERT, on security suppliers, that they have a certain script to scan for it." In sum, interviewees report a variety of dependencies that slow down analyses and the application of a patch.

5. Discussion

Here we reflect on our findings on the patching timelines, the effectiveness of regulatory deadlines for a highly selective group of vulnerabilities, and generalizability.

5.1. Patching under regulatory deadlines

Our study investigated how a large organization responds to high-priority security advisories that are connected with internal and external mandatory deadlines for patching. We found that most advisories do not lead to patching but still take a significant amount of time to resolve. To reach the conclusion that a vulnerability was not applicable to their systems took a median completion time of 1.4 days (average is 8.6 days). This indicates there is a lot of coordination effort in this process to identify whether patching is needed.

Compared to the best available measurement of enterprise patching speed, namely [11], we found that the regulatory pressure on a highly select set of vulnerabilities (about 1% of all published vulnerabilities) does lead to much faster patching. Kotzias et al. [11] showed how industry patching times can vary, which stresses this very need to compare across industries. A breakdown by sector revealed that in "Transportation Infrastructure", it took no less than 23 months to patch 90% of the server-side vulnerabilities. In contrast, we found that it took less than 2 months to complete 90% of all cases (Figure 2). If we only look at the cases where actual patching actions were needed, then it took 5 months for 90% to be completed. This is even twice as fast compared to the total set of enterprises in Kotzias et al., where patching 90% takes nearly 10 months.

The faster patching timelines do suggest that regulatory deadlines influence sysadmin behavior, even though this factor has not been identified by studies of self-reported sysadmin behaviors. This is perhaps because of the recent nature of these regulations, or because of sampling issues (organizations falling under such deadlines may be underrepresented). More conceptually, the interview and survey studies among sysadmins all frame patching as the outcome of sysadmin behavior – e.g., Jenkins et al. [14] refer to a

stage of 'deciding' that seems centred around the sysadmin. This framing is not wrong as such. The sysadmin is usually the final agent initiating the patching action. But it obscures that in some cases, the discretionary power of sysadmins is highly constrained by regulatory pressure. This is not per se negative for those sysadmins. The time pressure brings more organization resources, as we found in our interviews.

As an interesting aside, Kotzias et al. report the longest time scales for industries that operate a combination of IT and Operational Technology (OT), such as Energy Equipment and Services, Gas Utilities, Construction and Engineering, and Marine. Future research might investigate if and how the presence of OT impacts the patching of IT. This might introduce further constraints on the sysadmins that we do not yet understand.

The finding of faster patching becomes a bit more nuanced when we look at the actual compliance rates with the regulatory deadlines. Of the 92 advisory-based main tickets, only 35.9% were solved within the 48-hour deadline specified in organizational policy – and propagated as a best practice by CISA [16] and others. When actual patch deployment is needed, a mere 16.2% is completed within the deadline. On the face of it, this would imply poor compliance with the internal policy. Or, to phrase it with a bit more empathy, the deadline is very difficult to meet when patches need to be tested and rolled-out.

We also compared the patching timelines against the binding one-week deadline of the Dutch BIO policy and the typical deadlines of KEV, namely two or three weeks. Of all tickets that required patch deployment, 32.4% were solved within one week, 56.8% within two weeks, and 62.2% within three weeks. Compliance was even higher when all advisories were considered, so not only those that resulted in patch deployment: 35.9%, 73.9%, and 78.3% for the one, two, and three-week deadlines, respectively.

The somewhat paradoxical conclusion is that the regulatory deadlines speed up patching, even though they are often missed and appear too strict to comply with consistently. The deadlines need to balance realism with exposure. The 48-hour deadline was extremely difficult to meet when actual patch deployment was needed, while the three-week one was feasible for the majority of cases. So the latter is clearly more realistic. Yet it also implies a longer exposure to potential exploitation.

When a new CVE is published, the median time it takes for exploit code to become available is two days [28]. Attackers may start scanning for vulnerable systems within hours after CVEs are published [29]. In that light, three weeks is a long time. That said, patching can be preceded by mitigation measures, like adding firewall rules or isolating systems, which try to buy the organization enough time to roll out the patch. One might argue that, since most enterprises have not been breached with ransomware, attackers cannot exploit these vulnerabilities at scale within two weeks or three weeks. So, even though patching is slower than exploitation, attackers apparently have bottlenecks as well. Thus, for many enterprises, meeting the KEV deadlines might be just fast enough to be ready before it is 'their turn' to be attacked.

