Keepers of the Machines: Examining How System Administrators Manage Software Updates

Frank Li
University of California, Berkeley
frankli@cs.berkeley.edu

Lisa Rogers
University of Maryland
lmrogers@umd.edu

Arunesh Mathur
Princeton University
amathur@cs.princeton.edu

Nathan Malkin
University of California, Berkeley
nmalkin@cs.berkeley.edu

Marshini Chetty
Princeton University
marshini@princeton.edu

ABSTRACT

Keeping machines updated is crucial for maintaining system security. While recent studies have investigated the software updating practices of end users, system administrators have received less attention. Yet, system administrators manage numerous machines for their organizations, and security lapses at these hosts can lead to damaging attacks. To improve security at scale, we therefore also need to understand how this specific population behaves and how to help administrators keep machines up-to-date.

In this paper, we study how system administrators manage software updates. We surveyed 102 administrators and interviewed 17 in-depth to understand their processes and how their methods impact updating effectiveness. We find that system administrators proceed through software updates through five main stages that, while similar to those of end users, involve significantly different considerations and actions performed, highlighting the value of focusing specifically on the administrator population. By gathering evidence on how administrators conduct updates, we identify challenges that they encountered and limitations of existing procedures at all stages of the updating process. We observe issues with comprehensively acquiring meaningful information about available updates, effectively testing and deploying updates in a timely manner, recovering from update-induced problems, and interacting with organizational and management influences. Moving forward, we propose directions for future research and community actions that may help system administrators perform updates more effectively.

1. INTRODUCTION

System administrators serve as "keepers of the machines," entrusted by organizations to oversee their computers, many of which are vital to an organization's operations. Their duties include regularly applying software updates in a timely manner to ensure organizational safety against crippling attacks. Failure to patch known vulnerabilities can lead to devastating consequences [13] such as the colossal 2017 Equifax data breach which exposed sensitive personal data on over *140 million* individuals [38].

USENIX Symposium on Usable Privacy and Security (SOUPS) 2019.
August 11–13, 2019, Santa Clara, CA, USA.

While prior studies have investigated how end users deal with software updates [18, 19, 22, 30–32, 35, 40, 45, 46, 49, 50], there has been less attention on system administrators, whose technical sophistication and unique responsibilities distinguish them from end users. Industry reports and guides on administrator patching exist (e.g., Sysadmin 101 [41]), but these lack peer-review and transparent rigorous methods. Prior academic work on system administrators is often dated and focuses on aspects of administrator operations other than updating (e.g., on general tools used [11]) or specific technical (rather than user) updating aspects. Given the critical role that system administrators play in protecting an organization's machines, it behooves us to better understand how they manage updates and identify avenues for improved update processes. We therefore set out to answer two primary research questions: (1) what processes do system administrators follow for managing updates, and (2) how do administrator actions impact how effectively they perform system updates. To answer these questions, we surveyed 102 administrators and conducted semi-structured interviews with 17 of them.

Our study determined that system administrators proceed through software updates through five main stages: (1) learning about updates from information sources, (2) deciding to update based on update characteristics, (3) preparing for update installation, (4) deploying an update, and finally (5) handling post-deployment update issues that may arise. By analyzing the factors that system administrators considered and the actions that they performed, we identified challenges that they encountered and limitations of existing procedures at each stage of the updating process. We observed problems with comprehensively obtaining relevant information about available updates, effectively testing and deploying updates in a timely fashion, and recovering from update-induced errors. We also witnessed how organizational and management influence through policies and decisions can impact the administrator's ability to handle updates effectively at multiple stages, sometimes for better, sometimes for worse. In addition, we note that while high-level aspects of software update workflows for system administrators mirror those of end users [31, 46], we found that the particular factors considered and the actions taken by system administrators are significantly different across all stages of the update process. This difference highlights the value of specifically studying the administrator population.

Our evidence-based study extends the research literature on updating practices to system administrators, a unique population. In particular, our work makes two primary contributions: first, we provide empirical grounding on how administrators update multiple machines for their organizations, examining the consequences of their actions at depths beyond prior explorations [14]. This evidence includes

insights into how their actions impact how effectively they perform software updates to better secure their systems. Second, we make grounded recommendations for improving administrator update processes through better systems for managing updates, better designed updates, and a shift in organizational policies.

2. BACKGROUND AND RELATED WORK

In this section, we highlight existing studies related to our research and place our work in context.

2.1 End Users and Software Updates

Numerous works [18, 19, 22, 30–32, 35, 40, 45, 46, 49, 50] have examined end user perceptions, attitudes, and behavior towards applying software updates. Ion *et al.* [22] and Wash *et al.* [49] found that non-expert computer users failed to recognize the security benefits of updates and frequently avoided installing them. Other studies measured the time users took to apply updates and discovered that reaching half of all vulnerable desktop [35] and mobile applications [40] took nearly 45 days and 1 week, respectively. One set of studies has examined why users avoid or fail to install software updates, discovering a variety of factors related to costs, necessity, and risks [32]. Example factors include that updates cause unexpected changes to user interfaces [19, 31, 45, 46], that updates take a long time to install [31, 46], that updates raise privacy concerns [18], and that updates cause unnecessary restarts of applications [19, 31, 45, 46].

Given that automatic updates are more effective in keeping end user systems updated than manual updates [16, 20, 35], another set of studies has examined user attitudes towards and experiences with automatic updates [30, 50]. Rader and Wash [50] identified that automatic updates with only partial user involvement (e.g., during restarts) often led to poor mental models of updating and consequently resulted in less secure systems. More recently, Mathur and Chetty [30] found that negative experiences with automatic updating resulted in users disabling auto-updates on Android devices.

While these studies have shed light on how end users deal with software updates, their findings do not necessarily generalize to system administrators, who are more technically sophisticated and operate with expanded responsibilities.

2.2 Administrators and Software Updates

Several studies [12, 23–25, 47, 48] have examined the workflows and needs of administrators to enable better security practices but did not focus on software updating processes specifically. Kraemer and Carayon [24] conducted interviews with 16 network administrators and security workers, identifying that organizational structures and policies played an important role in how they handled security. Kandogan *et al.* [23] discussed various stories from IT administrators about their experiences. Krombholz *et al.* [25] investigated usability problems encountered by website administrators trying to securely deploy HTTPS. Chiasson *et al.* [12] devised usability and interface design principles to help system administrators better diagnose security issues. Velasquez and Weisband [47] conducted interviews with administrators and designed a model to understand their beliefs and attitudes. In this model, the authors identified that both informational factors (e.g., quality) and system factors (e.g., ease of use) informed these beliefs and attitudes. In a follow-up study [48], the same authors found that administrators largely acquired their knowledge through practice rather than education and certification. They recommended that software developers should design tools with administrator technical sophistication in mind.

Closely related to our own work is the preliminary study conducted over a decade ago by Crameri *et al.* [14]. Although not the primary focus of their work, these researchers conducted brief surveys of 50 system administrators to learn about their updating practices. They found that nearly 70% of administrators refrained from installing software updates and that administrators tested updates on a smaller set of machines before patching their production systems. The study investigated certain aspects of administrator behavior to inform the design of their update testing system, but did not perform a comprehensive and rigorous exploration of update management. More recently, Dietrich *et al.* [15] looked at how system administrator operations could result in security misconfigurations, finding that missing and delaying software updates are among the most commonly reported security misconfigurations.

Unlike these previous studies, our work provides an in-depth investigation of system administrator practices for updating the machines they manage. Using a combination of surveys and interviews, we examine a larger sample of administrators than Crameri *et al.* [14] and provide more recent and in-depth insights into their complete update management process.

3. METHOD

To investigate how system administrators manage updates at scale, we conducted a qualitative study of current administrators responsible for managing updates in their organizations. Our study proceeded in two phases. In phase one, we administered a large-scale survey of administrator updating practices, whose design was informed by pilot interviews. In phase two, we conducted semi-structured interviews with administrators. We specifically sought participants who had been working at an organization with five employees or more for a period of at least one year, to ensure they had job familiarity. We restricted participation to those over 18 years old residing in the United States (US). Both study phases received approval from the Institutional Review Boards (IRBs) of our universities. Our survey and interview questions are listed in the Appendix.

3.1 Preliminary Phase: Pilot Interviews

In Fall 2015-Spring 2016, to inform the design of our large-scale study, we recruited seven system administrators to participate in semi-structured pilot interviews about software updates. The interview questions were developed based on prior studies on software updating [31, 46] and previous knowledge about the software update development and management process (see Appendix A for details). We recruited participants via institutional mailing lists and social media, filtering for those who explicitly dealt with software updates. All interviews were conducted over the phone via Skype and recorded. The interviews lasted between 30-50 minutes. Participants were also asked to fill out a background survey that contained general questions about demographics, the type of software or programming languages used, the types of updates they handled, and any positive and negative aspects of their job responsibilities. Participants were compensated with $20 gift cards and a chance to win a hard drive.

