Asking for a Friend: Evaluating Response Biases in Security User Studies

Elissa M. Redmiles[1], Ziyun Zhu[1], Sean Kross[2], Dhruv Kuchhal[3], Tudor Dumitras[1], and Michelle L. Mazurek[1]
[1]University of Maryland, [2]University of California San Diego, [3]Maharaja Agrasen Institute of Technology

ABSTRACT

The security field relies on user studies, often including survey questions, to query end users' general security behavior and experiences, or hypothetical responses to new messages or tools. Self-report data has many benefits – ease of collection, control, and depth of understanding – but also many well-known biases stemming from people's difficulty remembering prior events or predicting how they might behave, as well as their tendency to shape their answers to a perceived audience. Prior work in fields like public health has focused on measuring these biases and developing effective mitigations; however, there is limited evidence as to whether and how these biases and mitigations apply specifically in a computer-security context. In this work, we systematically compare real-world measurement data to survey results, focusing on an exemplar, well-studied security behavior: software updating. We align field measurements about specific software updates (n=517,932) with survey results in which participants respond to the update messages that were used when those versions were released (n=2,092). This allows us to examine differences in self-reported and observed update speeds, as well as examining self-reported responses to particular message features that may correlate with these results. The results indicate that for the most part, self-reported data varies consistently and systematically with measured data. However, this systematic relationship breaks down when survey respondents are required to notice and act on minor details of experimental manipulations. Our results suggest that many insights from self-report security data can, when used with care, translate to real-world environments; however, insights about specific variations in message texts or other details may be more difficult to assess with surveys.

KEYWORDS

Data-Driven Security, Usable Security, Science of Security

ACM Reference Format:
Elissa M. Redmiles, Ziyun Zhu, Sean Kross, Dhruv Kuchhal, Tudor Dumitras, and Michelle L. Mazurek. 2018. Asking for a Friend: Evaluating Response Biases in Security User Studies. In *2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), October 15–19, 2018, Toronto, ON, Canada.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3243734.3243740

CCS '18, October 15–19, 2018, Toronto, ON, Canada

ACM ISBN 978-1-4503-5693-0/18/10…$15.00
https://doi.org/10.1145/3243734.3243740

1 INTRODUCTION

The security of computer systems often depends on choices made by human users, who may make mistakes or prioritize other needs and preferences. One solution is to limit the burden on users by removing them from the security loop, but this is not always possible or preferable [14]. Where humans remain involved, research in usable security often aims to make it easier for non-expert users to understand security protections and apply them effectively, in part by understanding how and why users make security-relevant decisions.

Typically, researchers attempt to achieve this understanding either via user studies (e.g., surveys and lab studies) or via field measurements on real-world data. Both approaches have strengths and limitations. Field studies can be difficult or expensive to conduct and are usually observational in nature, making it difficult to control for all possible confounding factors and to obtain strong evidence about why users make decisions. User studies, in contrast, offer more control, and potentially deeper insights, but have less ecological validity. As such, results from user studies, although valuable, have not always translated to the real world: Fahl et al. found that password creation studies only somewhat reflect users' actual choices [16], and researchers from Google found that the best TLS warning messages identified by surveys did not always pan out in A/B field tests [2].

There are a number of possible reasons for such discrepancies, including: (1) despite the best efforts of the research teams, the user studies may not have been designed most optimally to elicit accurate reports; (2) the user studies may have not been conducted with a sample that effectively represents the actual user population; (3) people may not know themselves well enough to accurately report on their in-the-wild behavior; or (4) the environment of user studies may simply not be effective for answering certain types of questions.

Other fields face similar challenges. For example, public health researchers who wish to measure and understand risky behaviors — e.g., heavy drinking, unprotected sex, smoking — often use surveys to measure the frequency of these behaviors and identify correlated factors to target with interventions [31, 35]. To enable good outcomes from these surveys, survey methodology researchers have painstakingly investigated how different survey designs and samples affect responses, and how these responses reflect real-world behavior [8, 18, 29, 32, 33, 64]. They discovered that cognitive biases, such as difficulty predicting behavior for hypothetical future situations, or reluctance to report socially undesirable practices,

affect survey results [31, 49]. To compensate for these biases, researchers developed new methods and best practices that can be used to obtain more accurate measurements [10, 29, 64].

It is not clear whether these compensatory approaches will translate to the digital security and privacy domain. For example, best practices from warning design literature did not automatically translate to better security-warning comprehension [17]. Prior work comparing survey samples also suggests that using online samples to ask questions about online security and privacy has unique biases that must be accounted for [27, 53]. Research is therefore needed to understand how user study data deviates from real-world observations, in order to understand how to best mitigate and correct these biases. So far there is unfortunately little such work comparing user study results to the real world [2, 16, 39].

Our work takes an important step toward more thoroughly measuring biases between digital security surveys and real-world security practices. To do so, we focus on one exemplar, well-studied [4, 11, 13, 15, 21, 23, 25, 36–38, 40, 41, 43, 46, 54, 56, 59, 62, 66–68, 71] security behavior for which significant measurement data is available: software updating. We compare the results of a systematic measurement (n=517,932) of user behavior in response to 11 different software updating messages, collected using the WINE system [12], to responses to a survey asking respondents to self-report their intended behavior and reasoning for updating in response to the same messages. To better understand self-report biases and sample source effects, we tested two different framings for our survey questions and collected responses from two different sources (n=2,092: 1,751 responses from Amazon Mechanical Turk and 455 from a demographically census-representative web-panel sample of U.S. internet users).

Our ultimate goal is to examine whether insights about our exemplar security behavior (software updating) derived from survey data match well with real-world results, and whether any deviation we observe is sufficiently systematic to be corrected in a straightforward manner. To this end, we quantify differences in reported and measured patching delay in response to the same update messages. We also examine whether features previously identified by prior work as important to update decisions — text of update message, length of update message, prior negative experiences with updates, and whether a restart is required — produce similar effects in both the survey and measurement data.

For the most part, we observe systematic, consistent differences between the measurement results and the survey results. For speed of updating, survey respondents tend to report faster update speeds than we observe in reality, and survey framing matters: respondents asked to make a recommendation to a friend advised updating immediately, respondents reporting on their own behavior said they would update within one week, and measurement data indicates that in reality most users updated within a few weeks.

We also observe systematic and consistent differences in the effect of high-level, user-specific factors – such as typical behavior and perception of risk – identified in prior work. For example, in both survey and measurement data, past tendency to update is significantly correlated with speed of applying a new update; however, survey data shows a medium effect size, while measurement data shows only a small effect.

However, we find no such systematic relationship for factors that require careful reading of update messages, such as the length of update message or whether they mention needing to restart. This may reflect that respondents are not reading carefully, or that they are not accurately assessing which features drive their real-world decisions.

Overall, our results suggest that some well-known concepts from survey methodology — systematic over-reporting of socially desirable behaviors and larger-than-reality effect sizes — also apply well in the security and privacy context. However, our findings that survey questions about specific message features may not work well and that demographically representative sample sourcing does not improve results, seem fairly specific to this context (and complement prior security-specific work [17, 53]). We conclude that certain kinds of survey self-report biases can be effectively corrected, but that for assessing detailed concepts like the specifics of messages, other approaches may be needed.

2 RELATED WORK

Here, we summarize prior work on user study biases in studies of security, as well as studies of other topics; we also briefly review the plethora of prior work on software updating behavior.

2.1 Evaluating Security User Studies

Real-World Behavior. A limited set of prior work compared real-world data to security user studies [16, 39]. Fahl et al. examined password-creation studies by comparing sets of passwords collected in an online and a laboratory study with real passwords belonging to the same participants for the same kind of services [16]. They found that 46% percent of participants produced data that matched the real-world. Removing participants who self-reported that they did not behave normally further improved results. Mazurek et al. also found that passwords created in online studies could serve as a reasonable proxy for real-world passwords created in similar conditions [39]. Both of these password validity studies involved laboratory-style methods, rather than surveys. Sotirakopoulos et al. compared survey responses on SSL warnings to laboratory observations, finding significant differences in results [60], while Akhawe et al. compare Sotirakopoulos et al.'s lab-study findings to the results of a field study, identifying significant differences [2]. Our work answers multiple open questions raised by these studies: exploring differences between survey and real-world observations, rather than lab observations; examining *why* response biases may occur; and examining multiple types of constructs, rather than a singular behavior such as creating a password or clicking through a warning.

Survey Framing. Additionally, prior work in security has examined the effect of role playing – asking the participant to imagine their own response vs. asking them to imagine someone else's response – on survey response, finding that role playing significantly alters survey results [3, 57]. However, neither of these prior studies was able to compare with "ground-truth" data of real-world behavior, thus limiting their ability to draw firm conclusions about accuracy differences based on framing; we address this in our work.

Survey Sample. Finally, three security and privacy studies have compared the results obtained by administering the same survey to

different samples [27, 53, 58]. The results of these comparisons suggest that MTurkers may express stronger privacy beliefs and more frequent reports of privacy or internet behaviors. More broadly, Redmiles et al. find that results from a crowdsourced sample are most accurate when considering younger and more educated users, but a demographically diverse web panel performed better for older and less educated users [53]. These prior studies, which focus on very general reports of behavior, do not examine factors that may be correlated with behavior or compare against any ground-truth measurement of behavior. In our work, we address these gaps, finding that the samples are relatively comparable, albeit with a difference in update frequencies and replicate similar tech-savviness effects.

2.2 Survey Bias Analysis in Other Fields

Prior work in survey methodology has explored reporting accuracy primarily for "sensitive questions" [51, 63, 64]. Sensitive questions typically ask about topics that are expected to be subject to social desirability biases, which lead respondents to answer as they think they should, rather than providing a truthful answer. Prior work in survey methodology has examined the effect of mode (telephone, mail, vs. web survey), sample (census-representative, crowdsourced, etc.), and survey design (different ways of framing survey questions, different interviewer demographics, etc.) on sensitive behavior reporting for school performance, crime, alcoholism, and smoking [8, 18, 29, 32, 33, 64].

These studies and a multitude of similar work led to the development of methodologies for more accurate measurements including the development of computer-assisted telephone interviewing systems, in which telephone survey respondents are transferred to an automated service to answer sensitive questions and list experiments, in which participants tally up a set of behaviors without explicitly designating which behaviors they do, and behavior prevalence is stastically imputed [10].

