# Mental models of software updates

Kami Vaniea, Emilee Rader, and Rick Wash

Michigan State University

## Abstract

One of the largest preventable sources of computer compromise is old software that has not been updated with the latest security-related updates. Security updates correct known vulnerabilities in software and protect the computer from future attacks. However, users do not always choose to install updates; instead, they avoid or delay installation, placing their computers and data at an increased risk of of harm. Our research explores different mental models users have regarding software updates and connects those models to the past update behavior of participants. We used a multi-method approach to collect interview, survey, and computer log data from 37 Windows 7 users. We analysed the qualitative data to understand how people conceptualized updates, and made decisions regarding them. We observed a disconnect between participants' understanding of update behavior and reality. Issues such as locus of control, and the unknown consequences of updating software negatively impacted update behavior.

Mental models of software updates

## Introduction

Software updates are one of the most effective mechanisms users have at their disposal to protect their computers. Software is rarely released problem-free; most companies release a number of *software updates* to fix bugs in the software and add in new features. *Security* updates are particularly important because they are one of the primary mechanisms for protecting home computers from malicious software that leveraged known vulnerabilities. The majority of computer compromises result from vulnerabilities for which a security update is available but has not yet been installed (Microsoft, 2012; Symantec, 2013). Timely installation of security updates can protect users from the most common attacks (Symantec, 2013).

Many software companies have worked to find ways to improve end-user compliance and increase the number of fully updated systems. For example, each successive version of Microsoft Windows has had additional features to automate the installation of software updates with less human involvement (Gkantsidis, Karagiannis, & Vojnovic, 2006). Current software updates (and Microsoft Windows Updates in particular) have largely removed the need for human decisions.

However, not all security technologies can completely remove the human from the decision-making process (Adams & Sasse, 1999). Cranor (2008) assembled a useful framework for reasoning about when it is advisable to keep a 'human in the loop'. This framework is relevant to software updates because updates cannot be installed completely without user intervention for three reasons: 1) occasionally, an update will introduce a new bug into the system, and users will want to postpone installing that update, 2) an update may introduce or remove features which impact user activities causing users to want to avoid installing the update, and 3) many updates require rebooting the computer to install, which is highly disruptive of user activities. Therefore, users need to be kept informed and given options during the update process. Software update systems have tried to accomodate users by finding an appropriate balance between forcing users to install updates to improve security, and giving them appropriate choices.

We conducted a multi-method user study to better understand how people make decisions about software updates. With each participant, we conducted semi-structured interviews to understand how the participant viewed software updates, had him or her take a survey to provide more structured opinions, and collected log data about update installation from his or her computer. We found that over half of our participants were not aware of what their computers' Windows update settings were or when the updates were being installed. Participants also discussed avoiding software updates to avoid user interface changes, because they did not understand what the software was, and because the software appeared to be currently functional. For the majority of users, their software updated in a way contrary to their intentions. However, many of these computers were also more secure than the user intended. This means that improving usability of software updates might not lead to improved security, which has interesting implications for the design of software update systems.

## Integrating Humans Into Security

Security failures are often seen as a human problem rather than a technological one. For example, West (2008) wrote, "The most elegant and intuitively designed interface does not improve security if users ignore warnings, choose poor settings, or unintentionally subvert corporate policies."

In the workplace, computer and information security is the joint responsibility of end users and system administrators, but end users are often seen as "inherently insecure" (Adams & Sasse, 1999; Kaemer & Carayon, 2007). With the rise of discretionary computer usage and employees bringing their devices to work, end users bear more of the responsibility for the security of their many devices in and out of the workplace. Such users are their own system administrators, whether they know it or not, and how to best support them is the subject of much research.

Users are perceived as the weak link for several reasons:

- The expectations placed on end users with respect to managing the security of their computers are unrealistic; users cannot be expected to think like system administrators (Besnard & Arief, 2004)
- Security only becomes apparent to end users when something has already gone wrong (Zurko, 2005)
- Security is not users' first priority, and given a choice, they will choose the insecure option if it gets them closer to their goals (Edwards, Poole, & Stoll, 2007)
- When users make mistakes, it makes the job of system administrators that much harder (Edwards et al., 2007)

System designers frequently attempt to either nudge (Thaler & Sunstein, 2008) or force users into making secure decisions. The designer might try to make security the user's top priority by creating mechanisms that prevent them from completing any action until the security aspects have been taken care of. The system might make the security-related actions so easy and unobtrusive that they can do whatever is necessary as part of their normal workflow or primary task (path of least resistance). Or, it might remove all ability to act from the user by completely automating the security aspects of the system, so users cannot make the wrong choice (Yee, 2002).

However, human capabilities are frequently necessary for the task at hand (von Ahn, Blum, Hopper, & Langford, 2003). A "default" level of security is not appropriate for all users in all situations (Furnell, 2005) and automatic security cannot be used when configuration decisions must be made, or when automation is too "restrictive, inconvenient, expensive, or slow." Cranor (2008) advocates that system designers should design for both automation and user responsibility for security by identifying which security aspects of the of the system cannot be automated and are likely to fail due to user intervention and better support users in those circumstances.

Software designers need to be aware that there is a tradeoff between visibility and intrusiveness. In circumstances when the user must remain "in the loop", communication between the system and the user is crucial, and it is the role of the software designer responsible for making sure the software is secure to figure out where this communication must take place (Cranor, 2008). Relegating security to “Advanced” tabs and burying it in menus is one way to (intentionally or unintentionally) ensure that the user retains the defaults (Furnell, 2005).

How that communication might best be accomplished is the subject of much usable security research. One of the core values of usability is “walk up and use” interfaces that do not require special learning or expertise; however, this approach may result in prioritizing the usability aspects of the system over the security aspects, because security may be more complicated than a “walk up and use” interface can communicate (Kainda, Fléchais, & Roscoe, 2010). Recommendations to improve the usability of the communication between the system and the user are often assumed to also improve security, because users will be more involved, but this is not always the case.