Demographics: All seven participants were male and lived in the US. They were predominantly 20–40 years of age and only one participant did not have a bachelor's degree. The majority of participants had 1–10 years of work experience as a system administrator.

Analysis: We transcribed all pilot interviews and three coders used inductive thematic analysis [42] to derive the following over-arching themes in administrator update management: finding information about available software updates, testing and preparing for updates, deploying updates, and monitoring for update-triggered issues post-deployment. We used these themes to design questions for our study's two phases.

3.2 Phase One: Survey

Based on the pilot interviews, we constructed a survey asking about a participant's organization and responsibilities (e.g., size of organization, number of machines managed), how they manage the security of their systems, how they handle each stage of the update management process, and what works well and poorly for them (see Appendix B for details). The survey consisted of 41 questions and took approximately 15 minutes to complete. We recruited system administrators in September and October of 2017 using social media, blogs associated with our research labs, and Reddit [7]. In addition, we recruited administrators attending the 2017 Large Installation System Administration Conference (LISA) by distributing fliers about our survey and providing a computer at the venue where administrators could complete the survey. As an incentive, we entered administrators who participated into a drawing for a Samsung S8 phone. In total, 102 system administrators completed the entire survey. We note that we recruited 22/102 survey participants at the LISA conference and the rest from online.

Data Analysis Method: The survey consisted of multiple-choice and open-ended questions. We focused our analysis on questions pertaining to software updating, as our survey also contained several less relevant questions on other security practices. We analyzed open-ended questions using open coding, identifying themes in the question responses [51]. Two researchers independently developed a set of codes across all questions and met to converge on a final codebook. Then, each researcher independently coded all question responses using that codebook. We had 199 codes with 611 coded segments in total, discussing themes of interest such as "Testing", "Update Issues", "Addressing Update Issues", "What Works Well", and "What is Challenging". We use Kupper-Hafner inter-rater agreement scores [26] to quantify the consistency of the coding, finding an average agreement of 0.83, indicative of largely consistent coding. The survey coders met and converged upon the final codes for all open-ended question responses.

3.3 Phase Two: Semi-Structured Interviews

Using the themes identified by our pilot interviews, we developed a guide for conducting semi-structured interviews with system administrators. The guide contained questions about a participant's demographics and job, and their update management process (see Appendix C for details). Throughout Fall 2017, we recruited 17 interview subjects through the same channels as with the survey. All but one of our subjects participated in the survey as well. Interviews ranged from 1 to 3 hours long, were conducted in person or over Skype, and were recorded. We compensated participants with a $20 Amazon gift card.

Data Analysis Method: Using transcriptions of the recorded interviews, we developed a codebook for the responses through regular peer review meetings, based on the themes of interest for the interviews such as "Job Responsibilities", "Update Importance", and the various update stages, including "Seeking Update Information", "Deployment", "Testing", and "Update Issues". The codes were initially created by one team member and refined by group discussions and consensus [51]. Two coders independently coded the interview responses using the resulting codebook using inductive thematic analysis [42]. We had 347 codes with 1447 coded segments in total. Calculating inter-rater reliability for such qualitative coding of non-survey data has been shown to be difficult because of the nature of assigning multiple codes to data and inherent biases of coders [10]. For completeness, however, we randomly sampled 6/17 transcripts and computed an average agreement percentage between the two independent coders of 0.77, indicating high consistency. We discussed points of disagreement and ensured that the resulting themes discussed in the paper were in line with both team members' interpretations of the data.

3.4 Participant Demographics

Here we present the demographics of the 102 survey respondents and 17 interview subjects.

3.4.1 Respondent Characteristics

The population was male-dominated; only 6/102 survey and 2/17 interview subjects were female. The most common age bracket was 26-35 years old, containing 43/102 survey participants and 8/17 interview subjects. Other common age brackets were 36-45 years old (24 survey and 4 interview participants), and 46-55 years old (14 survey and 2 interview subjects). Most administrators had some higher education; 57/102 survey and 10/17 interview participants had a bachelor's degree while 37 survey and 5 interview participants had some college education but no degree. Salaries varied widely, evenly distributed primarily between $35,000 to $150,000 (accounting for 93/102 survey and 14/17 interview participants). Survey respondents had a median of 11 years of experience, ranging from 1 to 35 years. In contrast, interview subjects had a lower median experience of 6 years, although the range was similar (1-34 years).

3.4.2 Organization Characteristics

About half of our study participants (56/102 in the surveys and 8/17 from the interviews) worked at larger organizations with over 500 employees. In comparison, only 13/102 survey and 2/17 interview participants worked for small organizations with fewer than 50 employees. In total, 22 survey respondents did not indicate the number of hosts they managed (all interview subjects did provide a response). However, the remaining typically oversaw many machines: only 12/102 survey and 3/17 interview participants maintained fewer than 100 hosts, while 36 survey and 8 interview subjects indicated they administered between 100-499 machines and 22 survey and 5 interview participants said they handled over 1000 machines. Servers were the most common type of machine managed, handled by 96/102 survey respondents. Over half of the administrators also dealt with desktops (63), routers (60), and laptops (57). Our participants maintained primarily Linux (73) and Windows (71) machines, and less so Macs (44).

3.5 Limitations

Studying system administrators is challenging as they are a specialized population that is difficult to recruit compared to end users. Thus, our study's approach may have limitations.

1. As administrators are often paid well, our study's participation compensation may not have influenced their decisions to contribute. Instead, those more ideologically motivated may have donated their time.
2. Due to our recruitment method, our study participants may not be representative of system administrators in general. For example, we only studied individuals from the US, so our findings may not apply globally. Similarly, we only recruited administrators fully employed by an organization, which does not capture those working part-time or as contractors.
3. Our results reflect our study's sample, which skewed towards certain demographics (e.g., males). Similarly, we recruited many of our participants via Reddit and the LISA conference. These subpopulations may exhibit certain skewed characteristics. For example, those attending the LISA conference may operate with a larger budget (covering conference expenses).
4. Our surveys and interviews contained open-ended questions.

During our analysis, we provide the number of study subjects who gave a particular response to these open-ended questions (and indicate when results are obtained from such questions). However, we caution that such counts are not necessarily reliable indicators of real-world prevalence. In particular, we cannot assume a respondent does not act a certain way just because they do not mention such behavior, as they may have simply focused on alternative discussion topics.

5. Our study is an exploratory one that focuses on the processes system administrators use to manage software updates. However, we did not investigate all updating aspects in depth. For example, we did not explicitly solicit recommendations from our study participants on how to improve updating tools and methods, nor did we tease apart the differences in updating between different types of organizations or machines. Moving forward, our study can help inform the design of broader quantitative explorations of these updating dimensions at scale.

4. OVERVIEW OF FINDINGS

From the responses to our system administrator surveys and interviews, we determined that administrator software update workflows consisted of five primary stages. These five stages, as illustrated in Figure 1, are: (1) learning about updates from information sources, (2) deciding to update based on update characteristics, (3) preparing for update installation, (4) deploying the update, and finally (5) handling post-deployment update issues that may arise. For each stage, our analysis determined the factors that system administrators considered and the actions that they conducted (also listed in Figure 1). This data affords us insights into the challenges that administrators encountered when updating and limitations of existing procedures. In Section 11, we discuss recommendations for improving administrator update processes grounded in our findings. We also compare how update workflows differ for system administrators versus end users in Section 11.1, identifying significant differences.

In the following sections on update stages, we explore how system administrators proceed through each stage and the security implications of their behaviors. Throughout the results, we designate quotes from survey respondents with *S* and interview participants with *P*.

5. STAGE 1: LEARNING ABOUT UPDATES

In both our surveys and interviews, participants reported that—before deploying software updates—they first had to learn about available updates and then make decisions about which updates to handle. We note that while automatically initiated updates circumvent the need to find and digest information, many of our study participants did not find them universally suitable. Thus, for our participants, it was still important to process update information efficiently.

5.1 Update Processes

We asked our study participants about how they discovered the updates they applied. In our survey, we asked a closed-ended question with 11 possible options and a free-form response (as shown in Table 1), while our interview question was open-ended. In total, 99/102 survey participants and all 17 interview subjects responded. The types of information sources discussed by interview subjects overlapped with our survey question options, but we note that the distributions among survey and interview participants differed, likely due to the open-ended nature of the interview question.