Would a more strict deadline than KEV be useful? In fact, various industry sources have pushed the 48-hour deadline as the best practice to adopt. We already pointed to this advice from CISA [16], but it is more widely supported. For example, the Australian intelligence agency (ASD) also recently updated its maturity model to state that for critical vulnerabilities "organisations should patch, update or otherwise mitigate vulnerabilities within 48 hours." [30].

One might argue that a more strict deadline cannot hurt and it might force the organization to try even harder to patch swiftly, even if it cannot make the deadline in the majority of cases. There is a subversive effect, however, of such inadvertent non-compliance. When a deadline is legally imposed, it typically carries with it a sense of obligation and expectation that it must be met. However, if it becomes apparent that achieving the deadline is impossible or highly unreasonable due to factors beyond the control of those tasked with meeting it, the normative force of the deadline diminishes. Worse, it might cause a "normalization of deviance" [31]. If a deadline is consistently not met and nothing happens, as in no breach occurs, then it becomes normal to deviate from the norm. In sum, it matters whether an organization can reasonably meet the deadline. As such, the KEV deadlines strike a better balance between speed and feasibility than the one-week or 48-hour deadlines.

5.2. Generalizability of findings

Like the bulk of all research on enterprise security, our research is case-driven. This reflects the difficulties of getting access to privileged enterprise data. Of course, this always raises the question of how well the findings translate to other organizations. Our partner organization is large, which allows it to operate its own SOC, for example. Differences with smaller organizations are numerous, including that small businesses may not even have an assigned 'IT person' [32]. Therefore, it is obvious that we cannot translate our findings to smaller organizations.

While we have to be careful with generalizing our findings, we do find support in prior and concurrent work for their wider relevance. Our case is located in the transportation sector. Kotzias et al. [11] found patching timelines that are variable per sector, but all in the same order of magnitude. Patching 90% of all server-side vulnerabilities takes 412 days on average in the faster sector and 709 days in the slowest. Transportation is inside that range, on the slower side: 703 days.

More direct evidence for generalizability stems from an industry study by BitSight [5] that was concurrent to ours. The data collection for this report is not public or transparent, but we can compare its findings to ours. BitSight found that organizations remediate KEV vulnerabilites 3.5 times faster than non-KEV ones (174 days versus 621 days). These findings are consistent with ours: there is an observable association between vulnerabilities covered by KEV and patching timelines. In fact, we found that patching of 90% of all vulnerabilities was faster by a factor of 4.6, when

compared to the timelines in transportation (as comparative to our context) reported by Kotzias et al. (5 vs. 23 months). This is remarkably close to the factor of 3.5 found by BitSight.

BitSight does report longer timelines for KEV vulnerabilities than we found. This might reflect the fact that Bitsight calculates the averages across all organizations, also those that do not formally fall under BOD 22-01, BIO or similar regulations. These organizations do not face actual regulatory pressure. Absent that pressure, the average timeline is likely longer. Our case does have regulatory pressure and an even more strict timeline than KEV: the BIO deadline of one week. This would explain why we found faster timelines than BitSight.

5.3. Recommendations

Informed by our results and analysis, we provide the following recommendations:

Develop best practices that are not liabilities. We recommend to reconsider the common "best practice" advice, given by CISA, ASD as well as other governments and experts, to patch critical vulnerabilities within 48 hours. In our partner organization this deadline was mandatory, and not just a recommended "best practice", and yet it seemed impossible to meet it in the majority of cases. Advice and policy requirements that are impossible to meet are at best discouraging, and at worst inviting counterproductive interventions and sanctioning. Various legal developments, such as the EU's NIS2 Directive [7], require organizations to adopt formal patching policies and adhere to them. If the policies are more aspirational than realistic, they turn into liabilities when incidents happen, such that post-incident investigations would find that the organization was systematically not meeting its own patching policies or mandatory regulatory deadlines. In other words, we need to develop best practices that are actual practices, not wishful thinking, so that organizations can base their policies on that guidance and have a fighting chance of living up to them.