To further complicate matters, end users often delegate the responsibility for the security of their systems, to technology, other people, organizations, or institutions (Dourish, Grinter, Delgado De La Flor, & Joseph, 2004). Delegating responsibility to technology—to the system itself—is like ‘set it and forget it’ security: do it once, and never have think about it again. Once this has taken place, security becomes invisible, and is not often revisited. This means that the system keeps going with the past settings indefinitely. Policies like this are too rigid, because an invisible policy can’t adapt to users’ changing needs and circumstances (Edwards et al., 2007).

Software updates are a particularly interesting case for studying how to include humans in security systems. From a security perspective, quickly installing security updates is the correct behavior, and can often be safely initiated without user intervention. However, many updates require that the computer reboot to complete installation, necessitating human involvement, and making the automated update process visible to users who may not understand why it is necessary (Vaniea, Rader, & Wash, 2014).

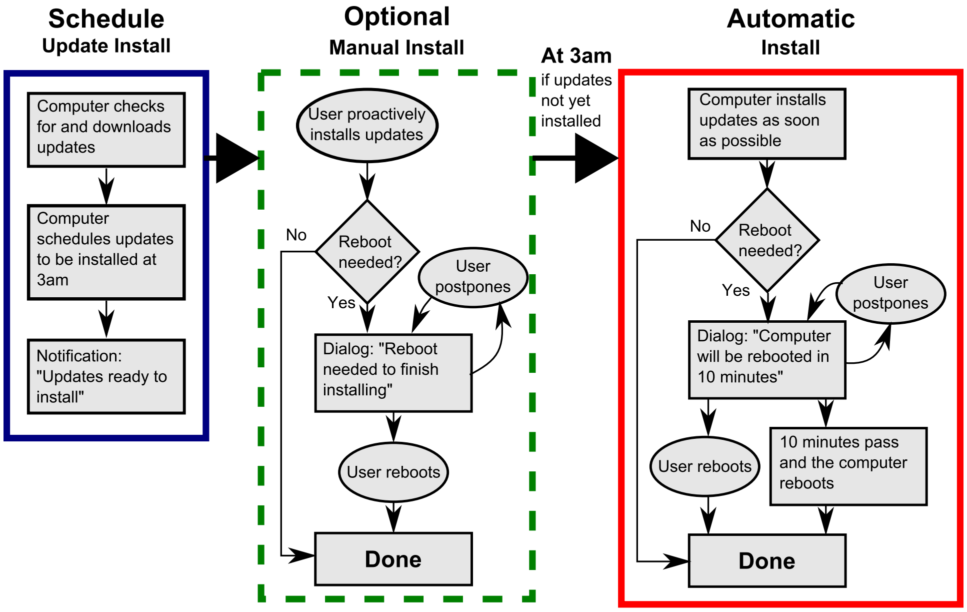
Schedule
Update Install
Computer checks for and downloads updates
Computer schedules updates to be installed at 3am
Notification: "Updates ready to install"
Optional
Manual Install
User proactively installs updates
Reboot needed?
No
Yes
User postpones
Dialog: "Reboot needed to finish installing"
User reboots
Done
At 3am
if updates not yet installed
Automatic
Install
Computer installs updates as soon as possible
Reboot needed?
No
Yes
User postpones
Dialog: "Computer will be rebooted in 10 minutes"
User reboots
10 minutes pass and the computer reboots
Done

*Figure 1*. The Windows Update process. Ovals represent user actions, diamonds decisions, and rectangles computer behavior. This diagram was created based on prior update work by Gkantsidis et al., and experimentation using a Virtual Machine with Microsoft Windows 7 Service Pack 1 installed.

## Software Updates Improve Security

Updating software is an important part of keeping a computer secure, and keeping all software up-to-date will protect a user against the most common security exploits. Symantec (2013) has data showing that the majority of computers are compromised using vulnerabilities where an update is available, but has not yet been applied. The majority of web exploits use the top twenty vulnerabilities, all of which have available updates (Symantec, 2013). Likewise, Microsoft observes that all of the vulnerabilities exploited by the most popular exploit kit have available updates (Microsoft, 2012).

It is important to update software as soon as possible after a security update is released. Updates correcting security vulnerabilities are released an average of 1.2 months after an exploit for the vulnerability seen in the wild (Marconato, Nicomette, & Kaaniche, 2012). However, exploits released before a vulnerability becomes public knowledge (zero-day vulnerabilities) are used to attack a relatively small number of computer systems. Once a zero-day vulnerability becomes public knowledge the number of exploits using it increases 183–85,000 times and the number of attacks increases 2–100,000 times (Bilge & Dumitras, 2012). Likely for this reason, 60% of Microsoft's vulnerabilities are made public knowledge the same day as the update correcting the vulnerability is released (Marconato et al., 2012), enabling users to protect themselves before exploits become readily available. For these and other security reasons, the faster the user updates their system the less likely they will be vulnerable to new attacks.

There has been limited investigation into what motivates users to update or not update software on their computer. LaRose et al. surveyed undergraduate students about their online safety behaviors and beliefs. They found that people who feel like online safety is their personal responsibility are more likely to want to perform safe online behaviors (LaRose, N. J. Rifon, & Enbody, 2008; LaRose, N. Rifon, Liu, & Lee, 2005). They also found that coping efficacy beliefs were correlated with intention to perform software updates (LaRose, N. Rifon, et al., 2005). These studies are based on self-report data, and are unable to examine whether subjects actually undertake their stated behaviors.