As shown in Table 1, our participants relied on various types of information sources. Most survey respondents reported a median of 5.0 different types of sources, and a quarter reported using seven or more types. (We do not report the same counts for interview data given that open-ended responses are not necessarily comprehensive nor indicative of prevalence, as discussed in Section 3.5.) This large quantity of source types suggests that update information is highly dispersed, requiring administrators to diligently peruse a variety of outlets to stay informed on available updates. Some interview participants described sourcing information in this manner as non-ideal, as typified by P5's discussion on discovering updates that patched newly identified vulnerabilities: *"There's not always a canonical*

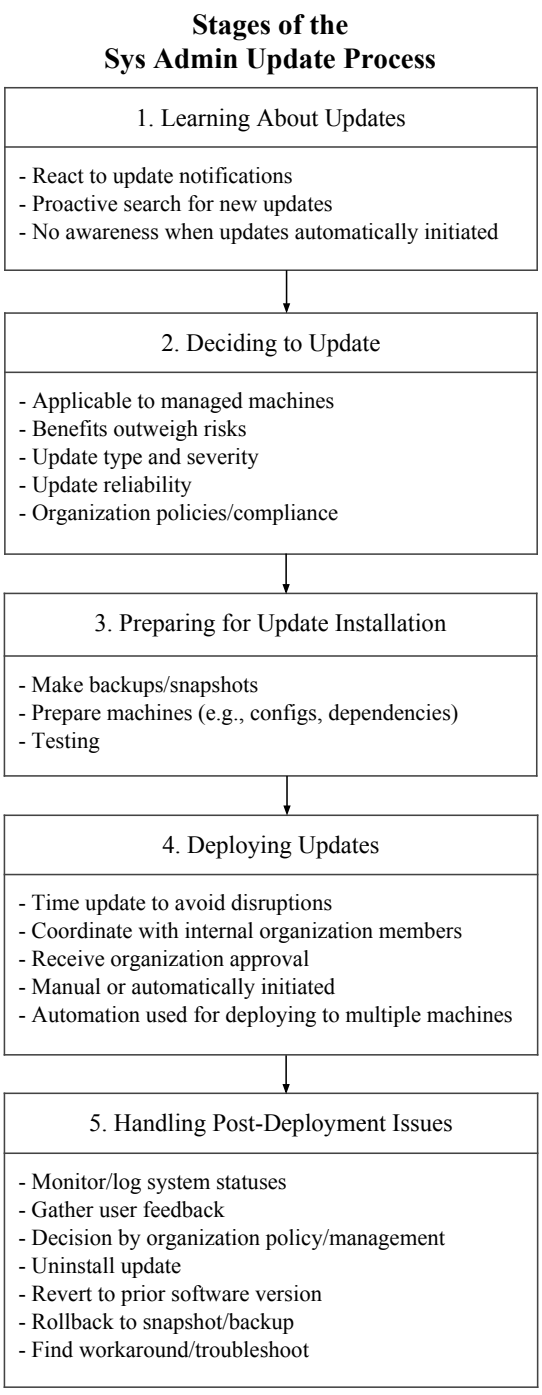

Figure 1: Our study identified five primary stages of the update process for system administrators. We list the salient considerations for each stage.

Table 1: Sources used for discovering available updates.

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

place to go for a web advisory. When these vulnerabilities get found on the Internet, they might affect you, it could be announced on the Apache web server mailing list, it could be on the Ubuntu server list, it could be a topic on Server Fault. There's a lot of places." Also, not all sources were ideal. For example, P13 stated that *"sometimes if there's a really critical vulnerability, email's not the most real-time method of getting things going."*

5.2 Impact on Updating Effectiveness

Our study participants revealed that they each relied on a diverse set of methods for retrieving update information from multiple sources. Due to the lack of a centralized source of information, we note that it is possible that some system administrators may lack the full coverage of relevant information if they miss an important source. We also observed that administrators used some sources that require active retrieval and digestion, such as news articles, blog posts, forums, and social media. These sources may require more time and effort, compared to sources that push information directly to the administrators, such as direct vendor notifications or mailing lists. Our study ultimately does not concretely reveal how comprehensive or effective administrators are at update information retrieval, but suggests that this is a nontrivial task for many.

6. STAGE 2: DECIDING TO UPDATE

For the second stage of their updating process, administrators in our study filtered update information to decide if they should deploy an update. This was a nontrivial task because of the profusion of update information from a variety of sources.

6.1 Update Processes

In our survey, we asked respondents about which types of updates they most frequently apply. In our interviews, we asked our participants how they determined which updates to deploy, and which types of updates they considered important. From the responses, we observed five primary factors that our participants discussed for assessing the cost-benefit trade-off of applying available updates. Our interview question was open-ended, so this set of factors may not be comprehensive or indicative of prevalence, as discussed in Section 3.5.

1. Update Type: In a closed-ended question, we asked our survey participants which updates they regularly installed: security or non-security related updates. In total, 97/102 administrators regularly installed security updates, whereas only 63/102 administrators did likewise for non-security related updates. (3 respondents did not answer.) We similarly asked our interview subjects an open-ended question about their views on which updates were important or not. Most interview participants (15/17) said that they considered security updates to be vital, but they disagreed on the importance of other updates; 7 administrators considered them important, whereas 5 administrators did not, often feeling they could be disruptive. For example, in a quote that is typical of what we heard, P16 explained: *"Least important, anything that's like feature updates or considered upgrades. I don't really want new features, because new features mean new problems, so I just want to get the security stuff tucked away."* Thus, our study participants typically found security updates important to apply.

2. Update Severity: In an open-ended interview question on how administrators decided to apply an update, the severity of the issues addressed by an update was a factor discussed by 9/17 interview participants. In a canonical example, P13 prioritized updates to *"Only critical security ones...It mostly depends on the severity and what the risk is."*

3. Update Relevance: When discussing their process for deciding to apply an update, five interview participants (29%) explicitly described update information overload, where much of the information they acquired did not apply to their machines. As a result, they said that they had to tediously filter out unnecessary information (or possibly avoid overly verbose feeds altogether). For example, P6 thought that *"Sometimes there's an overabundance of information...there are some products, things like that, that we don't use here. So I have to actively filter that out myself."* Others described receiving multiple emails about specific upgrades (e.g., Linux patches simultaneously released in batches) and how these emails were easily lost or hard to process in an overflowing inbox.

4. Update Reliability: Three interview subjects brought up known update issues as another factor in determining whether to update. For example, P11 cared about the update quality, saying *"a reliability score of an update would be my number one [update characteristic]."*

5. Organizational Factors: In many cases, organizational or management policies and decisions influenced or even dictated the update decision. We discuss in more detail in Section 10.

6.2 Impact on Updating Effectiveness

We found that system administrators prioritized updates that fixed security (or other severe) bugs. However, many software updates bundle bug fixes with feature or performance changes, including popular software such as the Mozilla Firefox Browser [4] and the Apache HTTP web server [2]. This entanglement suggests that it is challenging for administrators to specifically address the most urgent software problems without contending with other potential changes. Additionally, certain update characteristics (e.g., update reliability) were important to our study participants in deciding whether to apply an update. However, updates may not contain information to assess such characteristics (e.g., Firefox [4], Apache HTTP daemon [2]), or provide too much irrelevant information (described by study subjects as information overload), making it challenging for administrators to make informed updating decisions.

7. STAGE 3: PREPARING FOR UPDATE INSTALLATION

After identifying appropriate updates, our study participants reported that they had to make preparations for installation, which fell into three over-arching categories. First, administrators frequently *made backups/snapshots* in case problems arose through the updating process. Second, they *prepared machines* when necessary, such as by changing configurations or dependencies. These actions were

often necessary due to the manual nature of many updates. Finally, they often extensively *tested* updates for unintended side-effects or bugs. Here, we focus on the testing considerations of administrators as we cover the other two considerations in the remaining sections.

Threat of Bad Updates: We asked our survey and interview participants to describe their experiences with problems caused by updates on the machines they managed. In a closed-ended survey question, we asked how frequently an administrator encountered a problematic update. Of the 98/102 survey respondents that answered, all but 2 said that they had encountered bad updates; 54 indicated this happened infrequently, 36 found problems every few update cycles, and 6 said most update cycles produced complications. When asked an open-ended question on whether they tested updates and why, our interview subjects expressed the same sentiments on update risk; 8/17 recounted running into a recent faulty update. While the participants' recollections may not have been entirely accurate, it reflected a general sentiment among them that updating comes with non-trivial risks that they should manage. In the worst case, the negative experiences drove administrators towards fewer updates: *"I stopped applying updates because it was becoming more of a problem to apply them than not to. Production machines, they don't get updates"* (P12). Such behavior can leave hosts riddled with security vulnerabilities and ripe for compromise. To combat the risks of bad updates, many of our study participants engaged in the time-consuming process of update testing.

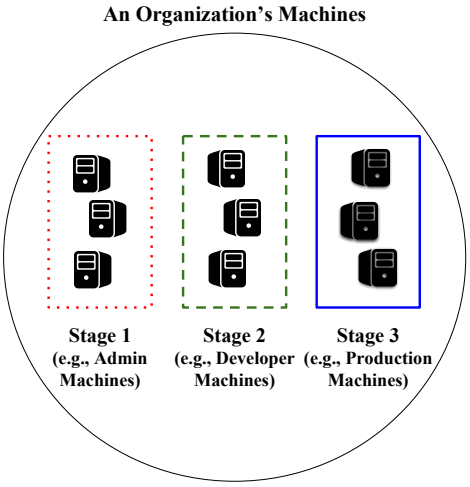

Figure 2: Staggered Deployment Testing: The system administrator allocates machines into stage groups, and updates stage by stage, waiting between each stage for update issues to manifest. If they arise, the administrator halts deployment and investigates the issues. For example, an administrator at a software company might first group only machines that they use as the first stage, then group developer machines as a second stage, and form a final stage of production machines.