We found that a majority of patching actions could successfully be completed within three weeks, the typical deadlines assigned to new vulnerabilities in the CISA KEV catalog. For that deadline, there seems to be a reasonable balance between feasibility and urgency. It is ironic that CISA and others recommend patching all critical vulnerabilities within 48 hours, yet the same CISA attaches a three-week deadline to the subset of the most critical vulnerabilities, namely those that are already being exploited in the wild – i.e., new KEV catalog entries. This contradiction suggests that the guidance to patch in 48 hours is not usable advice and should not be held up as a best practice.

Manage for non-compliance with patching policy. We noted the potential for a "normalization of deviance" [31]. We would recommend to record not only when vulnerabilities are patched in time, but to add additional detail for why other vulnerabilities are not patched within policy-defined deadlines, or result in an exception. We note this given that only 35.9% of high-high tickets were resolved in the partner organization within the 48-hour deadline. If there are infrastructure reasons or repeated causes for exceptions and delays, it would be the sum of such tickets which acts as evidence for the need for wider investment, to, e.g., replace elements of the digital estate which are routinely difficult to patch.

One patching policy, or multiple policies for differentiated system types? Although a minority of tickets involved suppliers, it is important to consider how outsourced or niche IT services and software are patched, alongside typical enterprise IT such as Microsoft platforms. It may be necessary to have realistic policies for homogeneous services (such as enterprise systems provided by Microsoft, SAP, and the like), but to account in policy for the differences seen in other, cloud-based or niche services; in effect, consider multi-tier patching deadlines. Without such a consideration, organizations may gravitate toward homogeneity if only to make maintenance (and patching) easier toward demonstrating compliance. Conformity to one accepted type of system would risk putting more power in the hands of fewer suppliers, as has already been observed in, e.g., the higher education sector [33].

6. Conclusion

For the organization studied here, we found that 40.2% of advisories required patching action, with a median completion time of 13.2 days; advisories that do not end in requiring a patch have a median of 1.4 days, pointing to the importance of coordination and finding potentially vulnerable systems. Completing the patching process under the mandatory deadline of 48 hours is achieved in 16.2% of all cases. For the deadline of one week, under the Dutch BIO regulation, patching is achieved in 32.4% of the cases, while the performance against the typical CISA KEV deadlines is a bit more hopeful: 56.8% is patched in two weeks and 62.2% in three weeks. All in all, we do find that regulatory deadlines for a highly-selective set of vulnerabilities is associated with much faster patching times than previously reported.

Acknowledgements

This work would not have been possible without the generous and courageous collaboration of our partner organization in the Dutch Rijksoverheid, for its willingness to open up its internal processes to outsider scrutiny. We also want to express our gratitude for the great help we received from Jeroen Gaiser and Marten Mooibroek, who were supportive and committed throughout the study's twists and turns. We thank Jonathan Spring for comments on an earlier version of the paper. This research was funded by the Dutch Research Council (NWO) via project THESEUS (grant nr. NWA.1215.18.006).

References

[1] D. Goodin, "Teens with "digital bazookas" are winning the ransomware war, researcher laments," Ars Technica, Nov. 2023. [Online]. Available: https://arstechnica.com/security/2023/11/teens-with-digital-bazookas-are-winning-the-ransomware-war-researcher-laments/

[2] A. Jenkins, P. Kalligeros, K. Vaniea, and M. K. Wolters, ""Anyone Else Seeing this Error?": Community, System Administrators, and Patch Information," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. Genoa, Italy: IEEE, Sep. 2020, pp. 105–119.

[3] N. Dissanayake, A. Jayatilaka, M. Zahedi, and M. A. Babar, "Software security patch management - A systematic literature review of challenges, approaches, tools and practices," *Information and Software Technology*, vol. 144, p. 106771, Apr. 2022.

[4] J. C. West and T. Moore, "Longitudinal Study of Internet-Facing OpenSSH Update Patterns," in *Passive and Active Measurement*, O. Hohlfeld, G. Moura, and C. Pelsser, Eds. Cham: Springer International Publishing, 2022, vol. 13210, pp. 675–689.