2.3 Prior Work on Updating Behavior

A large body of prior work has used measurement [4, 11, 13, 21, 41, 43, 54, 56, 59, 71] and user study approaches [15, 23, 25, 36–38, 40, 46, 62, 66–68] to measure and understand user's speed in updating their systems or software.

These works have identified a number of factors that may be related to updating speed, which we summarize here.

Risks. Negative experiences have been shown in self report work to drive users to not install manual updates or turn off auto-updating [36,67]. Prior self report work suggests that these negative experiences – with crashing, undesired features, etc. – inform an overall user perception of update risk, which is highly related to update decision-making. The effect of prior negative experiences has not been evaluated in measurement data, to our knowledge [38].

Costs. Gkantsidis et al. find in their measurement study that larger patches (those with greater filesize) are deployed more slowly [21], perhaps due to slowness of download for users. Relatedly, Mathur et al. find from a user study that "costs" such the "time it takes to install the update, whether a restart is required, and required space on disk" are related to users reporting not wanting to install updates, and note that these costs appear to be one of the main three reasons for update speed [38].

Message & Application Factors. Prior self-report work finds that users report choosing to update, or prioritizing certain updates, for a number of reasons, including because the update was about security [37, 66], the update was marked as critical [15, 23], was for an application they perceived as important [37,66] or from a vendor that they trusted [62]. Further, they report choosing not to update due to lack of understanding of the application that needs updating, and the introduction of undesirable features [67]. Measurement findings differ from these self-report results, however. Sarabi et al.'s measurement data analysis suggests that users' updating behavior can be summarized using a single-parameter geometric distribution and that updating speed does not depend on the type of improvements in new releases [56].

General User Tendencies. Finally, prior work using self-reports to measure updating behavior suggests that users may follow behavioral patterns, anchoring to their past behavior [25, 40, 46, 68]; this is a common phenomena shown also in psychological literature [61] and in observations of updating [56] and other security behaviors [50].

To maximize comparability with prior work, we draw our measurement data from Symantec WINE [12] , a measurement system that actively monitors users' behavior, and which has been used in multiple prior studies of updating behavior [43, 56].We collect our own survey data as no datasets with sufficient data to appropriately match the measurement data were available; to ensure comparability and generalizability we draw from survey questions used in prior work to develop our questionnaire.

3 RESEARCH QUESTIONS AND DATA SOURCES

To understand biases in self report data about digital security behavior, we conduct an in-depth comparison of empirical observations of host-machine updating behavior collected using the WINE system [12] (n=517,932) to survey data eliciting self-report responses (n=2,092) to the same update messages.

In particular, we address the following research questions:

RQ1: How well do self-reported security-behavioral intentions correlate with observed field data?

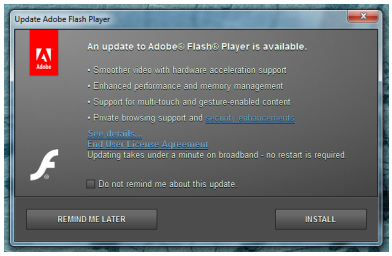
RQ2: How does framing the question in terms of the respondent's own behavior, as compared to behavior recommended for a friend, affect this correlation?

RQ3: How does sample source (i.e., demographic representativeness) affect these correlations?

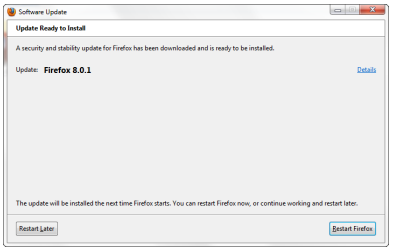
RQ4: How does the correlation between self-reports and measurement data differ for research questions relating to general perceptions and behaviors, as compared to research questions related to the update messages, which require respondents to carefully read specific, displayed information?

In this section, we connect these research questions to our data sources: the update messages for which we analyze behavioral and self-reported responses, the field measurement data we obtained, and the survey data we collected.

(a) Update message for Flash Player 10.0.45.2, which mentions only security and explicitly states that it does not require a restart.



(b) Update message for Flash Player 10.1.53.64, which mentions features and security and explicitly states that it does not require a restart.



(c) Update message for Firefox 8.0.1.4341, which states that a restart is required to install the update, and mentions both stability and security.

Figure 1: Examples of Update Messages

3.1 Update Messages

In order to compare the measurement and survey data, we want to contrast self-reported responses to a given update message to observed behavior when encountering the same message. To this end, we needed to find images for update messages in our field data.(See details of measurement data in Section 3.2 below.) Because neither our measurement dataset nor application release notes archive the images that were displayed to users when various updates became available, we instead searched for update messages by performing Google image searches and asking IT staff at two universities for any saved screenshots related to updates. In the end, we were able to obtain 11 messages for which we had patching records in our measurement dataset: six Adobe Flash messages, one Firefox message, two Adobe Reader messages, and two Opera messages. All messages were for updates released between 2009 and 2012 (in Section 5 we evaluate and discuss potential time confounds). Figure 1 shows three messages; Appendix B shows the remainder.

3.2 Measurement Data

We use patch deployment data sets from the Worldwide Intelligence Network Environment (WINE) [12], a platform for accessing Symantec field data for cybersecurity. WINE collects data from machines that have installed home (as opposed to corporate) versions of Symantec security products, and is designed to ensure that the available data is a representative sample of data collected by Symantec [12]. Symantec makes measurement data collected using WINE from 2008 to 2014 available to researchers.

Our dataset includes records of the timestamp when specific files first appear on a given machine. We use data from Nappa et al. [43] to map software version updates to specific file hashes, allowing us to identify when a particular software patch was installed. We can therefore calculate updating speed as the time interval between patch release time and installation time, for a given patch version and machine.

We also use WINE log data to measure features of individual hosts, such as their history of update responses, history of crashes for particular applications and for the entire system, and whether or not specific applications are installed.

Sampling Measurement Data. To obtain an appropriate subset of the measurement data, we selected only hosts for which we have a record that one of the 11 update versions we target was eventually installed. We then remove any machines whose local time is visibly incorrect: in particular, where the patch time is one day or more earlier than the actual patch release date. We retain only U.S. users, for ease of survey sample matching and reliability to findings from prior self-report work (nearly all of which were conducted with U.S. respondents).

Finally, we note that one machine can have multiple records in the data, if more than one of our eleven targeted updates was applied on the same machine. These repeated measures would complicate statistical analysis, particularly because we only have multiple records for a minority of hosts, so it would be difficult to account for them using standard methods. Instead, we randomly select only one of the available records for each host where multiple events were available. This random selection is performed last, after all other filtering steps, which selects 517,932 out of 730,270 update events that correspond to our 11 messages.

3.3 Survey Data

To compare with the measurement data, we collected self-report data about users' intended updating behavior using a between-subjects survey. Each survey began by showing the respondent exactly one of our 11 update messages; respondents were only shown update messages for an application which they reported either using or having on their device within the past 5 years. Section 5 provides more details on the demographic comparability of the survey and measurement samples.

The respondent then answered questions about how quickly they would apply the indicated update (RQ1) and then the reasoning behind their decision (RQ4). Appendix A shows the questionnaire.

Framing (RQ2). RQ2 addresses one possible source of potential discrepancy between survey results and real-world phenomena: social-desirability bias – which from respondents' beliefs about the proper or expected answers to survey questions – and personalization biases that arise from respondents' having difficulty accurately assessing their own behavior [19]. To investigate this, respondents were randomly assigned to one of two framing conditions: *self*, where they answer questions about their own intended behavior, or *friend*, in which they answer questions about what behavior they would suggest to a friend.

To measure self-reported updating speed, *self* respondents were told to "Imagine that you see the message below appear on your computer," and the update message image was displayed. They were then asked (on the same survey page) whether they would intend to update this application, with the following answer choices: "Yes, the first time I saw this message," "Yes, within a week of seeing this message," "Yes, within a few weeks of seeing this message," "Yes, within a few months of seeing this message," "No," and "I don't know."

In contrast, *friend* respondents were told to"Imagine that a friend or relative sees the message below on their computer and asks you for advice," and the update message image was displayed. These respondents were then asked (on the same survey page) how soon, if at all, they would recommend that their friend updated their application.

We hypothesized that asking about friends would provide respondents with a more neutral, less personal scenario. Asking about friends is a well-known tactic in behavioral economics and survey methodology for obtaining such normative judgements, and has previously been applied in human-centered security research [7, 19, 28, 44].

Recruitment (RQ3). To address RQ3, we collected responses to our survey using two sampling platforms: Amazon Mechanical Turk (MTurk) and Survey Sampling International (SSI).

Respondents from MTurk were invited to take a survey about online behavior, and were paid $0.50 for completing the brief (<5 min) survey. MTurk is known to produce demographically biased survey samples [26, 47, 55]; however, it is the most commonly used sampling platform in security research. In line with findings from prior work about response validity, we recruited only Turkers with 95% approval ratings [48].

Respondents recruited through SSI were sampled such that the demographic makeup of the respondent pool closely matched the demographics of the U.S. with regard to age, education, gender, race, and income (demographics for our SSI sample are shown in Appendix C). Such census-representative samples are expected to provide results more generalizable to the U.S. population [9]. SSI respondents took an identical survey to that shown to the MTurk respondents and were paid according to their agreement with SSI (compensation often takes the form of charity donations, airline miles, or cash).

We obtain a final survey sample of 2,092 respondents who use antivirus software and Windows computers (we refer to this dataset as the "full" survey dataset), which consists of 1,751 from Amazon Mechanical Turk (the MTurk dataset) and 455 from SSI (the SSI dataset).

Validity. To ensure that our survey was representative of surveys in the field, we drew our survey questions from prior work related to software updates [25, 37, 65, 66, 68], in some cases with slight modifications to specific questions. As described below, we selected and modified these pre-existing questions as needed to most closely match measurements available in the WINE data.

To maximize construct validity and ensure that our survey was easy for respondents to interpret, we conducted six cognitive interviews [49, 69] with a demographically diverse set of respondents. In these interviews we asked respondents to "think aloud" as they answered the survey questions and probed them on areas of uncertainty. We updated the survey after each interview and continued conducting interviews until areas of uncertainty stopped emerging.