7.1 Update Processes

Both our surveys and interviews contained an open-ended question asking respondents about what testing they do for updates, if any, and why. The majority of our participants (83/102 survey respondents and all 17 interview subjects) indicated they tested updates. (Seven survey participants did not respond.) Among those who tested, 22 survey participants and 3 interview subjects discussed only ad-hoc testing methods (e.g., testing basic software functionality) without discussing any strategies in detail. For the remaining administrators, we found that testing strategies varied but fell into two general classes: *staggered deployments* and *dedicated testing setups*. Regardless of the chosen strategy, testing was often a pain point for administrators: in open-ended survey questions on what works well and poorly in an administrator's updating process, only 14/102 survey respondents recounted positive testing experiences, and 12 reported that developing a reliable testing workflow was the most difficult aspect of updating. Thus, many of our study participants found it challenging to develop a dependable testing process.

1. Staggered deployments. When staggering update deployment (as illustrated in Figure 2), administrators in our study described separating their machines into multiple groups, deploying updates to a group at a time, and waiting some time between each stage of deployment to observe if update issues arise. In an example that summarizes this approach, S72 said that they first *"install on non-important machines and let them bake for 1+ months."* This strategy, which merges update testing and deployment, was the most commonly used among our study participants, leveraged by 43/102 of the survey respondents and 11/17 of the interview subjects.

We identified three different ways that participants used to grouped machines in each stage. First, 22 survey respondents and 4 interview subjects categorized machines into priority levels, testing updates first on lower priority machines. A second approach (10 survey respondents and 2 interview subjects) was to test first on the machines of end users who opted into assisting with update testing. For example, P10 talked about deploying updates to volunteers for a week prior to company-wide rollout, a strategy many spoke of using because: *"They're very good at reporting things that have gone wrong."* A final less-frequently used strategy was to pick pilot groups at random, only discussed by one survey participant and two interview subjects. While P11 selected machines completely at random, independent of the user, P5 chose randomly with more nuance: *"Usually, it's randomly picking something that I know is active but not the most active machine out there. If I pick something that nobody's using for anything, then, that's not a good place to test it. But, it's also not one of our highest risk servers."*

Our survey participants typically did not indicate how they monitored for update problems during staggered deployment, although four respondents mentioned gathering user feedback from those who piloted updates. Interview participants told us that they monitored how well updates were applied through monitoring software (6/17), lack of error messages (6/17), checking the machines for compliance (2/17), and user feedback (1/17).

2. Dedicated testing environments. Our survey participants often mentioned a dedicated testing setup, where they used machines provisioned specifically for testing (30/102 survey respondents) or relied on a testing or quality assurance (QA) team (9/102). (Five survey respondents used both approaches.) Figure 3 illustrates this process. Among interview participants, 8/17 used dedicated test servers, with two also having a QA team. S29 captured the gist of this approach: *"We test in a lab/test environment that has similar functions as our production environment. We do this to ensure we get accurate and reliable results that won't break our end users' applications."* Similarly, S19 gave an example of how QA teams conducted testing: *"For some third-party software (issue tracking, artifact management, etc.), our QA department has scripts to exercise business-critical functionality."* We note that 16 survey respondents and 5 interview subjects with dedicated testing also used staggered deployment, suggesting that often participants felt that dedicated testing was not sufficient by itself.

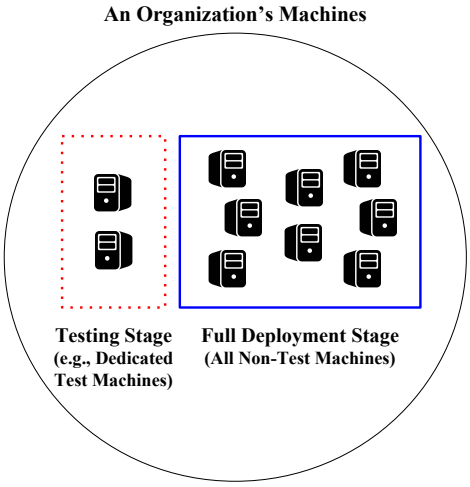

Figure 3: Dedicated Testing Environments: The system administrator evaluates an update in an environment configured specifically for testing (e.g., test servers). If they do not discover update issues, they fully deploy the update (potentially via staggered deployment).

Most of our study participants did not elaborate on the specific evaluation done in dedicated testing setups, although some mentioned automated software testing and manual investigation to confirm that critical software functionality remained. We note that no more than three survey or interview participants explicitly mentioned any particular method though, so further exploration on dedicated testing details is warranted.

4. No testing at all. A minority of survey respondents (10/102) did not test updates at all, and an additional two respondents indicated that they skipped testing on some of their systems as it was infeasible, without discussing testing on other systems. No interview subjects avoided testing. Three of the survey respondents who skipped testing did not provide their reasons. However, two survey respondents indicated they lacked the time, and three others deemed updates in their environment to be low-risk enough to deploy without testing. For example, S43 acknowledged, *"It is a poor habit but I don't ever experience any issues with Microsoft updates, so I see no reason to wait before applying them."* In another instance, a survey respondent skipped updates because testing on a diverse set of hosts seemed impractical, stating that with *"Too many different environments, would need to test a dozen different ways before deployment"* (S34). The final test-less respondent S37 stated that they skipped testing because *"security patches are a requirement, if it breaks something it gets fixed downstream."*

7.2 Impact on Updating Effectiveness

Those participants who used the staggered deployment testing strategy avoided the need for dedicated testing resources (although some used both strategies). However, we note that an important downside of staggered deployment for participants was that it could significantly delay updates to hosts in later stages. Some study participants indicated this delay could be on the order of weeks or months. Notably, administrators often spoke of updating production machines last, which is particularly concerning as these servers often directly interacted with external entities and hence, potential attackers.

Some of our administrators preferred dedicated testing environments for evaluating updates in a low-risk setting. However, we note this strategy requires additional computing resources or employees specifically for testing. In addition, we heard from participants about the challenges in replicating nuances of real-world deployments in testing environments. Ultimately, update testing was a challenging endeavor for most of the administrators in our study, driving some to even bypass testing.

8. STAGE 4: DEPLOYING UPDATES

Our study participants had to develop methods for deploying updates across the many machines under their purview.

8.1 Update Processes

Specifically, our study participants had to determine how to deploy updates and when to do it.

1. How to Deploy? In a closed-ended survey question, we asked survey participants whether they deployed updates manually, wrote their own programs or scripts to deploy updates, used third-party update management software, enabled automatic updates, or deployed in an alternate fashion (with a free-form response). Based on the 99/102 survey respondents who answered, we observed that administrators often lacked a single unified system for deploying updates. While 34 survey participants used a single method, the rest used multiple, with a median of 2 methods. We asked interview subjects an open-ended question on how they install updates, and interview participants also reported a mixture of deployment methods.

A majority of survey respondents used third-party update managers[1] (64/102), as did 12/17 of the interview subjects. P14 described their use of the update management software Ansible [1], explaining *"with Ansible you would just specify a list or subsection of a list of machines to run a particular command or update and it would run all of those in parallel on each of the machines and return the status of the request."* Some interview participants felt these tools could be improved to take snapshots of their systems and better indicate missing updates for specific machines.

Almost half of our participants (50/102 survey respondents and 7/17 interview subjects) created custom scripts or programs to automate the deployment process, while 44/102 survey participants and 2/17 interviewees enabled automatic updates for some software packages. Manual updates were still frequent though, conducted by 40/102 survey respondents and 4/17 interview subjects. One consequence of the heavy use of scripting and manual actions was the issue of legacy systems and processes. For example, P7 illustrated one scenario, saying *"If there's a legacy system in place and Jeff the sysadmin is the only dude who even knows how to run the scripts for that, or whatever service is running on there, you know, God forbid Jeff gets hit by a bus."*

On Automation: In response to open-ended survey questions on what aspects of an administrator's update management process work well and which are challenging, many study participants spoke of the importance of automation in the update deployment process. In a representative quote, S62 explained: *"Automating the process is essential for any environment with more than 10 endpoints as it greatly reduced the time involved and also improves the frequency of patch application."* S19 agreed in their response to the same survey question, stating *"There is no way our small team could manage this many machines without [automation]."* However, implementing automation often required significant effort. P15 stated they did not initially automate due to *"just the amount of time it would take*

[1]Tools mentioned included Ansible [1], SCCM [33], Chef [3], Spiceworks [8], Puppet [6], Terraform [9], and WSUS [34].

to implement all the automation." This participant did later deploy automation, stating it took them *"three months, to get it right."*

Even with the benefits of automation, our survey participants also highlighted that many situations still required manual actions, some in preparation for update installation (as mentioned in Section 7). For example, S14 sometimes still performed manual updates because *"Major OS updates require more manual intervention, such as updating custom scripts, updating or rewriting configuration files, or updating third-party tools."* In the interviews, some subjects mentioned that automation was not always desirable since update issues could arise unexpectedly.