## Windows Update

Windows Update is a software update service provided for free by Microsoft and the primary mechanium used to install Windows Updates. It began as a website that Windows 95 users had to visit to find out whether operating system updates were available. A new "Critical Update Installation Tool", introduced with Windows 98, included automatic checking for updates, and it also notified users about critical updates which they had to then manually retrieve and install. In 2000, Windows ME shipped with "Automatic Updates", a tool that could automatically download and optionally install software updates. Automatic installation of updates became the default with Windows XP SP2. Starting with Windows Vista updates categorized as "important," and also "recommended" updates (Wikipedia, n.d.) were automatically installed by default.

The result of this evolution, the Windows Update software used in Windows 7, demonstrates the compromise Microsoft software designers made between automating the update process for the safety of users and giving users responsibility for their computer use. As shown in Figure 1, by default each update in Windows Update goes through three stages: an install scheduling, a time for manual install, and an automatic installation.

**Stage 1**: (left blue box) The computer automatically checks for updates, downloads them, schedules them to be installed at 3am the next morning, and then notifies the user that updates are available to be installed. The notification appears temporarily in the bottom right of the screen, and a gold shield is added to the "Shut down" button on the start menu.

**Stage 2**: (green middle box) The computer waits silently for the user to manually initiate the install process. This gives the user the opportunity to take responsibility for their updates. Users may manually install updates by opening the Windows Update program and selecting "Install updates." If a reboot is needed, the user is notified by a dialog with a postpone option. However, the dialog only reminds the user, it does not compel a reboot.

**Stage 3**: (red right box) The computer starts installing updates automatically at 3am or the first time the computer is turned on after 3am. If any update requires a reboot the computer presents the user with a dialog warning that the reboot will happen in 10 minutes. The dialog countdown timer has options to "Reboot now" or "Postpone"; the user cannot escape the countdown completely. If the user does nothing, the computer will immediately reboot. However, if the user chooses to intervene during the 10 minute interval, they can "Restart now" which causes an immediate reboot of the system, or "Postpone" for an additional 10 minutes, 1 hour, or 4 hours. This stage automates security decisions, removing the human from the loop.

The design of Windows Update is a compromise between fully automating updates and giving users full responsibility for updates, and it has been successful at increasing security. After the release of Windows XP SP2, Gkantsidis et al. (2006) observed that only 5% of SP1 users had fully updated computers, but 90% of SP2 users had fully updated computers. They also observed that 80% of SP2 users downloaded the latest update within two days of release. In 2011, 66% of Windows users (all versions) were completely up-to-date, and 84% had at least one of the three most recent updates (Microsoft, 2012).

## Methods

Software updates are an instance where security system designers have mostly, but not completely, removed humans from security decision-making. To better understand user decision-making about software updates, we undertook a multi-method study that included semi-structured interviews, an online survey, and log-data analysis. This allowed us to measure both users' beliefs and impressions about what their computers were doing, and what their computers were actually doing.

### Participants and Protocol

We sent an email through the University Registrar to a random sample of 1000 graduate students, excluding Math and Engineering students, asking for volunteers to participate in the study. Ninety-five people took a screening survey to ensure that they were Windows 7 users (so we could collect log data) and did not have any formal training in computer management, IT, or system administration. Thirty-seven people who were eligible came to our lab to participate, and brought their laptop running Windows 7 with them. Three of these participants were Mac users running Windows in a virtual machine. Participants ranged in age from 21 to 57 with an average age of 31; Seventeen were male, and twenty were female. These demographics approximately match those of the larger graduate student population.

After informed consent, the study consisted of three parts: a brief survey, Windows log data collection, and a semi-structured interview. While one member of the research team administered the survey and interview, another member used a custom Powershell script to collect setting and log data on the participant's laptop. Participants were given the option of observing the data collection. Participants also received $25 as a thank you for helping us. This study was approved by our university's IRB.

## Three Types of Data

We collected three different datasets from each participant: a set of survey responses, log data from their Windows 7 laptop, and a transcribed, semi-structured interview. We began by analysing each type of data separately. Then, using a pseudonym assigned to each participant, we re-combined the three data sources to compare participant responses and behavior across data sources. This analysis structure ensured that we accurately understood the meaning of each separate type of data before comparing attitude, recall, and behavior across data sources.

**Semi-Structured Interviews.** System designers have made most software updates highly automated and relatively invisible to end users. Users don't spend much time thinking about software updates. This poses a challenge for conducting interviews: how can we get participants to talk about past experiences and reveal how they think about updates? And how can we avoid having participants think about updates too deeply during the interview – and change their opinions, which would lead to invalid data?

After a series of pilot tests, we decided to use three interviewing techniques: free-listing, hypothetical scenarios, and recollection of specific past instances.

We began by asking participants to complete a *free-listing activity* (Brewer, 2002): write down as many examples as came to mind for the prompt, "things that can happen if the software on your computer is too old or out of date". We then read each example and asked the participant to discuss his or her response further. Free-listing allows us to explore the semantic domain of updates; that is, it helps the participant to think through and explain the range of activities and concerns that are relevant to a discussion of software updates. The use of a non-specific prompt, reading items back to the participant, and using the items as semantic cues to discuss past instances help participants to fully explore the topic of software updates (Brewer, 2002).

Next, we presented participants with a series of five *hypothetical scenarios* paired with probing questions; we wanted the participant to do most of the talking so that we could uncover their attitudes, beliefs, and mental models about updates. The scenarios involved being prompted to restart an internet browser mid-task, seeing that a large number of urgent Windows updates were available, reading a news article about a virus, a software program that costs money to update, and a slow computer with lots of warnings. In each scenario the participant was asked what they would do, and if they had ever encountered a similar situation. A "browser" was mentioned in the first scenario and Windows 7 was explicitly mentioned in the second and fifth scenarios. All other software mentioned in this work was brought up by participants. Hypothetical scenarios are effective methods of learning how participants conceptualize their decisions relate to software updates (Wash, 2010). By asking to recall specific instances, participants provide more details and are better able to recall information that influenced their decision-making at the time. Recalling specific instances provides data that is more likely to represent broad decision-making patterns than asking participants to describe general patterns of past behavior (Onwuegbuzie & Leech, 2007).