4 EXPERIMENTAL APPROACH

Using the datasets described above, we developed experimental approaches to answer each of our research questions.

For all analyses, we use the updating speed measurement data and the main updating speed survey question defined in Section 3.3 above. We exclude any respondents who report that they would not install an update (n=138 in MTurk, n=64 in SSI) or that they do not know (n=45 in MTurk; n=19 in SSI), because we are unable to identify a parallel population in the measurement data.

Throughout our analysis, we apply Holm-Bonferroni correction as appropriate to account for multiple-testing effects [24].

4.1 RQ1–3: Comparing measurement and survey data

Our primary goal, encapsulated in RQ1, was to understand how well self-reported survey data can proxy for field measurements when considering users' security behavior. More specifically, we wanted to know whether, even if self-report data is not entirely accurate, it deviates systematically enough that it can still provide a useful understanding of end-user behavior. In the process, we compare across framing conditions (RQ2) and across sample sources (RQ3).

The answers to this updating speed question are thus treated as a 4-point Likert measurement. To align the survey answer choices with the measurement data we bin the measurement results to match the Likert responses: as soon as I see the message is equivalent to updating within 3 days, within a week is equivalent to updating between 3 and 7 days after the patch appears, within a few weeks is equivalent to updating between 7 and 30 days, and within a few months is equivalent to patching in 31 days or more.

To compare the update speeds observed in the measurement data and reported in the survey data, we use a X^2 proportion tests – which are robust to sample size differences – to compare updating speeds in the measurement and survey data, both over the full survey dataset and both conditions (RQ1), the full dataset by condition (RQ2), and by sample (RQ3). For the per condition and per sample comparisons, if the omnibus (e.g., friend vs. self vs. measurement) is

found to be significant, we conduct planned pair-wise comparisons: RQ2: friend vs. measurement and self vs. measurement on the full dataset; RQ3: MTurk vs. measurement and SSI vs. measurement, and a replication of the RQ2 analysis on the separated MTurk and SSI datasets, respectively.

4.2 RQ4: Comparing Question Types via Factors That Affect Updating

RQ4 investigates how the relation between self-report and measurement data is affected by the type of construct being measured.Within our exemplar context of software updates, we identified two types of constructs: general constructs, such as how often the respondent typically updates, or how often the respondent's computer typically crashes, and detailed constructs, such as self-reporting in the presence of a subtle experimental manipulation, such as the differences in the text of the update messages we tested.

For this investigation, we examine features that have been found in prior work to be relevant to update speeds and decision-making, and that were obtainable in our datasets:

- the **application** being updated;
- the **cost** of installing the update, in terms of whether it requires a restart;
- whether the update mentions **only security** (as opposed to other features) [1];
- the **length** of the message;
- the **risk** associated with the update, typically informed by the user's prior **negative experiences** with updating and stability;
- and the user's prior history of updating speed, which we refer to as **tendency to update**.

Table 1 summarizes how we instantiate these factors in each dataset, as well as which related work supports their inclusion.

The first several features — application being updated, whether a restart is required, whether security is the only feature mentioned, and message length — are determined by the update message under consideration. Table 2 summarizes the update messages we collected according to these features. Messages were considered to be security-only if they mentioned that the patch addressed security issues and made no mention of features or stability. For example, Figure 1a shows a security-only message, while the message in Figure 1b mentions both security and features. Message "cost" was characterized by whether the message mentioned requiring a restart (e.g., Figure 1a states that it requires no restart, while Figure 1c states that a restart is required). If restart is not mentioned in the message, then we consider it as "not required" since users are likely unaware of restart. Finally, message length was characterized as the number of words in the message.

We consider the first four features to be "detailed constructs," especially security only, restart, and the length of the message, which require respondents to be paying close attention to the displayed messages. The last two features: risk informed by prior experiences and general tendency to update, are "general constructs."

[1] All of the messages we collected mentioned security, thus we compare the effect of mentioning *only* security to mentioning both security and other enhancements, as prior work suggests that user may be wary of additional enhancements [66].

In order to isolate the effects of the detailed constructs as much as possible we identified sets of messages to compare:

- Application effects: we use the full dataset to compare effects among the four applications
- Cost effects: we compared the two Adobe Reader messages to each other, as one message mentioned a restart requirement and the other did not. (This is the only pair of messages with this within-application variation). The Reader messages were otherwise quite similar (same description of security and stability enhancements, same application), although the number of words in the messages did vary.
- Effect of message mentioning only security: we compared the six Flash messages to each other. Two of the six messages mention only security, while the other messages mention additional enhancements. Additionally, one of the security messages is the same length as a message that mentions security and features, allowing us to include message length in our model and control for this factor. All mention that you do not need to restart.
- Message length: we also use the Flash messages, as they have the largest variation in length and are similar on all other features, as just described.

The remaining features — update risk and tendency to update — are user-specific, and thus were inferred from measurement results and survey responses. For these two features, we compare messages within applications, to control for potential application effects, and between applications, to control for covariance with other features.

Inferring Features from Measurement Data.

Risk Metrics. We characterize update risk in terms of a user's prior experience with overall stability, as well as specifically how updates affect stability. To measure this in the measurement data, we use WINE's *binary stability* dataset, which records both system crashes and application crash/hang events.

We define four risk metrics:

- Average weekly frequency of system crashes and hangs during the year before the user installs the target patch.
- Average weekly frequency of crashes and hangs for the target application during the year before the user installs the target patch.
- Average change in the number of system crashes and hangs between the week before and the week after a new patch was installed. Averaged over all updates of the target application installed in the year prior to installing the target patch. If the average is positive, we consider this an increase in system crashes post-update.
- Average change in the number of crashes and hangs between the week before and the week after a new patch was installed. Averaged over all updates of the target application installed in the year prior to installing the targeted patch. if the average is positive, we consider this an increase in application crashes post-update.

The former two metrics are used to capture the overall crash tendency of the system or application, while the latter two are used to capture the user's past negative experience in system/application crashes when they update the applications.

	Feature	Measurement	Survey	Prior Work
Detailed Constructs	Application	Source application.	Same as measurement.	[37, 38, 62, 66]
	Update Cost	Whether the update mentions requiring a restart.	Same as measurement.	[38]
	Security-Only	Message mentions security but not features or stability.	Same as measurement.	[15, 23, 37, 62]
	Message Length	Number of words in message.	Same as measurement	[62]
General Constructs	Update Risk	Negative experiences characterized by two different features: average number of application and system crashes per week over past one year and the average change in crashes for the application and the overall system before and after the past updates within one year.	Responses to four survey questions about experiences with application and system crashes in general and related to updates of this application.	[22, 37, 38, 67]
	Tendency to Update	Mean updating speed for prior patches from the same application.	Responses to the following survey question: "In general, how quickly do you install updates for applications on your computer or for your computer itself (e.g., the computer operating system)?"	[54, 56]

Table 1: Summary of the factors considered in our models, how they were operationalized in each dataset, and from what related work they were drawn.

Version	Application	Release date	Security Only	Requires Restart	No. of Words	Risk Metrics Available
10.0.22.87	Flash	2/24/2009	✓		57	
10.0.45.2	Flash	2/11/2010	✓		57	
10.1.53.64	Flash	6/3/2010			48	
10.2.152.26	Flash	2/8/2011			55	✓
10.3.181.14	Flash	5/12/2011			50	✓
11.0.1.152	Flash	10/4/2011			57	✓
9.3.2.163	Reader	4/13/2010		✓	35	
9.5.1.283	Reader	4/10/2012			23	✓
10.61.3484.0	Opera	8/9/2010		✓	80	
11.64.1403.0	Opera	5/10/2012		✓	80	✓
8.0.1.4341	Firefox	11/22/2011		✓	45	✓

Table 2: Summary of update messages.

For ease of analysis, we center and normalize the raw crash counts. This data was only collected starting in 2011. Thus, we are only able to obtain stability features for the 5 update messages, 30,623 users, as indicated in Table 2.

General Tendency to Update. We define general tendency to update as the average update speed for all versions of a given application prior to the targeted update. Let V_N be the selected version, such that $\{V_1, V_2, \cdots, V_{N-1}\}$ are the prior versions. $D(v, m)$ is the speed of updating version v for machine m. Then the tendency to update for machine m is calculated as $\frac{1}{N}\sum_{n=1}^{N} D(V_n, m)$.

Inferring Features from Survey Data. Risk Metrics. To assess perceived prior negative experience with updating — specifically around crashing risk — we asked respondents a series of four questions. The first two were "Over the past year, how frequently do you feel like [application for which patch message is shown] has frozen (e.g., hang) or crashed?" and "Over the past year, how frequently do you feel like any application on your computer or your computer itself crashed?" Both questions provide answer choices on a four point scale: "Less than once a week", "At least once a week but not more than three times a week," "At least three times a week but not more than five times a week," and "Five times a week or more."

We also asked, "Over the past year, have you noticed that updating [application for which patch message is shown] changes how frequently it freezes (e.g., hangs) or crashes? and "Over the past year, have you noticed that updating [application] changes how frequently any application on your computer or your computer itself crashes?" These questions had the following answer choices: "Yes, my computer crashes more after I update," "Yes, my computer

crashes less after I update," and "No, updating [application] has no impact on how frequently my computer crashes."

General Tendency to Update. We assessed tendency to update by asking respondents "In general, how quickly do you install updates for applications on your computer or for your computer itself (e.g., the computer operating system)?" with answer choices: "As soon as I see the update prompt," "Within a week of seeing the prompt," "Within a few weeks of seeing the prompt," "Within a few months of seeing the prompt," "I don't install updates that appear on my computer," and "I don't know." This question was constructed to be similar to a question asked by Wash and Rader [68].

Statistical Modeling to Compare Effects of Relevant Factors. To compare the effect of factors suggested by prior work as related to people's updating behavior between the survey and measurement data, we construct ordinal logistic regression models, which accommodate Likert outcome variables such as our measure of update speed [45].

We construct one set of models to examine the detailed constructs; these models include all survey and measurement data for the messages being considered. We also construct a second set of models to examine the risk metrics, as these metrics were only available in the measurement data for five of our 11 messages. We refer to these as the *detailed* and *risk* models respectively.