Dependency and compatibility concerns posed particular problems for automation. In a prototypical example, S62 struggled with *"Maintaining compatibility with software that depends on platforms like Java/.Net/etc. Vendors tend to lag behind the platform by at least 1-2 release cycles preventing us from updating to the latest version."* Additionally, host heterogeneity (e.g., different software versions) complicated update deployment as illustrated by the following typifying example: S86 found deploying updates difficult when *"pushing to multiple versions of Linux with only one tool."*

Thus, while automatic updates and deployment automation was helpful and important for our study participants, they often could not fully automate updates across their machines due to some of the above reasons.

2. When to Deploy? In open-ended interview questions on how administrators deployed updates and whether they had to notify anyone about the update, our interview subjects frequently discussed the need to minimize disruptions for users and updated machines. (Our survey did not contain equivalent questions.) One strategy for mitigating disruptions (used by 13/17 interview subjects) was to update along a predictable schedule, such as P10's weekly patching program, so that users were not caught off-guard by the update timing. Another strategy mentioned by 12/17 interview participants was to update during off-hours. We also observed that organization and management decisions could dictate when updates occurred (described in Section 10).

In many cases, communication and coordination with those affected by an update were vital. This sentiment is best exhibited by P10's (who followed a weekly update schedule) discussion of their coordination efforts: *"On a given week, your machine might get software and it might reboot. We have a communication program that goes along with that, that we send out to the units about what's happening this week."* In a contrasting but similar example of coordination, P5 told us that they based update timing on user preferences: *"You send out an email to people and see what time works best for them. Usually, they can identify a time that is going to be idle for them or lower use than regular."*

8.2 Impact on Updating Effectiveness

Challenges in implementing automation for update deployment forced many of our participants to perform manual updates. In addition, administrators in our study often eschewed automatic updates so they could make proper preparations. We note that these manual actions could result in slower update rollouts leaving machines exposed to bugs and vulnerabilities for a longer duration. Also, manual updates may require further effort and be more prone to human error, potentially resulting in misconfigurations or functionality regressions. For our participants, the need to time updates in coordination with organization members or policies further widened the vulnerability window for machines.

9. STAGE 5: HANDLING UPDATE ISSUES AFTER DEPLOYMENT

Unfortunately, update testing did not always prevent issues from arising post-deployment. We asked our survey participants an open-ended question on how they became aware of problems caused by installed updates. In total, 56/102 survey participants found out about some update issues through user or client complaints, while 21 discovered problems through monitoring updated hosts. We further asked both our survey and interview participants open-ended questions about how they handled these post-deployment problems.

9.1 Update Processes

Of the 93/102 survey respondents that answered, only 3 indicated they lacked a process for managing post-deployments issues. From the interviews, 11/17 subjects reported recently running into post-deployment problems.

For the administrators that did deal with update complications, the most common approach was to uninstall an update. In total, 48/102 survey participants used this strategy, with 6 mentioning that they did so with custom scripts and 20 using third-party software or an update manager to do so (the rest did not specify). Similarly, 6/17 interviewees mentioned having to uninstall updates to resolve update problems. Another common approach was to revert to a previous snapshot or backup of the software or system. This strategy, used by 35/102 survey respondents and 7/17 interview subjects, did require proactive steps in preparation for update installation (namely, making a backup), as mentioned in Section 7. In an example of the forethought required of administrators, S5 discussed their backup strategy: *"I take an image of the entire disk once a month for non-critical machines and daily for critical machines."* Other rollback strategies mentioned less frequently during the surveys and interviews included downgrading to an earlier version of the software (possibly undoing several update cycles), manually negating an update's changes, or reverting to a mirrored/parallel environment.

The prior strategies all involve returning to a pre-update state, which can leave machines without patches for new vulnerabilities. Some administrators preferred to keep the updates in place, with 15/102 survey participants and 1/17 interviewees saying they attempted to find workarounds for problematic updates. Of these, 4 survey participants said they never roll back, focusing on keeping updates in place while managing any issues. Also, 7/102 of our survey participants relied on vendor assistance in resolving update issues.

9.2 Impact on Updating Effectiveness

After deploying an update, if problems arose, our study participants tended to revert to a functional but insecure prior state, demonstrating that they prioritized functionality over security. This behavior also suggests that system administrators found it difficult to identify workarounds or fixes for update problems, whether by themselves or via the software vendors.

10. ACROSS STAGES: ORGANIZATION AND MANAGEMENT INFLUENCE

A significant theme that emerged from our study participants was the important role that an organization's internal policies and management could play in update decisions. This theme provides new evidence extending the work by Dietrich *et al.* [15], who also observed that organizational factors impacted how administrators handled system misconfigurations.

We briefly note that we explored whether organizational structure, such as the number of employees or machines managed, affected our participant's update management practices, particularly related

to different testing and deployment strategies. To do so, we compared the distributions of the organization size and the number of machines managed between those adopting different updating behaviors. We used the Mann-Whitney-Wilcoxon test [29], with a p-value threshold of $\alpha = 0.05$, to determine if the distributions statistically significantly differed. However, we did not identify any significant differences; thus, the organizational structure did not appear strongly correlated with any particular update process.

10.1 Update Processes

Across responses to various open-ended questions, our study participants discussed situations where organizational policies and management affected updating practices.

1. Free Reign. In some organizations, administrators had decision-making authority and could apply updates as they saw fit. However, this put the onus on the administrator solely to keep machines secure. P11's company exemplified this approach: *"I don't have to run junk through a bunch of red tape to do anything. I just do it, knowing the consequences; things could break, could cause a lot of problems and lose a lot of money, but that's just part of having the responsibilities of that job I have. If I want to push out updates to all 1,800 machines, I don't have to really answer to anybody."*

2. Organizational Oversight. In other cases, administrators in our study told us they had to get management buy-in before taking certain update actions. A quote from S26 characterizes this setup, as they talked about applying updates only after management approval because *"I will be fired if I do so before I can convince management."* Similarly, in another representative example, S70 discussed that their update promptness was often delayed because *"Mostly the business being incompetent and not approving the work to go ahead. If it was up to me, [updates would be installed] as soon as they are released and after testing."* This setup often made updating challenging for participants. For example, S14 had to fight for maintaining Windows updates, as management felt that those updates were not trustworthy. These disagreements between administrators and management appeared to result in updating practices that the administrators in our study did not always support.

In some cases, organizational policies dictated the actions of the administrators. A canonical example from S37 illustrates the pressure on their update deployment timeline: *"Policy and compliance require deploying them within 5/10/30 days depending on severity."* In another example quote, P15 explained that their organization's requirements determined the priority of different updates: *"We have compliance implications around getting security updates out, so that's one. We have an organizational mandate to deliver a stable platform, so stability updates set prioritized as well."* With a potentially less secure outcome, P12's organization decided to reduce the frequency of machine updating, because *"that's just more of a decision that we've made as a business that...it's just better not to introduce a problem."*

Several study participants also commented on another important organizational decision: the budget allocated for system administrator operations. For example, S21 said they lacked the time for managing updates but *"My company won't let me buy anything to help with automatically deploying."* Similarly, P16 said that they lacked the budget for obtaining good software to handle updates until demonstrating their network's insecurities to management.

10.2 Impact on Updating Effectiveness

Organizational freedom allowed some of our study participants to more effectively apply updates, but placed the burden of security on their shoulders alone. We note that such freedom could result in ad-hoc decision making by administrators, potentially resulting in poor practices, or decisions that could negatively impact other aspects of an organization, such as the reliability or availability of an organization's production systems.

By contrast, requiring management approval complicated the update process for many system administrators and could delay or prevent the application of updates. Such barriers also drove down the updating frequency for those administrators who told us they can only request approval for the most severe updates, and often, some skipped less severe updates to avoid the hassle of getting approval.

11. DISCUSSION

Our study of system administrator software updating identified how administrators perform updates and the security implications of their behaviors. Future user studies on administrator software updating could extend our work to develop a richer model of update decision-making processes, investigate how updating differs for different types of organizations and machines, explore the effects of organizational policies on updates in more depth, and identify concrete steps for improving updating tools and interfaces. In this section, we synthesize our findings to identify how software updating differs between system administrators and end users, and how we can help administrators better keep machines updated through recommendations grounded in our results.

11.1 Comparison with End User Software Updating Practices

Prior work on software updating behavior has primarily studied end users. From synthesizing and comparing with the results from existing studies [19, 30–32, 45, 46], we find that end users follow similar stages of the updating process, but with differing considerations at each stage. Overall, we observe that administrators performed more sophisticated tasks (e.g., testing) and had unique aspects of their workflows as a result of managing numerous heterogeneous machines within an organizational context (e.g., staggered deployment, organizational influences). For each of our five updating stages (summarized in Figure 1), we highlight the salient differences between end user and system administrator considerations.