[5] B. Edwards, "A global view of the cisa kev catalog: Prevalence and remediation," Bitsight Technologies, Inc., 2024. [Online]. Available: https://www.bitsight.com/sites/default/files/2024-04/bitsight-a-global-view-of-cisa-kev-catalog.pdf

[6] E. Livne, "Why organizations struggle with patch management (and what to do about it)," Qualys Community Blog, 2022. [Online]. Available: https://blog.qualys.com/qualys-insights/2022/09/20/why-organizations-struggle-with-patch-management-and-what-to-do-about-it

[7] European Commission, "Directive on measures for a high common level of cybersecurity across the union (nis2 directive)," European Commission, 2023. [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/nis2-directive

[8] Cybersecurity & Infrastructure Security Agency (CISA), "BOD 22-01: Reducing the Significant Risk of Known Exploited Vulnerabilities," https://www.cisa.gov/news-events/directives/bod-22-01-reducing-significant-risk-known-exploited-vulnerabilities, Nov. 2021.

[9] Baseline Informatiebeveiliging Overheid (BIO), "Baseline Information Security Government (BIO)," https://www.bio-overheid.nl/.

[10] Cybersecurity & Infrastructure Security Agency (CISA), "CISA Known Exploited Vulnerabilities Catalog." [Online]. Available: https://www.cisa.gov/known-exploited-vulnerabilities-catalog

[11] P. Kotzias, L. Bilge, P.-A. Vervier, and J. Caballero, "Mind your own business: A longitudinal study of threats and vulnerabilities in enterprises." in *Network and Distributed System Security Symposium (NDSS) 2019*, 2019.

[12] Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey, and J. A. Halderman, "The Matter of Heartbleed," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. Vancouver BC Canada: ACM, Nov. 2014, pp. 475–488.

[13] B. Genge and C. Enăchescu, "ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services," *Security and Communication Networks*, vol. 9, no. 15, pp. 2696–2714, Oct. 2016.

[14] A. D. Jenkins, L. Liu, M. K. Wolters, and K. Vaniea, "Not as easy as just update: Survey of system administrators and patching behaviours," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–17.

[15] F. L. Marshini Chetty, "Keepers of the Machines: Examining How System Administrators Manage Software Updates," in *Keepers of the Machines: Examining How System Administrators Manage Software Updates*, Aug. 2019.

[16] Cybersecurity & Infrastructure Security Agency (CISA), "StopRansomware: AvosLocker Ransomware," https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-284a, Oct. 2023.

[17] T. Sasaki, A. Fujita, C. H. Gañán, M. Van Eeten, K. Yoshioka, and T. Matsumoto, "Exposed Infrastructures: Discovery, Attacks and Remediation of Insecure ICS Remote Management Devices," in *2022 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2022, pp. 2379–2396.

[18] C. Bennett, A. Abdou, and P. C. Van Oorschot, "Empirical Scanning Analysis of Censys and Shodan," in *Proceedings 2021 Workshop on Measurements, Attacks, and Defenses for the Web*. Virtual: Internet Society, 2021.

[19] C. T. Katharina Krombholz, "Security, Availability, and Multiple Information Sources: Exploring Update Behavior of System Administrators," in *SOUPS'20: Sixteenth USENIX Conference on Usable Privacy and SecurityAugust 10 - 11, 2020*. USENIX Association2560 Ninth St. Suite 215 Berkeley, CA, United States, 2020.

[20] S. de Smale, R. van Dijk, X. Bouwman, J. van der Ham, and M. van Eeten, "No one drinks from the firehose: How organizations filter and prioritize vulnerability information," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023.

[21] N. Dissanayake, M. Zahedi, A. Jayatilaka, and M. A. Babar, "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*, vol. 6, no. CSCW2, pp. 1–29, Nov. 2022.

[22] National Cyber Security Center, "National Cyber Security Center," https://www.ncsc.nl/.

[23] V. Braun and V. Clarke, "One size fits all? what counts as quality practice in (reflexive) thematic analysis?" *Qualitative research in psychology*, vol. 18, no. 3, pp. 328–352, 2021.

[24] ——, "Using thematic analysis in psychology," *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[25] S. C. Sundaramurthy, J. McHugh, X. Ou, M. Wesch, A. G. Bardas, and S. R. Rajagopalan, "Turning contradictions into innovations or: How we learned to stop whining and improve security operations," in *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, 2016, pp. 237–251.