To best isolate the effects of the individual constructs, we use a hierarchical modeling approach. We construct a baseline model and then add feature sets so we can examine their impact in isolation [70]. For both detailed and risk model sets, our baseline models contain a single feature: general tendency to update.[2] We then add sets of features to examine the constructs of interest. Specifically, for the detailed constructs, we construct the following models:

- Across All Applications (Construct of Interest: Application)
  - Baseline: General Tendency (ordinal DV, four-point scale)
  - General Tendency and Application (categorical DV, Flash is the baseline)
- Reader (Construct of Interest: Cost)
  - Baseline: General Tendency
  - General Tendency and Cost (boolean DV, whether the message mentioned a Restart)
- Flash (Constructs of Interest: Length, Security)
  - Baseline: General Tendency
  - General Tendency and Length (continuous DV, number of words)
  - General Tendency and Security-Only (boolean DV, whether the message mentioned anything other than security)
  - General Tendency, Length, Security-Only: constructed to control for covariance between length and security-only

For the risk model set, we construct models across all applications. The baseline model for each consists of general tendency to update, and for the survey data, sample source. The risk model for each consists of the four risk factors: frequency of system and application crashes (ordinal DVs) and existence of an increase in system crashes and application crashes post-update (boolean DVs), and controls for general tendency to update and application.

[2]To further address RQ3 and control for sample effects, we also include survey sample source as a factor in the survey models.

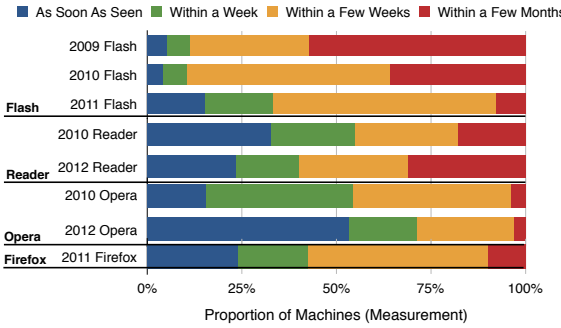


Figure 2: Measurement update speed by year and application.

To ensure model validity, we performed backward AIC selection on the baseline model in each case (retaining the baseline factors in all cases). For each model we report the log-adjusted regression coefficients, known colloquially as *odds ratios* (O.R.s), and indicate significance (p-values < 0.05). To further examine RQ3, we include the sample source (MTurk or SSI) as a factor in all of our survey regression models.

5 DATASET COMPARABILITY AND LIMITATIONS

We next discuss threats to validity related to our datasets and experimental approach.

Sampling. The majority of WINE hosts are located in the United States. For consistency, we sample only U.S. survey respondents and include only U.S. WINE hosts in our analysis. Additionally, we recruited only survey respondents who use Windows devices, as all WINE hosts are Windows. Finally, we conduct our modeling using only those survey respondents who reported using antivirus software, in order to closely mirror the measurement data (eliminates 416 respondents).

Differences in Timing. One crucial confounding factor in our analysis is the difference in time between when the measurement and survey data were collected. The measurement data available from Symantec was collected from 2009 to 2013, while the survey data was collected in 2018. We attempt to quantify the importance of this time delay by investigating how time affects each dataset.

To understand how updating frequency in the real world has changed over time, we tested the effect of time in measurement data. The effect is significant, but of small size (X^2=72412, $p < 0.001$, V=0.181). Additionally, although time is a significant factor, the effects are not in a consistent direction for each application (Figure 2): Opera is updated significantly faster in 2012 than in 2010, while Reader is updated slower in 2012 than in 2010; Flash is updated slower in 2010 than in 2009 and faster in 2011 than in 2010. Given the inconsistencies in these time biases, we do not suspect that time will create systematic biases in our results.

To evaluate whether self-reports about updating frequency have changed over time, we compared our results with the oldest work

with comparable data [68]. Wash and Rader conducted a census-representative survey of 2000 people, in which they asked respondents to report their general updating frequency, also on a five-point Likert scale. Using a Mann-Whitney U test, standard for Likert scale data [34], we find no significant difference in updating frequencies between their results and our survey.

Thus, while time confounds are possible,we hypothesize that they are unlikely to be so significant as to invalidate our results. Taking into account that real-world data of the size and quality provided by WINE is rarely available, we argue that our analysis can provide many valuable insights despite this potential confound.

Machines vs. Users. The measurement data measures machines, while our survey data measures users. For our analysis, we assume that there exists a one-to-one mapping in the measurement data between machine and user, but it is of course possible that one user manages multiple machines. Although we cannot determine how many of these cases may exist, we believe the effect of this should be relatively minimal given the large size of our dataset. Additionally, it is possible that some hosts in the measurement data are not personal computers, but rather corporate-managed machines. However, machines managed by large organizations typically use an enterprise Symantec product and therefore are not recorded by WINE. The percentage of corporate managed machines *not* using the enterprise software is anticipated to be quite low [43].

Self-Report Biases. As is typical of survey studies, self-report methodologies have a number of biases and limitations. For example, social-desirability bias, where people report what they think will make them seem most responsible or socially desirable [31]. However, it is important to note that in this study, we wanted specifically to compare the survey results, which are inherently biased in some ways, with the measurement data, which is inherently biased in other ways. We apply best practices for extensively pre-testing our survey, randomizing answer choices, and placing demographic questions last. Biases which are not mitigated by these steps are therefore a key aspect of our results.

Generalizability. Finally, our work has three potential threats to generalizability. First, we sample only antivirus users. However, as antivirus users are estimated to make up at least 83% of the online population [52], and it is unlikely to be able to draw a truly random sample of log data, we consider this population to cover the population of internet users relatively well. Second, we examine only software updating behavior. As such, we can indeed only hypothesize about similar bias effects in other security behaviors. We opt to provide detailed, in-depth analysis of a single behavior rather than more cursory analysis of multiple behaviors; this follows the approach of nearly all prior work in survey methodology, which tends to consider one behavior (e.g., smoking) at a time to enable thorough analysis. Third and finally, automatic updates have been growing in adoption since the time when our measurement data was collected. However, automatic updates may still offer users a choice to delay and require user-controlled application restarts. Thus, users still must make time-related software update choices, even if they may not have the option to chose *whether* to update.

	Comparison	X^2	p-value
RQ1	Measurement vs. Survey	103630	< 0.001
RQ2	Omnibus: Measurement vs. S: Self vs. S:Friend	103730	< 0.001
	Measurement vs. S: Friend	103310	< 0.001
	Measurement vs. S: Self	102850	< 0.001

Table 3: X^2 tests comparing the speed of updating reported in the surveys (S) with the speed of updating observed in the measurement data (WINE).

6 RESULTS

Below, we detail our findings by research question.

6.1 RQ1–3: Speed, Framing, and Sampling

We start by examining self-report biases in estimating update speed.

RQ1: Updating Speed.. To obtain an overall comparison between survey and measurement data, we compare the full survey dataset (which consists of responses from both the MTurk and SSI survey samples, across both framing conditions) with the measurement data. We find a significant difference ($X^2 = 103630$, $p < 0.001$) between the combined survey responses and the measurement data: the median update speed in the survey data is "Within a week" (Likert value 2), while the median speed in the measurement data is "Within a few weeks" (Likert value 3).

RQ2: Survey Framing.. To examine the effect of the survey framing, we separately compare the *friend* and *self* conditions (described in Section 4.1) to each other and to the measurement data. (This comparison also combines both sample sources.) We find significant and consistent differences in outcomes between our two survey framings (Table 3): median update speed in the Friend condition is "Immediately" (Likert value 1), compared to a median of "Within a Week" (Likert value 2) in the Self condition and "Within a Few Weeks" (Likert value 3) in the measurement data.

RQ3: Sample Comparison. We also compare update speeds by survey sample. We find a significant difference between update speeds reported in the MTurk sample and those reported in the SSI sample ($X^2 = 1256.3$, $p < 0.001$). SSI respondents report a median update speed of "Immediately" (Likert value 1) compared to MTurk respondents who report a median speed of "Within a Week" (Likert value 2). Finally, the effect of the survey framing on the survey results for both samples is significant (MTurk: X^2=40.19, $p < 0.001$; SSI: X^2=16.5, $p = 0.009$).

Summary: systematic over-reporting of update speed in surveys; survey framing matters. Figure 3 summarizes the results of our comparison of updating speeds reported in the two different survey framing conditions (friend vs. self) and samples (MTurk vs. SSI) against the measurement data. Overall, we find that survey respondents systematically report faster update speeds compared to the measurement data, and this bias is affected by survey framing. Finally, we observe reporting speed differences between the two

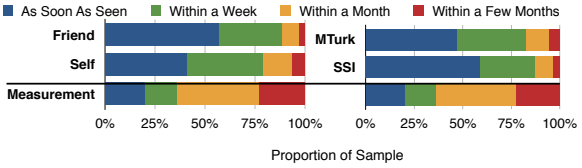


Figure 3: Comparison of self-reported update speeds by framing condition (left, full survey dataset) and survey source (right, per source over both framing conditions) to the measurement data.

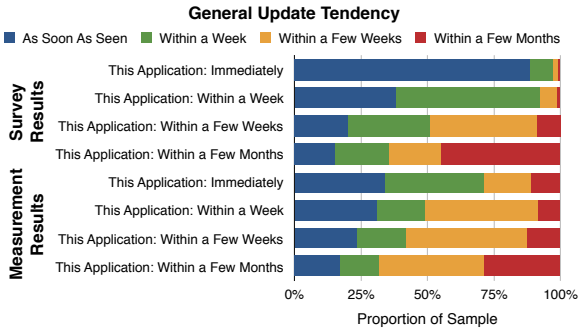


Figure 4: General tendency to update vs. update speed for a specific message in the survey (top) and measurement (bottom) data.

survey samples: Perhaps surprisingly, the responses of the MTurk participants are somewhat closer to the measurement data than are those of the census-representative participants.

6.2 RQ4: Factors Affecting Update Speed

Next we examined the impact of various factors that prior work suggests may affect update speeds. To do so, we construct hierarchical regression models on both the survey and measurement datasets to compare variables of interest while controlling for other potentially relevant factors, as described in Section 4.2. In interest of brevity, we summarize the results here, and include in Appendix D.1 tables of regression results for all models constructed.