- **Stage 1 (Learning):** Administrators relied on a diverse set of update information sources, including those from proactive searching. In comparison, end users primarily learned about updates through notifications or alerts from within their software and rarely sought updates by themselves [31, 46].
- **Stage 2 (Deciding):** Like end users, administrators in our study considered the benefits and risks of an update [19, 31, 32, 45, 46]. However, our participants had the additional facet of determining if and which updates affected the potentially heterogeneous hosts in their organization. Some administrators also had to abide by organization policies.
- **Stage 3 (Preparing):** We observed that update-induced issues concerned both our study participants and end users [19, 30, 31, 45, 46]. As a result, end users either avoided updating, updated after making backups, or dealt with update issues only after applying [46]. In comparison, administrators took more extensive preparatory steps, including backing up and snapshotting systems, modifying software configurations and dependencies, and testing updates before applying them.
- **Stage 4 (Deploying):** As administrators in our study deployed updates at scale, unlike end users, they had to consider the interruptions and downtime on machines they served, often requiring coordination with other organization members or or-

ganization approval to take actions. They also often employed automation to scale up their updating tasks.

- **Stage 5 (Remedying):** When updates caused issues, both populations employed similar high-level remedies (e.g., uninstalling updates, finding workarounds) [46]. However, administrators in our study had to contend with the challenge of identifying update issues across numerous machines that they updated, requiring them to consider monitoring systems and feedback from these machines' users. Additionally, as these issues could affect organizational operations, organization factors influenced how administrators handled these situations.

11.2 Reducing the Burden of Update Information Retrieval

In Section 5, we learned that information on software updates is widely dispersed across various sources. Our findings suggest that helping administrators more easily identify relevant updates for their machines would simplify their updating efforts and increase the likelihood of prompt updating. One solution could be to standardize and consolidate update information at a centralized repository (similar to efforts on aggregating vulnerability information [36]), providing a singular destination for identifying available updates.

Another intriguing approach is through outreach campaigns that inform administrators about severe vulnerabilities and promote updating to patch the security holes. Several recent works [17, 27, 28, 43, 44] have investigated the benefits of reaching out to the administrators of machines with publicly visible security issues, finding that the notification efforts resulted in a significant improvement in the remediation of the security problems. However, they also identified hurdles in contacting all administrators and promoting corrective actions, and there remain important research questions such as how to effectively deliver messages, whom to contact, how to establish trust with recipients, and how to incentivize remediation. Thus, we recommend further research on improving administrator notifications to overcome existing challenges and identify best practices.

11.3 Simplifying Update Decision-Making

Our findings in Section 6 indicate that administrators prioritize updates with certain characteristics (e.g., update severity), so standardizing update information to consistently include such characterizations would aid them in their decision-making. In particular, administrators differentiate update types. Thus, there is value in splitting all-inclusive updates into updates specific to one type of patch, as also recommended by prior work on end user updates [31,46]. For example, software vendors could bundle security patches separately from feature patches. With this segregation, administrators can better prioritize the updates they apply (e.g., security fixes). However, we recognize that splitting updates could complicate software development and release. Future work could therefore explore how best to separate and enable updates of different types, from both the software developer and administrator standpoints.

11.4 Improving Update Deployment Processes

There remains a salient need for advancements in the update tools that system administrators rely upon, as we observed that administrators encountered various hurdles throughout the preparation and deployment of updates (Sections 7 and 8), and the handling of post-deployment problems (Section 9). For example, the notion that automatic updates would solve the patching problem is overly simplistic, as our findings demonstrate the complexities of the updating process (particularly with situations still requiring manual actions, as discussed in Section 8).

While technical developments are certainly needed, we also lack a deep understanding of the usability of these tools. Therefore, the usable security community could contribute explorations into how administrators use update tools and how their interfaces could be improved. For example, our findings (in Section 8) indicate that many administrators use third-party update managers. What information do they display before, during, and after update deployment, and what missing information (such as on dependencies or affected configurations) would streamline administrator workflows if provided?

One notable deployment issue our administrators faced was timing updates to avoid operation interruptions. We believe that dynamic software updating [21] (DSU), a method that allows for live updates without restarts or downtime, could help with side-stepping update timing concerns. While it has not yet been widely deployed, the approach is promising as some major systems have adopted it, such as with the Linux kernel extension Ksplice [5]. However, we have little understanding so far of how using DSU systems affect developers writing patches and administrators operating such systems. For example, the use of DSU systems can result in complex data representations and less readable code, potentially impacting the software development process. Similarly, DSU systems may not serve as a complete solution for system administrators if they still require approval or coordination before initiating updates, even without system downtime. Research into the usability of dynamic updating systems and avenues for improvement could potentially eliminate update timing concerns for administrators in the future.

11.5 Shifting Organizational Culture on Software Updates

In Section 10, we identified that organization management and policies can impact administrator actions, often impeding secure updating practices. A culture shift at organizations to recognize the importance of expedient updates (particularly for security issues) would help administrators perform their jobs more successfully. If end users and management do not readily accept that updates should be routinely applied, it becomes difficult to balance system maintenance and security with minimizing operational interruptions. Similarly, if organizations do not devote enough resources for administrators to adequately perform update tasks or have some oversight for security operations, security lapses can occur (e.g., Equifax [38]).

Resolutions to this problem are not straightforward. Existing recommendations such as NIST SP 800-40 [37] provide some guidance on organizational structures that promote updating. However, investigating how administrators deal with data breaches (similar to studies on end users facing breaches [52]) could provide insights into how to better facilitate practices that enable, not hinder, security, beyond solely relying on organizational security education. Such studies could also inform regulatory policies on security oversight. For instance, Equifax currently reports to 8 US states about their security overhaul [39]. The usable security community could offer insights into whether such audits fit into administrator workflows and improve security overall, or whether other policy approaches may better incentivize organizations to implement and prioritize security best practices.

12. CONCLUSION

System administrators play a vital role in securing machines on behalf of their organizations. One of their primary tasks is to manage the updates on numerous hosts to counter emergent vulnerabilities. However, prior work has paid less attention to how exactly they do so. In this paper, we examined how administrators manage software updates, determining five primary stages of updating and the various

considerations and actions associated with each stage. We identified pain points in administrator updating processes, such as when learning about updates, testing for and handling update-caused issues, deploying updates without causing operation disruptions, and dealing with organizational and management oversight. Based on our findings, we developed recommendations grounded in our results, and provided research directions for better support of administrators in keeping their hosts updated and secure.

13. ACKNOWLEDGMENTS

We thank our study participants, as well as Josefine Engel for running the pilot portion of our study. We also thank Serge Egelman, Katharina Krombolz, and Emanuel von Zezschwitz for meaningful discussions, and Noah Apthorpe for providing feedback on an earlier version of our paper. Finally, we thank our anonymous reviewers for providing constructive feedback. This work was supported in part by the National Science Foundation under awards CNS-1518921 and CNS-1619620.

14. REFERENCES

[1] Ansible. https://www.ansible.com/.

[2] Apache HTTPD Changelog. https://www.apache.org/dist/httpd/CHANGES_2.4.

[3] Chef. https://www.chef.io/chef/.

[4] Firefox Release Notes. https://www.mozilla.org/en-US/firefox/releases/.

[5] Ksplice. https://www.ksplice.com/.

[6] Puppet. https://puppet.com/.

[7] Reddit. https://www.reddit.com/.

[8] Spicework. https://www.spiceworks.com/.

[9] Terraform. https://www.terraform.io/.

[10] D. Armstrong, A. Gosling, J. Weinman, and T. Marteau. The Place of Inter-Rater Reliability in Qualitative Research: An Empirical Study. *Sociology*, 31(3):597–606, 1997.

[11] R. Barrett, E. Kandogan, P. P. Maglio, E. M. Haber, L. A. Takayama, and M. Prabaker. Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices. In *ACM Conference on Computer Supported Cooperative Work (CSCW)*, 2004.

[12] S. Chiasson, P. van Oorschot, and R. Biddle. Even Experts Deserve Usable Security: Design Guidelines for Security Management Systems. In *SOUPS Workshop on Usable IT Security Management (USM)*, 2007.

[13] Cisco. Annual Security Report. https://www.cisco.com/web/offers/pdfs/cisco-asr-2015.pdf, 2015.

[14] O. Crameri, N. Knezevic, D. Kostic, R. Bianchini, and W. Zwaenepoel. Staged Deployment in Mirage, an Integrated Software Upgrade Testing and Distribution System. *SIGOPS Oper. Syst. Rev.*, 41(6):221–236, Oct. 2007.

[15] C. Dietrich, K. Krombholz, K. Borgolte, and T. Fiebig. Investigating System Operators' Perspective on Security Misconfigurations. In *ACM Conference on Computer and Communications Security (CCS)*, 2018.

[16] T. Duebendorfer and S. Frei. Why Silent Updates Boost Security. Technical report, ETH Zurich, 2009.

[17] 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 *ACM Internet Measurement Conference (IMC)*, 2014.

[18] S. Farhang, J. Weidman, M. M. Kamani, J. Grossklags, and P. Liu. Take It or Leave It: A Survey Study on Operating System Upgrade Practices. In *Annual Computer Security Applications Conference (ACSAC)*, 2018.