**Analysis**: After transcribing and anonymizing the interviews, we performed a bottom-up, inductive coding. We started with an initial list of themes identified by the research team, and expanded the codes as we read through transcripts. During this period, members of the team met frequently to discuss and revise the codes. Themes identified include "negative update experiences", "attitudes toward delaying updates", and "why updates are important."

As we created each code, we examined other transcripts to check for representativeness and identify which traits were common across participants. We also explicitly looked for negative cases: cases that share most of the pattern but are explicitly missing one or two key pieces.

When coding was complete, we summarized the data into a matrix that displayed themes by participant (Miles, Huberman, & Saldaña, 2013). This matrix allowed us to understand each individual's perspective on updates by reading down the column that summarizes their responses.

We then compared the summary data matrix to original interviews to verify the correctness of each summary, check for the meaning of outliers, verify surprises, specifically look for evidence for negative cases, and try to prevent researcher confirmation bias in our data. (Onwuegbuzie & Leech, 2007). This process provides confidence that our summaries are valid representations of participant views as expressed in the interviews.

**Survey.** We used an in-lab computer survey to ask structured, closed-ended questions. A survey allowed us to ensure that all participants were asked the same set of factual and opinion-based questions in a consistent, comparable manner. In addition to background information such as participant demographics, computer skills, and installed software, we also asked participants for their current understanding of the state of software updates on their computer. This includes whether automatic updates were enabled and whether updates were usually installed manually or automatically. Questions were written following the guidance of Dillman, Smyth, and Christian (2009) and were pre-tested to ensure participants understood the questions.

**Analysis**: We generated descriptive statistics for each participant, as well as extracting the specific questions about the user's knowledge of current state of the automatic updates setting, their belief about whether updates are installed manually or automatically, and their belief about the timing of install.

**Windows Logs.** The Windows operating system, along with many Windows services, records information about system events in log files which contain detailed records of system and user behavior. Our Powershell script collected the current Windows Update settings, which allowed us to determine whether updates were turned off, set to notify the user before download, or set to install automatically without user intervention (default behavior). The script did not collect any personally identifiable information.

We also collected a list of installed updates from the Windows Update API, and a copy of all Windows Update log files which provided detailed event information from the last several months of use. This allowed us to calculate the time between when an update had been downloaded and when it was installed, which is important because this is the part of the update process that the user has the most control over—i.e., when the update is installed and when the computer reboots to finish installing an update (if necessary). One limitation of this method is that the detailed logs represented between 1 and 17 months (average of 6) of usage data depending on how often the participant had been using the machine.

**Analysis:** We first looked at each update separately. We limited our log analysis to updates which were associated with a Microsoft Knowledge Base (KB) number, which allowed us to link update events across log files. We marked the update as proactively installed by the user if it was installed before 3am[1] the morning following the update's download. We marked it as automatically installed by Windows Update if it was installed after 3am. Then we aggregated all updates for a user: did the user always install proactively, usually ($> 50\%$) install proactively, usually automatic install, or always automatic install?

To determine if the user was updating individual programs on their computer we looked at three programs frequently mentioned by users in interviews: iTunes, AdobeReader, and Java. For each of these we compared the user's currently installed version to the most recent version available at the time of the interview. In the case of AdobeReader versions 9.X, 10.X, and 11.x are supported, so we marked which version was installed, and if it was updated.

## Combining Data for Analysis

In order to compare user attitudes, user beliefs, and user behavior, we constructed a data matrix that combined data from all three sources of information (Miles et al., 2013). For each participant, we created entries on three topics: general updates, the automatic updates setting, and the timing of update installs. For each of these topics, we included a row of data from each of the three data sources: the participant's attitude and understanding of the topic summarized from the

[1] One user had a scheduled install time setting of 4am, all other users had the default of 3am, for simplicity we always refer to this time using the default of 3am or "overnight".

interviews, the participants current beliefs from the survey, and the participant's past behavior summarized from the log data.

After creating the combined data matrix, we again examined our data to ensure validity (Onwuegbuzie & Leech, 2007). All members of the research team participated in looking for patterns across participants, checking for negative cases, verifying summaries with original source data, and including footnotes and caveats for our summaries. in two, and verified each piece with the source data.

## Windows Updates

We used our interview data and our survey data to characterize two things: what the user thought the computer was doing, and what the user wanted the computer to do. We then compared these two perceptions with the log data from that user's computer to determine if they matched. That is, we compared user's stated *understanding* of what their computer was doing with log data and settings that indicated what the computer actually did, to see whether users understood what was happening on their computer. Then we compared each user's stated *intentions* — what they wanted their computer to be doing — to the log data and settings to determine whether they were actually able to make the computer do what they wanted.

### Understanding Windows Updates

Many of our participants misunderstood what their computers were doing regarding Windows updates. Twenty-eight of the 37 participants (78%) had at least one inconsistency between what the subject thought their computer was doing and what the log data indicated it was doing. There are two topics that participants had misunderstandings about: the Windows Update setting about whether it installed updates automatically, and how quickly updates were installed.

***Automatic Updates Setting.*** Automatic update settings were a prevalent source of misunderstanding for our participants. There are four possible settings in Windows Update: 1)

| *Consistent* | | *Inconsistent* | |
|---|---|---|---|
| Changed Setting | 4 | On, but thinks Off | 4 |
| Default Setting | 8 | Off, but thinks On | 2 |
| | | Download but not Install | 5 |
| | | Notify, but not Download | 14 |
| Total | 12 | Total | 25 |

Table 1

*Misunderstandings of Automatic Updates (Number of participants)*

*On*, the default setting where Windows automatically downloads and installs updates according to the process described in Section  (31 participants had this setting), 2) *Download* available updates but do not install them (0 participants), 3) *Notify* the user when updates are available, but do not automatically download or install them (4 participants), and 4) *Off*, where Windows Update must be manually run for anything to happen (2 participants).