We detail our results by factor: general tendency to update, crash risk, and then the four message features. Finally, we review sample effects related to these factors (RQ3).

6.2.1 General Tendency to Update. In regression models for both the measurement and survey data, we find a significant relationship between general tendency to update and update speed for all applications. People who typically update more quickly, or report typically updating more quickly, are also more likely to report updating (or actually update) a given application faster. This is illustrated in Figure 4). This significant relationship holds in every model we test, for survey and measurement, both for the full dataset and for individual applications. However, the effect is larger in the survey data than in the measurement data: the odds ratios (O.R.s) for the survey models average 5.85 (SD=0.834), compared to 1.55 (SD=0.220) for the measurement data.

Summary: General tendency to update is significant in both datasets, but the effect is larger for survey data. In sum, we observe that we would draw similar conclusions about general tendency to update being an important covariate from either the survey or the measurement data, but the effect sizes in the survey data are consistently larger than those in the measurement data.

6.2.2 Risk. We consider four risk metrics: average frequency of system and application crashes, and increases in system and application crashes after updating. In the measurement data, we observe mixed results regarding the relationship of these risk metrics to updating speed, finding a lack of consistency in which risk metrics, if any, are related to updating behavior; especially when controlling for other covariates. The relationship between prior negative experiences and updating speed was previously unstudied in measurement data.

In regression models controlling for general tendency to update and for the application being updated, we find in the measurement data that more frequent system crashes are associated with slower updating speeds (O.R.=1.03, $p = 0.005$), while increased crashes after prior updates are associated with faster updating speeds (O.R.=0.89, $p = 0.026$). These effects are fairly small. In contrast, in the survey data, none of the risk metrics show a significant relationship to updating speed.

To see if the discrepancy in results may have been caused by issues of respondent quality, we reconstruct our survey regression models using a smaller dataset of only "high-quality" survey responses. We borrow this approach from Fahl et al., who found that user study data more closely matched real-world data when filtering out low-quality responses [16]. In our context, we define low-quality responses as those who gave nonsensical answers: those who cited lack of restart as a reason to install an update message, but who saw a message did in fact require a restart (and reciprocally, those who cited needing to restart as a reason not to update, but who saw a message that did not require a restart) and those who cited like or dislike of features as a reason for installing, or not installing, but who in fact saw a message that mentioned only security (see Appendix D.2 for more detail). Examining the regression models built on this "filtered" survey dataset (n=981), we find significant effects, in the same directions as in the measurement data, albeit with larger O.R.s: perceived average number of system crashes (O.R. = 1.76, $p = 0.044$) and perceived change in crashes of the given application (O.R. = 0.53, $p = 0.440$) are related to self-reported update speed.

Summary: Risk effects replicated in survey data after filtering. In sum, we observe a small but significant relationship between update speed in response to a particular message and crash risk factors in the measurement data. After filtering for respondent quality, we observe a similar effect in the survey data.

6.2.3 Message Features. We compare the effects of four features related to the message text: the *application* being updated, the *cost* of installing the update (whether it requires a restart), the *length* of the update message, and whether the message mentions *only security* or also other features or stability enhancements.

Application. To examine the effect of the application on our results, we construct models over the full dataset, with application as a covariate. We find that the application is significantly related to the speed of updating in both the survey and the measurement data. The regression results for the measurement data show that Flash is updated more slowly as compared to Firefox (O.R.=0.66, $p < 0.001$) and Adobe Reader (O.R.=0.63, $p < 0.001$), and much more slowly than Opera (O.R.=0.29, $p < 0.001$). In the survey data, the overall effect is slightly smaller, but still significant: Firefox and Reader are have faster reported update speeds than Flash (O.R. = 0.82, $p = 0.048$; O.R.=0.81, $p = 0.007$). The survey model shows no significant result for Opera, however.

Cost: Reader. To examine the effect of mentioning a restart requirement (implicitly suggesting a time or effort "cost" to the user) in update messages, we compare two Adobe Reader messages. We find that the message that mentions a required restart is updated more slowly in the measurement data than the message that does not mention such a cost (O.R. = 0.53, $p < 0.001$ in a regression model controlling for general tendency). In the survey results, this effect is not mirrored.

Length: Flash. We compare the six Flash update messages to examine the impact of message length. In the update data, message length has a significant, albeit small effect on update speed: he length of the update message is significant both in the model that controls only for general tendency (O.R.=0.98, $p < 0.001$) and the model that also controls for mentioning security only (O.R.=0.93, $p < 0.001$); there are no significant effects in the survey data.[3]

Mentions Only Security: Flash. Finally, the measurement data shows that users who saw one of the Flash messages that only mentioned security vs. mentioning security and features or stability improvements updated faster, even when controlling for the user's typical update frequency (O.R. = 3.33, $p < 0.001$) and typical update frequency as well as message length (O.R. =4.54, $p < 0.001$). The survey data does not mirror this effect.

Filtering Respondents and Internal Consistency. We reconstructed each of the above models for message features using only the filtered subset of high-quality respondents (as described in Section 6.2.2 above. This approach did not produce any improvements in matching significant effects seen in measurement data.

To further investigate, we examined the *internal consistency* of the survey responses: how well users' responses about why they would (not) choose to install or recommend an update matched the actual properties of the messages they saw. Appendix D.3 details this answer-choice consistency mapping and results in table format.

We find that for the most part, reasons for updating that mentioned specific message properties were unrelated to the actual properties of the assigned message. Specifically, self-reports about update motivation related to a new version having features the user would want were not related to whether the update message mentioned features in addition to security (X^2=4.72, $p = 0.067$). Similarly, reports about not wanting to update because of the new version having features the user would *not* want were also not related to whether the update message mentioned features in addition to security (X^2=0.050, $p = 0.823$) Reports of not wanting to update because of needing to restart or because of time constraints (e.g., costs) were not related to the update message mentioning a restart (X^2=0.917, $p = 0.384$). On the other hand, participants who reported wanting to install or recommend an update because it looked fast or did not require a restart were more likely to have seen a message that did not mention a restart (X^2=6.39, $p = 0.024$). Figure 5 summarizes these results.

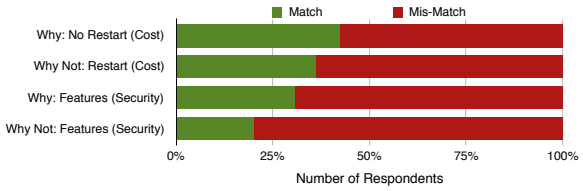


Figure 5: Comparison of the internal consistency of survey responses related to two of the three message features.

The application being updated, however, seems to be more salient than other message properties. Reporting that you would update because the given application was important (X^2=38.2, $p < 0.001$), or would not update (X^2=11.8, $p = 0.019$) because it was unimportant both varied significantly based on the queried application.

RQ3: Survey Sample Effects. We note that all survey regression models controlled for sample source. When looking at the full dataset, the baseline model shows no effect from sample source, but controlling for application type shows that MTurk respondents updated significantly more slowly (OR=1.30, $p < 0.001$) than SSI respondents. This effect is also seen in the Flash-only models.

Summary: Survey respondents inattentive to most message features. Overall, we observe small but significant effects in the measurement data for all message-related factors. However, we only observe application-related effects — not more detailed message-related effects — in the survey data. Internal consistency checks suggest that this may relate to survey respondents not noticing these specific details in the update messages.

7 DISCUSSION AND TAKEAWAYS

Below, we summarize our results, review our findings in context of prior work on updating behavior, and address implications for future use of survey studies in digital security.

7.1 Summary of Results

Overall, we find that surveys appear to closely mirror the measurement data, albeit with systematic biases, for our general constructs (response to a given updating message, general tendency to update, and — with dataset filtering — risk metrics) and for the most general

[3]We could not control for the other message feature, restart, because no Flash messages mentioned a restart requirement.

of our detailed constructs (application), but not for our detailed constructs related to message text (mentioning only security, message length, and mentioning a restart requirement).

We identify consistent biases between the measurement and self-report data for intended update speed overall, as well as the *effect* of general updating tendency on updating speed. In both cases, the survey data produces results that are more extreme: survey respondents are over-optimistic about how quickly they would update in response to an update message, and the survey results show a larger effect from general update tendency than the measurement results. After filtering the dataset to remove inattentive respondents, we observe similar, but again larger, effects from the two risk factors that are significant in the measurement data.

In sum, survey respondents over-report "good behavior" compared to measurement data, and recommend this behavior to friends even more strongly. Understanding this systematic "do as I say, not as I do" effect, which may be an extension of social desirability bias, or optimism bias – in which people think they are better than their friends (and therefore need less help) – can help to properly interpret survey results.

In contrast, we find no similarities between the survey results and the measurement results for any "detailed" feature except application. Filtering does not improve these results. We hypothesize that this is because respondents are less attentive and more inconsistent in their perceptions of message details than for general constructs. This hypothesis is supported by our finding that there is low internal consistency in the survey reports about these particular features. Further, we note that the internal consistency results for the "general constructs" are contrastingly high: For example, reporting that you would update an application because you "always update" or that you would not because you "rarely update" were both significantly related to the user's later, self-reported general tendency to update (always update: X^2=141.2, $p < 0.001$; rarely update: X^2=7.77, $p = 0.042$).

We offer two possible hypotheses for these results: (a) survey respondents may be exhibiting "satisficing" behavior – a well studied phenomenon in survey response in which respondents pick the minimum acceptable answer, without paying close enough attention to surface the true answer [30], and/or (b) the salience of such factors in reality cannot be replicated in a survey – for example, respondents are already being paid for their time, so the cost of reading a longer message may not matter to them.

Finally, we investigate the impact of the sample source (census-representative web panel vs. crowdsourced) on the relation between self-report data and measurement data. We find that crowd- sourced responses, which are cheaper and easier to obtain, are also a somewhat better match for the measurement data. Sample source effects are significant for updating speeds, but have less effect on results related to construct effects.These results complement prior results comparing samples within survey data [27, 53, 58], which suggest tech-savviness may matter more than demographics for security and privacy user studies.

7.2 Replication and Contrast with Prior Work on Updating Behavior

Our findings offer replication of the plethora of prior work on software updating, enriching the current body of knowledge on user patching delay, which can inform the increasingly relevant issue of restart delay for automatic updates [36].