[26] Microsoft Security Team, "Understanding the performance impact of Spectre and Meltdown mitigations on Windows Systems," Microsoft Security (blog). [Online]. Available: https://www.microsoft.com/en-us/security/blog/2018/01/09/understanding-the-performance-impact-of-spectre-and-meltdown-mitigations-on-windows-systems/

[27] N. Dissanayake, M. Zahedi, A. Jayatilaka, and M. A. Babar, "A grounded theory of the role of coordination in software security patch management," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. Athens Greece: ACM, Aug. 2021, pp. 793–805.

[28] A. D. Householder, J. Chrabaszcz, T. Novelly, D. Warren, and J. M. Spring, "Historical analysis of exploit availability timelines," in *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*, 2020.

[29] K. Metrick, J. Semrau, and S. Sadayappan, "Think fast: Time between disclosure, patch release and vulnerability exploitation — intelligence for vulnerability management, part two," Mandiant blog. [Online]. Available: https://www.mandiant.com/resources/blog/time-between-disclosure-patch-release-and-vulnerability-exploitation

[30] "Essential eight maturity model changes," Australian Signals Directorate, 2023. [Online]. Available: https://www.cyber.gov.au/resources-business-and-government/essential-cyber-security/essential-eight/essential-eight-maturity-model-changes

[31] D. Vaughan, *The Challenger launch decision: Risky technology, culture, and deviance at NASA*. University of Chicago press, 1996.

[32] S. Parkin, A. Fielder, and A. Ashby, "Pragmatic security: modelling it security management responsibilities for sme archetypes," in *Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats*, 2016, pp. 69–80.

[33] T. Fiebig, S. Gürses, C. H. Gañán, E. Kotkamp, F. Kuipers, M. Lindorfer, M. Prisse, and T. Sari, "Heads in the clouds: Measuring the implications of universities migrating to public clouds," in *Proceedings on Privacy Enhancing Technologies 2023*, 2023.

Appendix A. Interview questions

Interview example questions, round 1

Introduction

- Have you been working here for a long time, and what is your role now?
- What is your role in security patching?
- How much of your work involves security patching?
- We know from tickets that national CERT high-high advisories lead to a stream of activities. Besides these, can you describe the other patching activities at this organization?
- How are these other patching activities different from national CERT high-high advisory initiated ones?

Incoming national CERT advisory

- Can you tell me what a national CERT advisory is?
- How are you informed about national CERT advisories?
- Can you guide me through the process if a high-high national CERT advisory comes in?
  probe: Can you tell me about the prioritization of advisories and what that means for 'due dates'?
  probe: What people are involved?
  probe: Are the same people involved all the time?
  probe: What activities take time?
- How do you conclude if an advisory applies to the organization or not?

Ticketing

- What role does the ticketing system play in handling advisories?
- What other communication runs alongside the ticketing system?
- Can you say when communication is difficult, maybe by example?
- Can you say when communication takes time?

Priority

- Can you tell me how the priority of a ticket is established?
- Is that priority easy to determine? Can you maybe give examples of when this is easy or difficult?
- Do you discuss that with others?

Allocation

- Can you tell me how it becomes clear to whom a ticket should be allocated?
  probe: when allocating a ticket based on a high-high advisory, how sure are you that the ticket shall lead to patching?
- What things make the assignment take time?
  probe: Can you maybe give examples of when this took time?
  probe: Do you discuss assigning with others?

Closing of a ticket

- How is decided that a ticket can be closed?
- Do you check whether patching has actually taken place? If so, how?
  probe: Can you give an example of an easy and difficult situation?

Are there any other tasks or issues that come to mind regarding national CERT high-highs? And if so, what in these activities takes time?

Interview example questions, round 2

Introduction

- Have you been working here for a long time?
- What is your role in security patching, and how much of your work involves security patching?

In general, on incoming high high National CERT advisory tickets

- Can you guide me through the process if a high-high national CERT advisory ticket is assigned to you?
  probe: What people are involved?
  probe: What activities are involved?
- Can you give an example of when a ticket is fast and when it is slow?
- Can you tell me what factors determine how much time it takes to handle and close a ticket? [Then summarize each factor that the respondent mentions, and ask if there are others until they cannot think of any other factor]
- How do you come to the conclusion that a patch is not applicable, or whether the vulnerability needs to be mitigated?
- How is decided that a ticket can be closed?