Among our 37 participants, 25 had some form of inconsistency between what they stated they thought their computer's auto-update setting was, and the recorded settings on the computer (See Table 1). Of these, five participants were close to correct: they thought that their computer automatically *downloaded* updates and prompted them to install. While this is true, their actual setting automatically installs the downloaded updates at 3am if the user hasn't already installed them; these five participants frequently installed their updates proactively so rarely encountered the 3am automatic install.

This leaves 20 participants who had an inconsistency in their understanding of their auto-update setting. Four participants believed that their auto-updates had been turned off, when in reality they had the default, secure setting of automatically installing updates. Two participants believed the opposite; they thought they had auto-updates turned on, but auto-updates had been disabled on their computer[2]. The remaining 14 participants expressed a belief that automatic updates only notify them about available updates but do not install them. However, these 14

[2] One of these participants may be running a third-party updating system designed for pirated Windows systems.

participants all had the default setting of automatically installing updates. For example, Justin[3] told us “I mean it usually prompts me when there is an update to be installed, but I don’t know if that means auto-update or not.” His survey answers also indicated that he thought that Windows notified him, but did not install updates.

As a comparison case, 12 participants were completely consistent in their understanding of auto-updates. Eight had the default setting, and correctly understood that setting as automatically downloading and installing updates. Rachel said, “ I guess my current belief is that the operating system doesn’t give you a choice about updating things, it just does it for you.” And four participants had intentionally changed the setting to *Notify Before Download* (i.e., the computer notifies the user that new updates are available but does not download or install them), and also correctly understood their change.

Six participants’ computers did not have the default auto-updates setting, *Scheduled Install*, in which software updates are automated as much as possible. Two of these participants didn’t understand the setting and thought they were still on. However, the remaining four participants correctly understood that their computers would not automatically install updates. An additional 14 participants, who had the default setting of *Scheduled Install*, believed that they were only notified about updates and that no updates were installed automatically. These findings indicate that many misunderstandings exist regarding whether users are updating Windows, and that sometimes these misunderstandings mean that updates are not installed.

***Timing of Update Installation.*** The timing of updates is another source of inconsistency between participants’ stated intention and log data. Common security advice is that software updates, and particularly security updates, should be installed as quickly as possible to protect against in-the-wild exploits and zero-day vulnerabilities (Symantec, 2013). However, installing software updates usually interrupts what the user is doing on their computer, and often requires a severely disruptive reboot (Vaniea et al., 2014).

[3] All participant names have been anonymized.

In our log data analysis, we characterized each update as either *proactive* or *automatic* depending on if the user proactively installed the update, or if Windows automatically installed the update. Each participant, then, made a series of choices that either resulted in the participant installing most of their updates proactively, or mostly allowing Windows to automatically install.

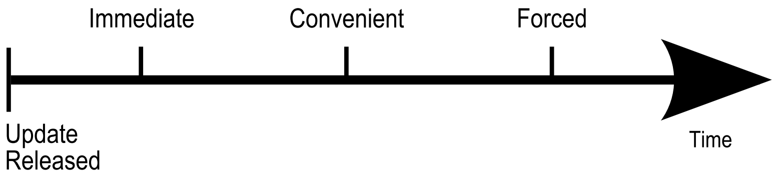

*Figure 2.* Perceived Times When Updates Can Be Installed

However, participant understanding of update timing doesn't exactly match this characterization. Instead, we found three timing categories for when updates might be installed (See Figure 2). The fastest possible update installation happens when a user is notified about an available update, and interrupts what they are doing to *immediately* and manually install the update. An intermediate timing occurs when a user is notified about an update, but doesn't interrupt their work to install it immediately. Instead, they wait until a *convenient* time to manually install the update. Both these categories involve manual installation, though some users may not find convenient times and end up with Windows automatically installing some updates. Finally, the slowest timing that actually results in the update being installed corresponds with the *forced* timing, and occurs when the user waits too long and the computer automatically installs the update and reboots the computer.

This difference in technical coding and user understanding poses an analysis challenge: when a participant indicates that they install their updates "when convenient," how do we characterize whether their behavior is consistent with their understanding? To address this, we first looked at the logs for whether most of an individual participant's updates were automatic or manually installed. If updates were mostly automatic, then that is a clear disconnect from the participant's stated understanding of installing when convenient; since the automatic install happens as pre-specified times, it is unlikely that that is happening "when convenient."

However, if the participant mostly installed updates manually, then this could be consistent with a desire for convenience (if they waited until it was convenient to install and reboot), or it could be inconsistent (if they interrupted themselves to install the updates). Since whether or not a participant was interrupted is entirely in the opinion of that participant, we looked to the survey data for guidance on how to categorize them. On the survey, we asked each participant how likely they would be to interrupt themselves to install Windows updates. Consistent with traditional interpretations of similar Likert scale survey questions (Dillman et al., 2009), we took this question to represent the participant's memories of whether they were frequently interrupting their work to install updates. If they answered "Likely" or "Very Likely", then we took this as inconsistent with their stated desire for convenience. Any other answer was considered consistent.

| *Consistent* | | *Inconsistent* | |
|---|---|---|---|
| When Convenient | 8 | Want Convenience, but Automatic | 8 |
| | | Want Convenience, but Proactive | 6 |
| Wait till Forced | 6 | Thinks Delay, Installs Proactively | 2 |
| | | Wants Only AV updates | 2 |
| | | Turned Auto-updates Off | 1 |
| Total | 14 | Total | 19 |

Table 2
*Inconsistencies in Timing of Update (Number of participants). We excluded four participants from the table due to insufficient information.*

**Results**: Nineteen of our 37 participants expressed a desire about the timing of updates that was inconsistent with the log data on their computer. Of these, ten participants installed updates more quickly than their stated intention, and nine participants installed updates more slowly (Table 2). Four participants had insufficient interview data to accurately judge their desires.