Our results showing that people systematically report that they would recommend a friend update faster than they would intend to update themselves, suggest that people know they should update their devices. Additionally, our work largely replicates findings from prior self-report and measurement studies that risk, cost, message and application factors all affect update speed.

Risk. Our work is the first measurement study to evaluate the effect of risk — defined as prior negative experiences with crashing and as typical frequency of system and application crashing — on updating speed. We find a relatively small effect for frequency of system crashes in the measurement data, and no effect for the other two message features. Thus, we do not fully replicate strong findings regarding negative experiences from prior work [36, 67]. This may be due to a mismatch between our measurement of risk and users' perceptions of it.

Cost. Our findings complement prior findings from measurement studies, which suggested that the size of the update message was a cost that slowed updating speeds; we conclude that restarts are also a cost that deter users [21].

Message & Application Factors. Our measurement results confirm findings from prior self-report work [37, 62, 66] that the application affects updating speed. Additionally, we confirm results from prior qualitative studies that longer messages slow updating speed [38]. We complement findings from prior work that suggest security-related messages may be updated especially quickly [37, 66], finding that, when comparing amongst messages that all mention security, those that mention *only* security are updated $3-5\times$ as fast as those that mention security and features or stability enhancement. We hypothesize that our self-report results may not match this prior work because qualitative studies address general experiences, and may also overcome inattention problems.

Our measurement results on these features *contrast* with prior measurement results, which suggested that updating speed was related only to general updating tendency. Our models that also include message attributes fit the measurement data significantly better (log-likelihood test of model fit on Flash dataset: $p < 0.001$) than our baseline model, which, like the Sarabi et al. single factor model, contains only general updating tendency.

General Tendency to Update. Finally, we find a consistent effect between general tendency to update and speed of applying a new update. This finding replicates prior results suggesting that users anchor to typical behavior [25, 40, 46, 50, 56, 68], and is among the strongest effects we observe.

7.3 Moving Forward with Security User Studies

Our results imply that we should consider security user studies in terms of the types of constructs they evaluate.

Filter and Weight Survey Data for General Constructs. Our findings for *general constructs* align with prior work on password behavior, another general construct, showing that filtering out low

quality responses can produce more consistent matching between user study and observation data [16]. Additionally, the fact that the survey and measurement data in our study find the same significant effects for general constructs suggests that insights from such studies may be meaningful in the field. Because these effects are systematically larger in the survey data, it may be possible in future work to produce a statistical weighting procedure [5] to correct for the systematic bias in effects observed in survey data, assuming our results are replicated for other behaviors. Alternative survey approaches, such as list experiments [10], can also help to compensate for these biases.

Consider Alternate Methodologies for Studying Detailed Constructs. While the application effects observed in the measurement data were replicated in the filtered survey data, the effects were smaller than in the measurement data, and none of the effects for the other detailed constructs were replicated. These inconsistencies align with findings from prior work showing inconsistencies between survey results and real-world behavior [2], and may explain why prior work based on survey results has suggested using rather extreme variants [6] to attract user attention. In concert, these findings suggest that capturing user attention in surveys — at least attention for message details — may be more difficult than in real life. Thus, we suggest that future work explore new survey designs that attempt to improve attention capture, such as employing more interviewer-facilitated studies. Researchers can also look toward using A/B tests, field observations, or lab-observation hybrids such as the Security Behavior Observatory or Phone Lab [20, 42].

Future User Studies on the Validity of Security Measurement. Finally, our work lays a foundation for future user studies exploring the consistency of security measurements. Future work in this direction may include user studies similar to ours, which may seek to vary the choice of sample (e.g., exploring Prolific as an alternative to MTurk) and survey questions and constructs measured.

ACKNOWLEDGEMENTS

This research was partially supported by the National Science Foundation (grant CNS-1464163). Elissa M. Redmiles acknowledges support from the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1322106 and from a Facebook Fellowship.

REFERENCES

[1] American community survey 5-year estimates, 2016.
[2] D. Akhawe and A. P. Felt. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *USENIX security symposium*, volume 13, 2013.
[3] H. Almuhimedi, A. P. Felt, R. W. Reeder, and S. Consolvo. Your reputation precedes you: History, reputation, and the chrome malware warning. In *Symposium on Usable Privacy and Security (SOUPS)*, volume 4, 2014.
[4] W. A. Arbaugh, W. L. Fithen, and J. McHugh. Windows of vulnerability: A case study analysis. *IEEE Computer*, 33(12):52–59, 2000.
[5] P. P. Biemer and S. L. Christ. Weighting survey data. *International handbook of survey methodology*, pages 317–341, 2008.
[6] C. Bravo-Lillo, S. Komanduri, L. F. Cranor, R. W. Reeder, M. Sleeper, J. Downs, and S. Schechter. Your attention please: designing security-decision uis to make genuine risks harder to ignore. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, page 6. ACM, 2013.
[7] F. Carlsson. Design of stated preference surveys: Is there more to learn from behavioral economics? *Environmental and Resource Economics*, 46(2):167–177, 2010.
[8] J. M. Converse and S. Presser. *Survey questions: Handcrafting the standardized questionnaire*, volume 63. Sage, 1986.
[9] M. P. Couper. Web surveys: A review of issues and approaches. *The Public Opinion Quarterly*, 64(4):464–494, 2000.
[10] E. Coutts and B. Jann. Sensitive questions in online surveys: Experimental results for the randomized response technique (rrt) and the unmatched count technique (uct). *Sociological Methods & Research*, 40(1):169–193, 2011.
[11] T. Dübendorfer and S. Frei. Web browser security update effectiveness. In *International Workshop on Critical Information Infrastructures Security*, 2009.
[12] T. Dumitras and D. Shou. Toward a standard benchmark for computer security research: The Worldwide Intelligence Network Environment (WINE). In *Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, 2011.
[13] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, and et al. The matter of Heartbleed. In *Internet Measurement Conference*, 2014.
[14] W. K. Edwards, E. S. Poole, and J. Stoll. Security automation considered harmful? In *Proceedings of the 2007 Workshop on New Security Paradigms*, pages 33–42. ACM, 2008.
[15] M. Fagan, M. M. H. Khan, and R. Buck. A study of users? experiences and beliefs about software update messages. *Computers in Human Behavior*, 2015.
[16] S. Fahl, M. Harbach, Y. Acar, and M. Smith. On the ecological validity of a password study. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, page 13. ACM, 2013.
[17] A. P. Felt, A. Ainslie, R. W. Reeder, S. Consolvo, S. Thyagaraja, A. Bettes, H. Harris, and J. Grimes. Improving ssl warnings: Comprehension and adherence. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2893–2902. ACM, 2015.
[18] M. Fendrich and C. M. Vaughn. Diminished lifetime substance use over time: An inquiry into differential underreporting. *Public Opinion Quarterly*, 58(1):96–123, 1994.
[19] R. J. Fisher. Social desirability bias and the validity of indirect questioning. *Journal of consumer research*, 20(2):303–315, 1993.
[20] A. Forget, S. Komanduri, A. Acquisti, N. Christin, L. F. Cranor, and R. Telang. Security behavior observatory: Infrastructure for long-term monitoring of client machines. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA PITTSBURGH United States, 2014.
[21] C. Gkantsidis, T. Karagiannis, P. Rodriguez, and M. Vojnovic. Planet scale software updates. In *ACM SIGCOMM Computer Communication Review*, 2006.
[22] J. Gray. Why do computers stop and what can be done about it? In *Symposium on reliability in distributed software and database systems*, pages 3–12. Los Angeles, CA, USA, 1986.
[23] M. Harbach, S. Fahl, T. Muders, and M. Smith. Towards measuring warning readability. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 989–991. ACM, 2012.
[24] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.
[25] I. Ion, R. Reeder, and S. Consolvo. "...no one can hack my mind": Comparing expert and non-expert security practices. In *Eleventh Symposium On Usable Privacy and Security*. USENIX Association, 2015.
[26] P. G. Ipeirotis. Demographics of mechanical turk. 2010.
[27] R. Kang, S. Brown, L. Dabbish, and S. Kiesler. Privacy attitudes of mechanical turk workers and the us public. In *Symposium on Usable Privacy and Security (SOUPS)*, volume 4, page 1, 2014.
[28] P. G. Kelley, L. F. Cranor, and N. Sadeh. Privacy as part of the app decision-making process. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3393–3402, New York, NY, USA, 2013. ACM.
[29] F. Kreuter, S. Presser, and R. Tourangeau. Social desirability bias in cati, ivr, and web surveysthe effects of mode and question sensitivity. *Public opinion quarterly*, 72(5):847–865, 2008.
[30] J. A. Krosnick. Response strategies for coping with the cognitive demands of attitude measures in surveys. *Applied cognitive psychology*, 5(3):213–236, 1991.
[31] J. A. Krosnick. *Handbook of Survey Research*. 2010.
[32] I. Krumpal. Determinants of social desirability bias in sensitive surveys: a literature review. *Quality & Quantity*, 47(4):2025–2047, 2013.
[33] M. W. Link and A. H. Mokdad. Effects of survey mode on self-reports of adult alcohol consumption: a comparison of mail, web and telephone approaches. *Journal of Studies on Alcohol*, 66(2):239–245, 2005.
[34] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.
[35] A. C. Marcus and L. A. Crane. Telephone surveys in public health research. *Medical care*, pages 97–112, 1986.
[36] A. Mathur and M. Chetty. Impact of user characteristics on attitudes towards automatic mobile application updates. In *Symposium on Usable Privacy and Security (SOUPS)*, 2017.
[37] A. Mathur, J. Engel, S. Sobti, V. Chang, and M. Chetty. " they keep coming back like zombies": Improving software updating interfaces. In *SOUPS*, pages 43–58, 2016.
[38] A. Mathur, N. Malkin, M. Harbach, E. Peer, and S. Egelman. Quantifying users? beliefs about software updates.