Specific cases of high high National CERT advisory tickets

- Discuss three specific tickets
- Do you remember this ticket?
- Can you describe the activities in this ticket?
- Can you describe how much time each of these activities took?
- Can you describe why this ticket was fast/slow? (depending on example)

Appendix B.
Interview codebook

Code group	Code
Advisory characteristics	Vulnerability documentation
	Time of day of advisory arrival
	Patch complexity increasing outage risk
Ownership / responsibility	Ease of determining problem owner
	Unclear system demarcation
	Grey areas due to separation of responsibilities
	Hard to determine owners of old systems
Routine and familiarity	Familiarity with software and owner
	Addressing quickly based on a similar ticket
	Routine of patching
Prepared for patching	Available rollback
	Available test environment
System Expertise	Being specialized in the affected software
	Very technical people can act faster
Proactiveness	Proactive administrators
	Pre-scheduled patching
Dependencies	Insight in dependencies
	User base
	Patch requiring ordering of hardware
	Depending on others in shared infrastructure
	Speed of supplier response
	Supplier availability
Patch impact on business continuity	Need for high availability
	Avoiding reboots
	Avoiding risk of application outage
	Freezes requiring patch postponing
	Mission-critical system
	Estimating patch impact
	System importance
Accountability	Being able to explain the need for downtime
Help / shared ad hoc tooling	Available vulnerability-specific scanning tools

TABLE 4: Interview codebook (1/2) – the 'Code group' column represents clusters of codes, forming themes. Each Code group then consists of multiple individual codes, which were created based on the coding of interview transcripts.

Code group	Code
Organizational size	Organizational size preventing personal approach
	Approaching administrators personally
	Organizational size taking extra time for coordination
	Patch management takes more time with the number of teams
	Patch management takes more time with the number of applications
	Patch management takes more time with the number of servers
	Patch management takes more time with the number of non-standards
	Patch management takes more time with complexity
Software/system characteristics	Software customizability causing patch side effects
	Tight coupling passes on patch consequences
	Patch that spawns an upgrade cascade
	Software not ready for platform upgrade
	Skip patching of end-of-life products
	Common off-the-shelf products have better supplier contact
Testing time	Testing cost
	Decision to spend time on testing
	Pre-scheduled testing windows
	Mobilizing test team
Responding to advisory arrival	Time of day of downtime
	Being busy when advisory arrives
	[National CERT] taking time
	[National CERT] email notification hinder automation
Priority setting	CERT for more resources
	Mitigate first and patch later

TABLE 5: Interview codebook (2/2).

Appendix C.
Meta-Review

The following meta-review was prepared by the program committee for the 2025 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

C.1. Summary

This paper investigates the effectiveness of regulatory deadlines on the patching process for one large Dutch organization. The authors analyze the organization's ticketing system for tracking patching, and conduct semi-structured interviews with organization members involved in the patching process.

C.2. Scientific Contributions

- Independent Confirmation of Important Results with Limited Prior Research
- Provides a Valuable Step Forward in an Established Field

C.3. Reasons for Acceptance

1) This paper provides independent confirmation of important results with limited prior research. There are few prior large-scale studies of security patching behavior. This paper provides a longitudinal evaluation of an organization's patching practices, combined with a qualitative analysis of those involved in the patching process. The authors then compare their insights with those from the limited prior research.
2) This paper provides a valuable step forward in an established field. Security patching is a critical process that has been previously studied. However, this paper provides a novel empirical evaluation of the effectiveness of patching regulations in a real-world setting, using a unique data source from a large organization, shedding new light on how regulation influences organizational patching behavior.

C.4. Noteworthy Concerns

1) Given the paper's focus on one Dutch organization and its adherence to a Dutch security regulation, it is unclear the extent to which the findings will generalize to other organizations or regulations in other contexts.
2) The study compares its finding to a prior large-scale patching study. While synthesizing a study's new insights/findings is valuable, the prior study considered a very different patching context. Thus, some of the takeaways from the comparison do not seem appropriate, given the context differences.