Twenty-two participants stated that they wanted to install updates manually at a convenient time; however, eight of them never actually got around to running the updates and the computer ended up automatically installing the update — which means the participants installed updates slower than intended. Six participants interrupted their work and installed the updates very quickly. In Figure 2, all 22 of these participants' stated intentions were to install in that middle range of timing — when convenient. Eight actually installed at that time; eight actually installed when forced (to the right), and six actually installed immediately (to the left).

Two participants stated that they usually delay updates, particularly updates that require a restart. These participants, however, usually installed updates very quickly according to the logs. Three participants said they only do updates labeled "urgent"; two of them successfully installed all updates quickly, but one participant had auto-updates turned off and didn't install any updates.

When a participant has an inconsistency about when updates are being installed, this isn't a technical misunderstanding. Participants aren't misunderstanding how the computer is working. Rather, they are misunderstanding their own behavior. Such a misunderstanding is important because it can form the basis for further decisions, such as "is my computer secure?" But since it is not a technical misunderstanding, greater education will not necessarily solve it.

***Difficulty Understanding Updates.*** As indicated by the inconsistencies mentioned above, many of our participants misunderstood what was happening on their computers. In examining our interview data, we found two reasons they were having problems.

First, the computer wasn't very clear about what it is doing and when it is doing it. Many participants talked about how it was difficult to understand what was going on. Nicole, for example, could not tell whether she permitted her computer to automatically update or not:

> *"Actually I didn't know that I clicked yes for auto updating. It just popped up. So, that's why I know about the auto updating. And other stuff, I didn't know that I clicked yes for auto updating or something like that."*

In the interview, she indicated that she thought it was important to install urgent and critical updates, and in the survey she indicated that she thought her updates were automatically installed. However, her computer actually had automatic updates turned off.

Second, even when our participants tried to look at settings and dig deeper, they found most of the settings to be confusing and difficult to use. Matt said that he "[doesn't] even know where I'd go to do that." Will wanted to turn off automatic updates:

> *"But I know I played around with some of the settings on my computer so that it wouldn't automatically update everything. Because it would just slow down my computer to a crawl. And several computers that I've had, it makes it harder when you're trying to get a task done."*

However, Will's computer still had the default setting and all updates released had been installed. Furthermore, most of his updates were automatic installs, rather than being installed manually.

Many of these misunderstandings stem from design choices that try to remove the need for humans to make decisions about software updates. Windows Update has automated as much as possible and moved many updates actions into background, invisible processes. That automation made it difficult for many of our participants to understand what was happening on their computer at any time, and even whether updates were being installed at all. Additionally, to discourage users from changing settings, Windows Update makes it difficult for users to find the settings in the first place. So even if our participants did want to change the settings, they couldn't figure out how. Removing the participants' decision-making ability had the side effect of also making it difficult for them to learn about updates and understand what their computers were doing.

### Intentions and Security

In addition to describing their current understanding, our participants also described what they wanted to be doing about software updates. Did our participants intend to put off updates because they felt like updates weren't important, or did they intend to install them immediately but ended up delaying indefinitely? Here, we describe whether these stated *intentions* match what was actually happening on the computer. Mismatches between intentions and behavior indicate usability problems, or what would change if we made software updates easier to understand and use.

For this analysis, we consider installing updates to be secure, and installing them sooner is more secure than waiting and installing them later. While users may have good reasons to choose to be less secure, we focus primarily on the security consequences of those choices.

Two participants provided short answers during their interviews and did not clearly describe their intentions for what they wanted their computers to be doing. Therefore, these participants were removed from this analysis of intentions.

| *Consistent* | | *Inconsistent* | |
|---|---|---|---|
| Notify but not Auto-Install | 3 | More Secure | 12 |
| Not urgent, so wait till Forced | 3 | Less Secure | 9 |
| Always install Immediately | 8 | | |
| Total | 14 | Total | 21 |

Table 3
*Whether Intentions are Consistent with Reality (Number of participants)*

***When Intentions Don't Match Reality.*** Twenty one participants had a disconnect between their stated intentions for installing software updates and what the log data indicated their computer was actually doing (Table 3).

For nine of these participants, the computer ended up being less secure than the participant intended. Three participants intended to install updates regularly and automatically, but actually had their automatic updates turned off (or to notify) and had almost no updates installed on their computer. The remaining six participants all stated that they intended to proactively install updates as soon as it was convenient, but rarely actually got around to installing the updates until the computer automatically did so. This mismatch between intention and behavior led to the updates being installed, but left a larger window of vulnerability than the participant intended.

As an example, Dan talked about how he chose when to install updates:

> *“If I were doing something fun I would interrupt it, no problem. If I were just surfing the web, it’s like, oh, whatever, I’ll update my computer. But if I’m writing an email, if I’m working on a paper, if I’m working on a homework assignment, then that usually takes priority. If I can put it off for 15, 20 minutes, I’ll just do that later then, ’cause when I’m in the zone studying, I don’t wanna be interrupted with anything.”*

This is a typical representation of a “convenient” intention: he wanted to install updates, but didn’t want to be interrupted. So he said he’d finish what he was doing and then install the updates. However, Dan’s computer logs indicated that Windows Update automatically installed most updates; he rarely installed them manually. This means that his computer was vulnerable for the maximum amount of time that Windows Update allows.