[39] M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur. Measuring password guessability for an entire university. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 173–186. ACM, 2013.
[40] A. Möller, F. Michahelles, S. Diewald, L. Roalter, and M. Kranz. Update behavior in app markets and security implications: A case study in google play. In *Research in the Large, LARGE 3.0: 21/09/2012-21/09/2012*, pages 3–6, 2012.
[41] D. Moore, C. Shannon, and K. C. Claffy. Code-red: a case study on the spread and victims of an internet worm. In *Internet Measurement Workshop*, 2002.
[42] A. Nandugudi, A. Maiti, T. Ki, F. Bulut, M. Demirbas, T. Kosar, C. Qiao, S. Y. Ko, and G. Challen. Phonelab: A large programmable smartphone testbed. In *Proceedings of First International Workshop on Sensing and Big Data Mining*, pages 1–6. ACM, 2013.
[43] A. Nappa, R. Johnson, L. Bilge, J. Caballero, and T. Dumitras. The attack of the clones: A study of the impact of shared code on vulnerability patching. In *IEEE Symposium on Security and Privacy*, 2015.
[44] A. Nuno and F. A. S. John. How to ask sensitive questions in conservation: A review of specialized questioning techniques. *Biological Conservation*, 189:5–15, 2015.
[45] A. A. O'Connell. *Logistic regression models for ordinal response variables*. Number 146. Sage, 2006.
[46] K. Olmstead and A. Smith. Americans and cybersecurity. 2017.
[47] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on amazon mechanical turk. 2010.
[48] E. Peer, J. Vosgerau, and A. Acquisti. Reputation as a sufficient condition for data quality on amazon mechanical turk. *Behavior research methods*, 46(4):1023–1031, 2014.
[49] S. Presser, M. P. Couper, J. T. Lessler, E. Martin, J. Martin, J. M. Rothgeb, and E. Singer. Methods for testing and evaluating survey questions. *Public Opinion Quarterly*, 2004.
[50] E. Redmiles, M. Mazurek, and J. Dickerson. Dancing pigs or externalities? measuring the rationality of security decisions. In *Economics and Computation (EC)*, 2018.
[51] E. M. Redmiles, Y. Acar, S. Fahl, and M. L. Mazurek. A summary of survey methodology best practices for security and privacy researchers. Technical report, 2017.
[52] E. M. Redmiles, S. Kross, and M. L. Mazurek. How i learned to be secure: a census-representative survey of security advice sources and behavior. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 666–677. ACM, 2016.
[53] E. M. Redmiles, S. Kross, A. Pradhan, and M. L. Mazurek. How well do my results generalize? comparing security and privacy survey results from mturk and web panels to the us. Technical report, 2017.
[54] E. Rescorla. Security holes... who cares. In *USENIX Security Symposium*, 2003.
[55] J. Ross, L. Irani, M. Silberman, A. Zaldivar, and B. Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI'10 extended abstracts on Human factors in computing systems*, pages 2863–2872. ACM, 2010.
[56] A. Sarabi, Z. Zhu, C. Xiao, M. Liu, and T. Dumitraş. Patch me if you can: A study on the effects of individual user behavior on the end-host vulnerability state. In *International Conference on Passive and Active Network Measurement*, pages 113–125. Springer, 2017.
[57] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The Emperor's New Security Indicators. *IEEE Symposium on Security and Privacy*, pages 51–65, 2007.
[58] S. Schnorf, A. Sedley, M. Ortlieb, and A. Woodruff. A comparison of six sample providers regarding online privacy benchmarks. In *SOUPS Workshop on Privacy Personas and Segmentation*, 2014.
[59] M. Shahzad, M. Z. Shafiq, and A. X. Liu. A large scale exploratory analysis of software vulnerability life cycles. In *International Conference on Software Engineering*, 2012.
[60] A. Sotirakopoulos, K. Hawkey, and K. Beznosov. On the challenges in usable security lab studies: lessons learned from replicating a study on ssl warnings. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, page 3. ACM, 2011.
[61] F. Strack and T. Mussweiler. Explaining the enigmatic anchoring effect: Mechanisms of selective accessibility. *Journal of personality and social psychology*, 73(3):437, 1997.
[62] Y. Tian, B. Liu, W. Dai, B. Ur, P. Tague, and L. F. Cranor. Supporting privacy-conscious app update decisions with user reviews. In *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 51–61. ACM, 2015.
[63] R. Tourangeau and T. W. Smith. Asking sensitive questions: The impact of data collection mode, question format, and question context. *Public opinion quarterly*, 60(2):275–304, 1996.
[64] R. Tourangeau and T. Yan. Sensitive questions in surveys. *Psychological bulletin*, 133(5):859, 2007.
[65] K. Vaniea, E. J. Rader, and R. Wash. Betrayed by updates: How negative experiences affect future security. In *ACM Conference on Human Factors in Computing*, 2014.
[66] K. Vaniea and Y. Rashidi. Tales of software updates: The process of updating software. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3215–3226. ACM, 2016.
[67] K. E. Vaniea, E. Rader, and R. Wash. Betrayed by updates: how negative experiences affect future security. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 2671–2674. ACM, 2014.
[68] R. Wash and E. J. Rader. Too much knowledge? security beliefs and protective behaviors among united states internet users. In *SOUPS*, pages 309–325, 2015.
[69] G. B. Willis. *Cognitive Interviewing: A Tool for Improving Questionnaire Design*. 2005.
[70] G. Y. Wong and W. M. Mason. The hierarchical logistic regression model for multilevel analysis. *Journal of the American Statistical Association*, 80(391):513–524, 1985.
[71] S. Yilek, E. Rescorla, H. Shacham, B. Enright, and S. Savage. When private keys are public: Results from the 2008 Debian OpenSSL vulnerability. In *Internet Measurement Conference*, 2009.

A SURVEY QUESTIONNAIRE

- Condition: Self
  - Q1S: Imagine that you see the message below appear on your computer. [image of update message] Would you install the update?
    - Yes, the first time I saw this message.
    - Yes, within a week of seeing this message.
    - Yes, within a few weeks of seeing this message.
    - Yes, within a few months of seeing this message.
    - No.
    - I don't know.
  - Q2S: What would make you want to install this update? [multiple selection, optional]
    - I always install updates *(Mapping: General Tendency)*
    - I trust this software company *(Mapping: Application)*
    - The features seem like something I would want *(Mapping: Features / Security-Only)*
    - I wasn't satisfied with the current version
    - The current version was broken
    - It was a security related update
    - I use this software frequently, so keeping it updated is important *(Mapping: Application)*
    - Previous updates that I have installed for this software made the software or my computer crash less *(Mapping: Risk)*
    - I don't have to restart to install this update *(Mapping: Cost)*
    - It seemed like it wouldn't take very long to complete this update *(Mapping: Cost)*
    - Other: [text entry]
  - Q3S: What would make you not want to install this update? [multiple selection, optional]
    - I rarely install updates *(Mapping: General Tendency)*
    - I wouldn't have time *(Mapping: Cost)*
    - I wouldn't want to restart *(Mapping: Cost)*
    - I wouldn't want to lose stuff while updating *(Mapping: Risk)*
    - It looked like it would be disruptive
    - This update didn't seem important
    - The update was not related to security
    - I do not use this software frequently, so keeping it updated is not important *(Mapping: Application)*
    - I wouldn't want the features it would add *(Mapping: Features / Security-Only)*

    - I'm satisfied with the current version
    - The update might make the application harder to use *(Mapping: Risk)*
    - I don't trust this software company *(Mapping: Application)*
    - Too many updates for this software
    - The software or my computer crashed more after I have updated in the past *(Mapping: Risk)*
    - I have had trouble updating this application in the past *(Mapping: Risk)*
    - I would worry about compatibility issues *(Mapping: Risk)*
    - I wouldn't want to lose stuff while updating *(Mapping: Risk)*
    - Other: [text entry]
- Condition: Friend
  - Q1F: Imagine that a friend or relative sees the message below on their computer and calls you for advice. What would you tell them?
    - Install the update immediately.
    - Install the update sometime this week.
    - Install the update within a few weeks.
    - Install the update within a few months.
    - Don't install the update
    - I don't know
  - Q2F: What would make you tell your friend to install this update? [multiple selection, optional]
    - I always install updates
    - I trust this software company
    - The features seem like something they would want
    - They weren't satisfied with the current version
    - The current version was broken
    - It was a security related update
    - They use this software frequently, so keeping it updated is important
    - Previous updates that they have installed for this software made the software or their computer crash less
    - They don't have to restart to install this update
    - It seemed like it wouldn't take very long to complete this update
    - Other: [text entry]
  - Q3F: What would make you not recommend that your friend install this update? [multiple selection, optional]
    - I don't install updates
    - They wouldn't have time
    - They wouldn't want to restart
    - They wouldn't want to lose stuff while updating
    - It looked like it would be disruptive
    - This update didn't seem important
    - The update was not related to security
    - They do not use this software frequently, so keeping it updated is not important
    - They wouldn't want the features it would add
    - They are satisfied with the current version
    - The update might make the application harder to use
    - I don't trust this software company
    - Too many updates for this software
    - The software or their computer crashed more after they have updated in the past
    - They have had trouble updating this application in the past
    - I would worry about compatibility issues
    - They wouldn't want to lose stuff while updating
    - Other: [text entry]

  The order of [Q4-7], Q8, and Q9 was randomized.
- Q4: Over the past year, how frequently do you feel like [application] has frozen (e.g., hung) or crashed?
  - Less than once a week
  - At least once a week but not more than three times a week
  - At least three times a week but not more than five times a week
  - Five times a week or more
- Q5: Over the past year, have you noticed that updating [application] changes how frequently it freezes (e.g., hangs) or crashes?
  - Yes, it crashes more after I update.
  - Yes, it crashes less after I update.
  - No, updating [application] has no impact on how frequently it crashes.
- Q6: Over the past year, how frequently do you feel like any application on your computer or your computer itself crashed?
  - Less than once a week
  - At least once a week but not more than three times a week
  - At least three times a week but not more than five times a week
  - Five times a week or more
- Q7: Over the past year, have you noticed that updating [application] changes how frequently any application on your computer or your computer itself crashes?
  - Yes, my computer crashes more after I update.
  - Yes, my computer crashes less after I update.
  - No, updating [application] has no impact on how frequently my computer crashes.
- Q8: In general, how quickly do you install updates for applications on your computer or for your computer itself (e.g., the computer operating system)?
  - As soon as I see the update prompt.
  - Within a week of seeing the prompt.
  - Within a few weeks of seeing the prompt.
  - Within a few months of seeing the prompt.
  - I don't install updates that appear on my computer.
  - I don't know.
- Q9: Do you use any of the following software on your home or work computer? [Multiple answer]
  - A Norton software product (for example, Norton AntiVirus, Norton Family Premier, Norton Mobile Security, Norton Small Business)
  - A Symantec software product (for example, Symantec AntiVirus, Symantec Endpoint Protection)
  - Another anti-virus software (for example, McAfee Antivirus Plus, Kaspersky AntiVirus, Bitdefender Antivirus Plus)
  - None of the above

– I Don't Know

B UPDATE MESSAGES

Figure 6 shows all 11 update messages. Crash data is available for 5 versions: Adobe Reader 9.5.1.283 (Figure 6b), Flash Player 10.3.181.14 (Figure 6g), Flash Player 11.0.1.152 (Figure 6h), Firefox 8.0.1.4341 (Figure 6i) and Opera 11.64.1403.0 (Figure 6k).