[19] A. Forget, S. Pearman, J. Thomas, A. Acquisti, N. Christin, L. F. Cranor, S. Egelman, M. Harbach, and R. Telang. Do or Do Not, There Is No Try: User Engagement May Not Improve Security Outcomes. In *USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2016.

[20] C. Gkantsidis, T. Karagiannis, and M. Vojnovic. Planet Scale Software Updates. *ACM SIGCOMM CCR*, 36(4):423–434, Aug. 2006.

[21] M. Hicks and S. Nettles. Dynamic Software Updating. *ACM Transactions on Programming Languages and Systems*, 27(6):1049–1096, Nov. 2005.

[22] I. Ion, R. Reeder, and S. Consolvo. "...No one Can Hack My Mind": Comparing Expert and Non-Expert Security Practices. In *USENIX Symposium On Usable Privacy and Security (SOUPS)*, 2015.

[23] E. Kandogan, P. Maglio, E. Haber, and J. Bailey. *Taming Information Technology: Lessons from Studies of System Administrators*. Oxford University Press, 2012.

[24] S. Kraemer and P. Carayon. Human Errors and Violations in Computer and Information Security: The Viewpoint of Network Administrators and Security Specialists. *Applied Ergonomics*, 38(2):143 – 154, 2007.

[25] K. Krombholz, W. Mayer, M. Schmiedecker, and E. Weippl. "I Have No Idea What I'm Doing" - On the Usability of Deploying HTTPS. In *USENIX Security Symposium*, 2017.

[26] L. L. Kupper and K. B. Hafner. On Assessing Interrater Agreement for Multiple Attribute Responses. *Biometrics*, 45(3):957, Sept. 1989.

[27] F. Li, Z. Durumeric, J. Czyz, M. Karami, M. Bailey, D. McCoy, S. Savage, and V. Paxson. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *USENIX Security Symposium*, 2016.

[28] F. Li, G. Ho, E. Kuan, Y. Niu, L. Ballard, K. Thomas, E. Bursztein, and V. Paxson. Remedying Web Hijacking: Notification Effectiveness and Webmaster Comprehension. In *World Wide Web Conference (WWW)*, 2016.

[29] H. Mann and D. Whitney. On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[30] A. Mathur and M. Chetty. Impact of User Characteristics on Attitudes Towards Automatic Mobile Application Updates. In *USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[31] A. Mathur, J. Engel, S. Sobti, V. Chang, and M. Chetty. "They Keep Coming Back Like Zombies": Improving Software Updating Interfaces. In *USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2016.

[32] A. Mathur, N. Malkin, M. Harbach, E. Peer, and S. Egelman. Quantifying Users' Beliefs about Software Updates. In *NDSS Workshop on Usable Security*, 2018.

[33] Microsoft. System Center Configuration Manager. https://www.microsoft.com/en-us/cloud-platform/system-center-configuration-manager.

[34] Microsoft. Windows Server Update Services. https://docs.microsoft.com/en-us/windows-server/administration/windows-server-update-services/get-started/windows-server-update-services-wsus.

[35] A. Nappa, R. Johnson, L. Bilge, J. Caballero, and T. Dumitraş. The Attack of the Clones: A Study of the Impact of Shared Code on Vulnerability Patching. In *IEEE Symposium on Security and Privacy (S&P)*, 2015.

[36] National Institute of Standards and Technology. National Vulnerability Database. https://nvd.nist.gov/.

[37] National Institute of Standards and Technology. Special Publication 800-40 Revision 3: Guide to Enterprise Patch Management Technologies. https://doi.org/10.6028/NIST.SP.800-40r3, 2013.

[38] L. H. Newman. Equifax Officially Has No Excuse. https://www.wired.com/story/equifax-breach-no-excuse/, September 2017.

[39] L. H. Newman. Equifax's Security Overhaul A Year After Its Epic Breach. https://www.wired.com/story/equifax-security-overhaul-year-after-breach/, July 2018.

[40] M. Oltrogge, Y. Acar, S. Dechand, M. Smith, and S. Fahl. To Pin or Not to Pin—Helping App Developers Bullet Proof Their TLS Connections. In *USENIX Security Symposium*, 2015.

[41] K. Rankin. Sysadmin 101: Patch Management. Linux Journal. https://www.linuxjournal.com/content/sysadmin-101-patch-management, 2017.

[42] I. Seidman. *Interviewing As Qualitative Research: A Guide for Researchers in Education and the Social Sciences*. Teachers college press, 2013.

[43] B. Stock, G. Pellegrino, F. Li, M. Backes, and C. Rossow. Didn't You Hear Me? Towards More Successful Web Vulnerability Notifications. In *Network and Distributed System Security Symposium (NDSS)*, 2018.

[44] B. Stock, G. Pellegrino, C. Rossow, M. Johns, and M. Backes. Hey, You Have a Problem: On the Feasibility of Large-Scale Web Vulnerability Notification. In *USENIX Security Symposium*, 2016.

[45] K. Vaniea, E. Rader, and R. Wash. Betrayed by Updates: How Negative Experiences Affect Future Security. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*, 2014.

[46] K. Vaniea and Y. Rashidi. Tales of Software Updates: The Process of Updating Software. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*, 2016.

[47] N. F. Velasquez and S. P. Weisband. Work Practices of System Administrators: Implications for Tool Design. In *ACM Symposium on Computer Human Interaction for Management of Information Technology (CHiMiT)*, 2008.

[48] N. F. Velasquez and S. P. Weisband. System Administrators As Broker Technicians. In *ACM Symposium on Computer Human Interaction for the Management of Information Technology (CHiMiT)*, 2009.

[49] R. Wash and E. Rader. Too Much Knowledge? Security Beliefs and Protective Behaviors Among United States Internet Users. In *USENIX Symposium On Usable Privacy and Security (SOUPS)*, 2015.

[50] R. Wash, E. Rader, K. Vaniea, and M. Rizor. Out of the Loop: How Automated Software Updates Cause Unintended Security Consequences. In *USENIX Symposium On Usable Privacy and Security (SOUPS)*, 2014.

[51] C. Weston, T. Gandell, J. Beauchamp, L. McAlpine, C. Wiseman, and C. Beauchamp. Analyzing Interview Data: The Development and Evolution of a Coding System. *Qualitative Sociology*, 24(3):381–400, 2001.

[52] Y. Zou, A. H. Mhaidli, A. McCall, and F. Schaub. "I've Got Nothing to Lose": Consumers' Risk Perceptions and Protective Actions after the Equifax Data Breach. In *USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2018.

APPENDIX

A. PRELIMINARY PHASE - PILOT INTERVIEW QUESTIONS

Below we list the questions from our semi-structured pilot interviews (the preliminary phase of the study, as described in Section 3).

Job responsibilities and processes

1. Tell me more about your main job responsibilities (how does he/she keep machines up to date).
2. Tell me about any relationships you have with the vendors that develop the software updates for the programs your organization/employees depend on.
3. Can you walk me through your process of how you find out about an update?
4. Why do you find out about updates in this way?
5. How do you determine which updates to deploy on the machines you manage?
6. Tell me more about how this process differs for the types of machines you manage?
7. Why does your deployment process differ for different machines?
8. How does the process differ depending on who owns the machines, if at all?
9. Tell me more about how you install the software updates (manually, automatic, silent) you apply.
10. Why do you apply the software updates in this way?
11. Can you walk me through the process of testing whether an update will be compatible with the machines?
12. Why do you do this testing for the updates? Do you test all updates and why?

Software Update Information

1. Tell me about the information you currently receive when an update is available.
2. How do you usually receive this information?
3. Do you ever seek additional information about updates? Why or why not?
4. What are the main advantages of the current update information? Why?
5. What are the main disadvantages of the current update information? Why?
6. Which is the least important part of the current update information for you?
7. Which is the most important part of the current update information for you?

Securing the Users

1. Tell me about what you do to protect your users.
2. Tell me about what kinds of online hazards you are protecting them from.
3. Once an update is deployed how do you communicate the information to the end users?
4. What do you expect of the end users once the updates are released?

5. Can you tell me about the process of deciding what updates you can trust?

Software Updates in General

1. What updates are most important to you? Why?
2. What updates are least important? Why?
3. Tell me what cybersecurity means to you.
4. What are the most important things to consider to secure the network?
5. What are the least important things to consider to secure the network?
6. What are the main advantages of the current software updating process? Why?
7. What are the main disadvantages of the current software updating process? Why?
8. What changes would you want to make to software updates? Why?
9. Is there anything else you would like to tell us about how you manage software updates?

B. PHASE ONE - SURVEY QUESTIONS

Below we list the questions from our survey (phase one of the study, as described in Section 3).