Twelve participants had a disconnect between their stated intentions and the log data that left their computer more secure than they had intended. Two of these users explicitly stated that they wanted to turn automatic updates off, but their computer still had the default setting of automatically downloading and installing updates. Another example is a participant who wanted to continuously delay updates, indefinitely, but had the default auto-update setting that automatically installed updates in a relatively timely fashion.

One participant from this group, James, expressed an intention to delay updates until a convenient time, but always ended up interrupting what he was doing to manually install updates. He described one instance that illustrated his intention to install when “convenient”:

> *“What was I gonna do? I was working on homework for something and I was loading a video on my browser to watch while I ate food. It was buffering and loading, and I usually will take a meal break and watch a movie at the same time. And I realized if I restarted, then that would have to reload, the movie would have to reload all the way from the beginning. And I would lose that time because I was going to eat in 15 or 20 minutes and then I had to go somewhere, I had a class. So I decided, you know what, I’ll just postpone.”*

However, according to James’s computer logs, all of the updates on his computer were installed, and were installed manually in less than 24 hours after being downloaded. James actually interrupted his computer use at some point rather than postponing, and ended up with a smaller window of vulnerability than he would have if he had waited to install when convenient.

These disconnects are interesting when we look at what would happen if we improved the usability of software updates and did a better job of including the user in the loop. Nine of our participants' computers would be more secure if they were able to execute on their intentions, while twelve would be less secure. The sample for this study is not representative, so we cannot claim that these 21 out of 37 participants (59%) generalize to the larger population of computer users. However, our sample has a relatively large number of both people who would be more secure if usability improved, and a similar number who would be less secure if usability improved. We suspect that both groups are well-represented in the larger population.

***When Intentions Match Reality.*** Fourteen of our participants were able to successfully execute on their intentions: the log data from their computer was consistent with these participants' stated intentions for software updates. However, these participants had varying levels of security.

Eight participants fell into the most secure category; these participants all had the default setting that automatically downloads and installs updates. These participants felt strongly that installing updates is important, and manually installed updates soon after they were notified that the updates were available. These participants didn't wait for the computer to automatically install the update. By manually installing the update, they minimized the window of vulnerability.

Three participants had a strong objection to the way that Windows compels the computer to reboot; these participants felt rebooting seriously interrupted their work. These participants changed their settings so that Windows notified them that updates were available, but did not download or install them. They manually downloaded and installed updates at a convenient time. Everyone in our study who had changed their auto-update setting to *Notify Before Download* or *Notify Before Install* fell into this group; people who change this setting seemed to understand that updates are important and still install them, but not as quickly.

Finally, three participants didn't feel like updates were that important, and wanted to have the computer deal with the updates for them. They continually postponed updates until the computer automatically installed the updates, and rebooted their computer.

***Would Better Usability Be More Secure?.*** Many people in the HCI community emphasize usability; if we make computers easy to walk up and use, then people will be able to accomplish more with them. When people form intentions about what they want their computer to do, but cannot execute on those intentions, HCI professionals naturally suspect a usability problem. Indeed, Windows Update seems to have a usability issue; 21 of our 37 participants (approximately 59%) were not able to use the system the way they wanted to.

However, it isn't clear whether better usability would actually be an improvement in this case. Only 9 of 21 participants whose behavior did not match their intentions were less secure than they wanted to be; these participants would end up more secure if we were to improve usability. But for the remaining 13 participants whose behavior did not match their intentions, the computer was more secure than it would be if usability were improved. These participants wanted to be less secure, and poor usability was preventing them from executing on that intention.

Many of our participants had misunderstandings about what their computer was doing with software updates. And many of our participants had trouble executing on their intentions. One reasonable assumption is that the second statement — the difficulty in executing on intentions — is caused by the first. However, we don't believe this is the case. A couple of participants completely understood what their computer was doing, but still could not execute on their intentions. For example, Rachel understood that the computer was installing updates, but felt like auto-updates were controlling her and forcing her to install them. And there were many participants who didn't understand what their computer was doing, but ended up doing exactly what they wanted to. Brittany believed that her computer only notified her but didn't install updates; however, she wanted to control her updates and ended up installing almost all of her updates manually at convenient times. It seems that understanding is not necessary to be able to execute on security intentions.

### Why do users avoid software updates? [1]

Participants also discussed updating third party software installed on Windows. Similar to Windows updates, participants' understanding of how the software update process worked had minimal impact on the updates that were ultimately installed on their computer. Additionally, some software was more likely to be updated than others. Java – a program other programs need to function – was fully updated on only 28% of participants' computers, while iTunes – a multimedia player – was updated on 67% of participants computers.

| | Not installed | Updated | Not Updated | % Updated |
|---|---|---|---|---|
| Java | 5 | 9 | 23 | 28% |
| Adobe Reader | 7 | 22 | 8 | 73% |
| iTunes | 13 | 16 | 8 | 67% |

Table 4
*Update status for common software installed on participants' computers.*

Our participants talked about three overall themes for reasons they did not install software updates: trepidation about surprise new features in an update; the difficulty of assessing whether an update is "worth it" given uncertainty about what a program does and why it is needed; and confusion about why an update is necessary, if the program seems to be working fine. We discuss each theme in the context of a commonly mentioned piece of software that illustrates both the reason for not updating and the security implications. We also present results from the analysis of installed updates and survey data.

**Surprise UI Changes: "Just, like, leave it alone...".** User interface changes were disruptive for respondents' workflows, and were commonly mentioned. They found new interfaces "annoying" to learn, and reported that they avoided updating software that had a history

[1] The majority of this section was previously published in the ACM conference on Computer Human Interaction 2014 (Vaniea, 2014).

of frequently changing the user interface. Nick discussed how new user interfaces were hard to learn and made him reluctant to update:

> *"I just hate the idea of having to just relearn everything, where everything is, how to access 'help', something as basic as that. And when the new Office suite came along with Excel especially. Word, bad; Excel, worse, as far as just being completely different, counter-intuitive. I try to stick with what I know rather than just take the time to relearn a whole new system."*

In addition to being annoying, user interface changes were perceived to have an immediate, negative impact on productivity. Kim talked about how she had "been working with these programs for 15 years. I know them like the back of my hand." To interact with the programs quickly Kim learned the hot keys, specific combinations of keyboard keys that can be used instead of clicking buttons. However, "every now and again, you get a new version and they change the hot keys."