C SURVEY DEMOGRAPHICS

We have demographics only for the SSI participants as our surveys were conducted in a privacy preserving manner and participant demographics were not collected in the survey directly. SSI produces aggregated reports on sample demographics; AMT does not. Table 4 presents a comparison of the SSI sample demographics with the U.S. Census [1].

Metric	SSI	Census	Metric	SSI	Census
Male	49.7%	48.2%	H.S. or below	40.7%	41.3%
Female	50.3%	51.8%	Some college	22.2%	31.0%
			B.S. or above	37.1%	27.7%
Caucasian	67.5%	65.8%	18-29 years	29.6%	20.9%
Hispanic	9.1%	15%	30-49 years	39%	34.7%
African American	12.2%	11.5%	50-64 years	27.8%	26.0%
Other	11.2%	7.6%	65+ years	3.1%	18.4%
<$20k	19.6%	32%			
$20k-$40k	23.6%	19%			
$40k-$75k	28.6%	18%			
$75k-$100k	11.8%	11%			
$100k-$150k	11.8%	12%			
$150k+	4.5%	8%			

Table 4: Demographics of the 455 respondents in the SSI sample compared to U.S. Census demographics [1].

D ADDITIONAL ANALYSIS

D.1 RQ4: Regression Models

Table 5 presents the results for the hierarchical regression modeling conducted using a dataset containing observations or responses to all eleven messages; these models do not include any risk metrics, as this risk data was only available for the five update messages released in 2011 or 2012. Table 7 presents the results for the risk-related modeling, conducted using only data pertaining to the five update messages with risk metrics available.

D.2 Survey Filtering

We mapped the answer choices for the second and third survey questions, which queried why respondents would and would not want to update in response to the given message. This mapping is indicated in Appendix A above. In line with the approach of Fahl et al., who filtered out survey respondents who self-reported not answering their survey honestly, we filter out respondent's who's answers to Q2 and Q3 are clearly illogical: that is we remove (1) any respondents who noted that they would not install the update shown because it required a restart, but the update message they saw explicitly stated that it did not require a restart, (2) any respondents who indicated that they would install the update shown because it did not require a restart, but in fact saw an update message that stated that it did require a restart, (3) any respondent who noted that they would not install the update because it contained features they did not want, but who saw an update message that mentioned only security and no other enhancements, and (4) any respondent who noted that they would install an update because it contained features they would want, but who saw a message that mentioned security and no other enhancements. This filtering results in a dataset consisting of 981 respondents (44% of the original 2,092): 749 (43% of the original) from MTurk and 232 (51% of the original) from SSI.

D.3 Survey Internal Consistency

In addition to filtering the dataset, we also checked for internal consistency more broadly by testing for independence (X^2, corrected with Holm-Bonferonni procedure) between responses to Q2 and Q3 and: the actual message features (for security-only, cost, and application) or later responses to Q4-8 (risk and general tendency). Internally consistent responses should not be independent (i.e., should produce a significant X^2 independence test result). Table 6 shows our results.

		Comparison	X^2	p-value
Detailed Constructs	Cost	Why: Low cost | Message: Restart	6.39	0.024*
		Why Not: High cost | Message: Restart	0.917	0.384
	Security / Features	Why: Features | Security-Only: Restart	4.72	0.067
		Why Not: Features | Security-Only	0.050	0.823
	Application	Why: Application | Application	38.2	<0.001*
		Why Not: Application | Application	11.8	0.019*
General Constructs	General Tendency	Why: Always Update | General Tendency	141.2	<0.001*
		Why Not: Rarely Update | General Tendency	7.77	0.042*
	Risk	Why: Risk | Sys. Crash. Freq.	15.6	0.005*
		Why: Risk | Sys. Crash. More	4.95	0.040*
		Why: Risk | App. Crash. Freq.	15.5	0.005*
		Why: Risk | App. Crash. More	5.56	0.401
		Why Not: Risk | Sys. Crash. Freq.	3.09	0.031*
		Why Not: Risk | Sys. Crash. More	17.3	<0.001*
		Why Not: Risk | App. Crash. Freq.	10.5	0.028*
		Why Not: Risk | App. Crash. More	7.59	0.0166*

Table 6: X^2 tests comparing respondent's reported reasons for updating with the true message features or their later survey responses.

Factor	Full					
	Baseline			Risk		
	Measurement	Survey	Survey: F	Measurement	Survey	Survey: F
Gen. Tendency	1.56*	4.69*	4.36*	1.54 *	4.75*	4.82*
Risk: Sys. Crash Freq.				1.03*	0.82	1.76*
Risk: Sys. Crash More				1.00	0.87	1.09
Risk: App. Crash Freq.				1.00	1.14	1.37
Risk: App. Crash More				0.89*	1.06	0.53*
Application: Firefox				0.66*	0.74*	0.72
Application: Opera				0.29*	1.02	1.22
Application: Reader				0.63*	0.72*	0.64*
Sample: MTurk	–	1.37	1.17	–	1.06	1.10
n	41,551	749	480	41,551	749	480

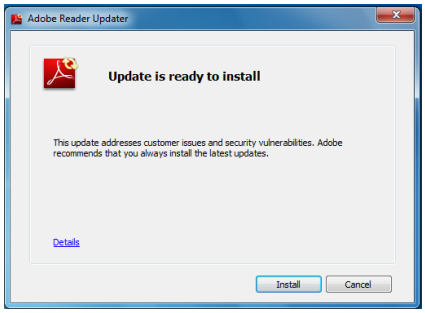
Table 7: Table of hierarchical regression models for risk factors in the dataset containing the five messages for which these features are available; p-values significant at *alpha* = 0.05 are marked with *. Survey: F is the filtered survey data.

Factor	Full						Reader						Flash					
	Baseline			Application			Baseline			Cost (Restart)			Length (words)			Security, Length		
	S	SF	WINE	S	SF	WINE	S	SF	WINE	S	SF	WINE	S	SF	WINE	S	SF	WINE
Gen. Tendency	6.09*	4.36*	1.60*	4.39*	4.39*	1.58*	4.82*	4.19*	1.84*	5.61*	4.10*	1.89*	6.47*	4.10*	1.45*	6.48*	4.10*	1.44*
Security-Only																6.48*	4.11*	1.35*
Length											1.02	1.03	0.98*			1.05	0.84	3.33*
Cost							0.99	1.47	0.86*							1.02	1.06	0.93*
App.: Firefox				0.70*	0.69*	0.66*												
App.: Opera				0.98	0.29*	1.17												
App.: Reader				0.71*	0.68*	0.63*												
Sample: MTurk	1.29	1.17	–	1.30*	1.18	–	0.90	1.39	–	1.23	1.41	–	1.40*	1.16	–	1.41*	1.19	–
n	2,092	981	517,932	2,092	981	517,932	386	219	306,654	386	219	306,654	1,149	301	343,697	1,149	301	343,697

Factor	Flash					
	Security-Only			Security, Length		
	S	SF	WINE	S	SF	WINE
Gen. Tendency	6.47*	4.12*	1.28*			
Security-Only	0.94	0.65	4.54*			
Length						
Cost						
App.: Firefox						
App.: Opera						
App.: Reader						
Sample: MTurk	1.41*	1.17	–			
n	1,149	301	343,697	1,149	301	343,697

Table 5: Table of hierarchical regression models on the full datasets, the Reader messages, and the Flash messages. S indicates for models built on the survey data, SF indicates for models built on the filtered survey data, and WINE indicates for models built on the measurement data. p-values significant at *alpha* = 0.05 are marked with *.

(a) Adobe Reader 9.3.2.163

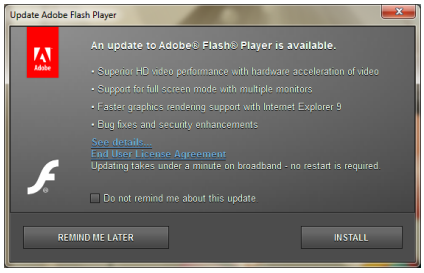


(b) Adobe Reader 9.5.1.283 (crash data available)

(c) Flash Player 10.0.22.87

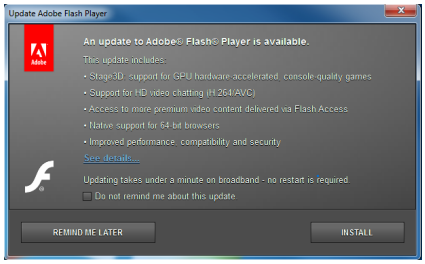
(d) Flash Player 10.0.45.2

(e) Flash Player 10.1.53.64

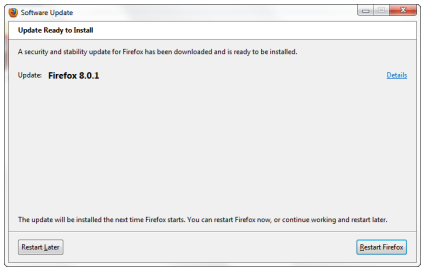


(f) Flash Player 10.2.152.26

(g) Flash Player 10.3.181.14 (crash data available)



(h) Flash Player 11.0.1.152 (crash data available)



(i) Firefox 8.0.1.4341 (crash data available)

(j) Opera 10.61.2484.0

(k) Opera 11.64.1403.0 (crash data available)

Figure 6: Update Messages