1. How old are you?
   (a) 18-25
   (b) 26-35
   (c) 36-45
   (d) 46-55
   (e) 56-65
   (f) Over 65
   (g) I do not wish to disclose
2. Which state do you live in?
3. What is your gender?
   (a) Male
   (b) Female
   (c) Other
4. What is your annual income?
   (a) Less than $25,000
   (b) $25,000 to $34,999
   (c) $35,000 to $49,999
   (d) $50,000 to $74,999
   (e) $75,000 to $99,999
   (f) $100,000 to $124,999
   (g) $125,000 to $149,999
   (h) $150,000 or more
5. What is your job title?
6. For how many years have you worked as a System Administrator in your current role?
7. For how many years have you worked as a System administrator before you entered your current role?
8. What is the highest level of education that you have completed?
   (a) 12th grade or less
   (b) High school degree or equivalent
   (c) Some college, no degree
   (d) Bachelor's degree
   (e) Master's degree
   (f) Other graduate degree
9. What was the subject area of your highest level of education (if above high school)?
10. What technical certifications, courses, or degrees have you completed, if any? You may paste entries from your resume or CV if you wish.
11. When did you complete these certifications or education? (Check all that apply)
    (a) Before I took up my current role
    (b) After I took up my current role
12. How have these technical certifications, courses, or degrees helped you complete your current role?
13. What is the industry of the organization that you work for?
14. How large is the organization that you work for?
    (a) $\leq$10 employees
    (b) 11 - 50 employees
    (c) 51 - 100 employees
    (d) 101 - 500 employees
    (e) 501-2000 employees
    (f) More than 2000 employees
15. What is the main purpose of the organization you work for?
16. How many machines/devices do you manage?
    (a) *Sliding scale between 0 and 1000+*
17. What type of machines/devices do you manage? (Check all that apply)
    (a) Laptops
    (b) Desktops
    (c) Servers
    (d) Mobile devices
    (e) Routers/network appliances such as firewall middleboxes
    (f) Embedded devices/ Internet of Things
    (g) Other: *free response*
18. What are the operating systems on the machines that you manage? (Check all that apply)
    (a) Mac
    (b) Windows
    (c) Linux
    (d) iOS
    (e) Android
    (f) Blackberry
    (g) ChromeOS
    (h) None
    (i) Other: *free response*
19. What is the predominant operating system, if any?
    (a) Mac
    (b) Windows
    (c) Linux
    (d) iOS
    (e) Android

(f) Blackberry
(g) ChromeOS
(h) Other: *free response*

20. What are these machines used for? (Check all that apply)
    (a) Education or training
    (b) Personal
    (c) Research
    (d) Servers
    (e) Work
    (f) Testing
    (g) Other: *free response*
21. Which of the following applies to the machines you manage? (Check all that apply)
    (a) The machines are used internally by the organization you work for
    (b) The machines are used externally by customers of the organization you work for
    (c) Other: *free response*
22. What updates are most important to you and why?
23. What updates are most important to your organization and why?
24. Are you solely responsible for updating the machines you manage?
    (a) Yes
    (b) No
    (c) Other: *free response*
25. How many updates do you run on the machines that you manage per week?
    (a) *Sliding scale between 1 and 500+*
26. How do you manage the updates across the machines/devices you manage? (Check all that apply)
    (a) I log into each system to perform updates
    (b) I use 3rd party software to manage the updates
    (c) I write programs to manage updates
    (d) I enable automatic updates
    (e) Other: *free response*
27. What type of updates do you install regularly? (Check all that apply)
    (a) Security updates
    (b) Non-security related updates
    (c) Other: *free response*
28. Select all of the security measures you take to protect your machines.
    (a) Firewall
    (b) Intrusion Detection System
    (c) Intrusion Prevention System
    (d) Antivirus System
    (e) Security updates
    (f) Different accounts with varying access (admin, regular, etc.)
    (g) Access codes/Passwords
    (h) Port scanners
    (i) Vulnerability testing
    (j) Backup and Disaster Recovery
    (k) Other: *free response*
29. How are the security measures you use deployed? (Check all that apply)
    (a) On the hosts
    (b) On the network
    (c) Other: *free response*
30. How do you find out about the updates you apply on the machines you manage? (Check all that apply)
    (a) Online forums
    (b) Security advisories
    (c) Blogs
    (d) News
    (e) Social media
    (f) RSS feeds
    (g) Professional mailing lists
    (h) Project mailing lists
    (i) Direct notification from vendor
    (j) Third-party service
    (k) When the software pops up a notification
    (l) Other: *free response*
31. When do you apply security updates? (Check all that apply)
    (a) As soon as they are released
    (b) After testing
    (c) On a regular cadence
    (d) After a specific amount of time since its release has elapsed
    (e) Applied automatically
    (f) Other: *free response*
32. What is the reason for applying updates in the frequency described above?
33. When do you apply non-security related updates? (Check all that apply)
    (a) As soon as they are released
    (b) After testing
    (c) On a regular cadence
    (d) After a specific amount of time since its release has elapsed
    (e) Applied automatically
    (f) Other: *free response*
34. What is the reason for applying non-security related updates in the frequency described above?
35. What kind of testing do you do with updates (if any), before applying them to the machines/devices you manage? Please explain why.
36. How frequently do you find an update to cause problems on the machines you manage?
    (a) Never
    (b) Rarely
    (c) Occasionally (every few update cycles)
    (d) Frequently (most update cycles)
37. How do you become aware of any problems caused by updates that you install?
38. What, if any, is your process for rolling back or undoing updates that cause problems on the machines you manage?
39. What aspects or steps in your update management process work well for you?
40. What aspects or steps in your update management process are most challenging to handle?
41. What would help you to better manage software updates for multiple machines?

C. PHASE TWO - INTERVIEW QUESTIONS

Below we list the questions from our semi-structured interviews (phase two of the study, as described in Section 3).

Job responsibilities and processes

1. Tell me more about the company you work for?
2. Tell me more about your main job responsibilities (how does he/she keep machines up to date)
3. How long have you worked in your job?
4. Have you had any training in IT? If so, tell me more about that.
5. Have you had any training in security? If so, tell me more about that.

Machines/Devices Managed

1. Does your organization have any security related policies for their machines?
2. How many machines/devices do you manage?
3. What kinds of machines/devices do you manage?
4. What are these machines used for?
5. Who are these machines used by?

Managing Software Updates for Multiple Machines

1. Does your company have any policies on software updates for their machines?
2. How do you handle security for these machines?
3. How often do you update these machines? Does the frequency differ for different machines? If so, why?
4. Who do you have to notify about updates that you are applying? Why?
5. In an average week, how many hours do you spend dealing with software updates?
6. Can you walk me through your process of how you find out about an update?
7. What are the advantages of using this process?
8. What are the disadvantages of using this process?
9. How do you determine which updates to deploy on the machines you manage?
10. When do you apply updates for the machines you manage? Why?
11. What is your process for applying updates on the machines you manage?
12. Tell me more about how this process differs for the types of machines you manage.
13. Why does your deployment process differ for different machines?
14. How does the process differ depending on who owns the machines, if at all?
15. Tell me more about how you install the software updates (manual, automatic, silent) you apply.
16. Why do you apply the software updates in this way?
17. Do you use any tools/programs to help you manage updates on multiple devices? What are these tools? Why do you use them?
18. Do you test whether an update will be compatible with the machines you manage in any way? How so?
19. Why do you do this testing for the updates? Do you test all updates and why?
20. How do you track which updates different machines need?
21. Do you prioritize any particular type of updates for any machines? Why/why not?
22. How do you track how well updates have been installed on different machines?
23. If any update requires a restart, what is your process for managing the restart?
24. Do you have to notify anyone about updates that you have applied or are about to apply?

Software Update Information

1. Tell me about the information you currently receive when an update is available.
2. How do you usually receive this information?
3. Do you ever seek additional information about updates? Why or why not?
4. What are the main advantages of the current update information? Why?
5. What are the main disadvantages of the current update information? Why?
6. Which is the least important part of the current update information for you?
7. Which is the most important part of the current update information for you?
8. What improvements would you make to the information that is included with current updates?

Securing the Users

1. Who are the users that you manage machines for?
2. Tell me about what you do to protect your users.
3. Do you use any technical solutions to protect users?
4. Do you use any educational solutions for protecting your users?
5. Are these solutions driven by your own or company policy? Tell me more about that.
6. Tell me about what kinds of online hazards you are protecting them from.
7. Once an update is deployed how do you communicate the information to the end users?
8. What are your responsibilities for handling updates for your users?
9. What are the responsibilities of your users for handling updates?
10. Can you tell me about the process of deciding what updates you can trust?

Software Updates in General

1. What updates are most important to you? Why?
2. What updates are most important to your organization? Why?
3. How does your organizational policy influence how you manage updates if at all?

4. What updates are least important to you? Why?
5. What updates are least important to your organization? Why?
6. Tell me what cybersecurity means to you.
7. What are the most important things to consider to secure your machines?
8. What are the least important things to consider to secure your machines?
9. What are the main advantages of your current software updating process? Why?
10. What are the main disadvantages of your current software updating process? Why?
11. What would your ideal way to handle software updates be? Why? What changes would you want to make to software updates themselves? Why?
12. Is there anything else you would like to tell us about how you manage software updates?