***Case: iTunes 11.*** The impact of interface changes on participants' willingness to install current and future updates was particularly highlighted by changes Apple made in version 11 of iTunes, their multimedia player. In iTunes version 10.7 the user interface had a navigation sidebar permanently visible on the left side of the screen. However, version 11 removed the navigation, and the user interface became modal with different available actions depending on the type of content being viewed.

The version 11 update was very unpopular with our participants. Ashley expressed a common reaction: "all of a sudden, like, *Where did my things go?*" Participants installed the update not realizing that the user interface was going to change, and then became upset when it did. This caused some participants to become wary of iTunes updates in general. Amber said that she had "ignored [iTunes updates] quite a few times because I'm like: *I don't need that update*." Rachel expressed worry about future iTunes updates:

> *"I also always worry that everything is gonna get screwed up, especially for iTunes*

> *updates or things like that because they're always reconfiguring the layout of stuff, and I'm like, 'I don't want you to do that. Just, like, leave it alone...'"*

Some participants had learned from past experiences to check technology blogs and forums as a way of finding out what new updates might do before installing them. Melissa learned that the iTunes update would change the interface and explained that she had waited to update iTunes because she "was not ready to get used to [the new interface]." Chris talked about deciding not to update iTunes after reading that other people didn't like the new interface.

> *"One of the prior iterations of iTunes, it just wasn't well received and there were some... A lot of complaints about the new version. People saying, 'Don't update'. I'm like, okay, I won't update."*

Other participants had less foresight and ended up installing an unwanted iTunes update. In Lauren's case, an update disabled her ability to manage her old iPod. After the loss, Lauren began refusing all iTunes updates because she was "mad at them."

> *"You can't update it and and you can't change the songs because the new version of iTunes is like, 'Even though you took really good care of your machinery, we don't want you to have that one anymore.'"*

On the surface, iTunes might not seem like it would be associated with security issues, and none of our participants mentioned any concern about security in relation to it. However, software that displays web pages (HTML/Javascript) is the most common vector for compromise on Windows computers (Microsoft, 2012) and iTunes displays web pages as part of the iTunes store. In its version 10.7 release of iTunes, Apple patched 163 vulnerabilities, the majority of which involved the web page display functionality. Most users don't browse the web using iTunes; rather, they use a web browser. But when they click on specially formatted links it can cause iTunes to automatically launch and attempt to display the web content as part of the iTunes store. If a user running the 10.6 version of iTunes were to click the wrong link it could be used to install any software on the user's computer[4].

[4] http://www.zdnet.com/google-helps-close-163-security-vulnerabilities-in-itunes-7000004186/

In the survey we asked participants what parts of the update process for multimedia players were generally automated and what parts were user controlled. We then compared their answers to the installation status of iTunes (Table 5). We found that 8 of the 24 of users with iTunes installed had not updated to the latest version. Participants understood that multimedia players generally check for updates automatically but wait for user approval before installing updates.

| *Auto multimedia update behavior (Survey)* | *iTunes State (Logs)* | | | |
|---|---|---|---|---|
| | Not installed | Not updated | Updated | Total |
| Not installed | 2 | - | - | 2 |
| Auto checks for updates | 10 | 8 | 15 | 33 |
| Auto installs updates | - | - | 1 | 1 |
| Not automatic | 1 | - | - | 1 |
| Total | 13 | 8 | 16 | 37 |

Table 5
*Multimedia update survey answers versus observed state of iTunes (Number of Subjects).*

Participants talked about the need to update web browsers like Firefox, and Chrome as a way of protecting themselves from dangers on the internet. But they reported thinking about iTunes as a multimedia player that plays their songs, videos and interacts with their Apple devices. Their update choices were based on how they wanted to interact with the software and the functions they needed.

**I Don't Understand It, so I Won't Update It.** Participants differentiated between programs they used regularly and programs they used infrequently, or not at all. They were more inclined to update software they used frequently because they recognized that doing so would bring them the latest features and make it easier to interact with other people who were running updated versions of software. They were less inclined to update rarely used software without a good reason. If a program stopped being used entirely, they tended to either stop updating it or uninstall it completely. Brandon talked about how update prompts caused him to either remove the program, or find a way to turn off updates.

> *"If I'm not using the program . . . either the software update will prompt me to go remove the program because I'd say, "I am not using and just get rid of it." Or, somehow turn off the preferences to say "Don't remind me to update this" 'cause I am not using the program."*

***Case: Java.*** Java in particular was problematic for our participants, because they didn't understand what it was, and they didn't think they used it. Java is a program that provides functionality to other programs installed on a computer. Users rarely, if ever, directly interact with Java even though they may frequently use programs that need Java to function. Amy's experience highlights the confusion:

> *"It's annoying and I don't think I need Java, so I just deleted the program. However, when I visit some website they asked me to install Java. 'Okay. I will install it, but if you ask me to update again, I will delete you.'"*

To correct several serious security issues, four Java updates were released in six weeks in the beginning of 2013 just before our study. Typically Java releases updates about once every two months, and this escalation may have contributed to our participants' irritation. Some participants had formed a general animosity against Java because of the constant requests for updates. They reported feeling that Java wanted to update "all the time" and was really annoying. Lauren didn't understand why she needed Java, and got irritated by the repeated requests to update:

> *"I don't know why the hell I need a Java so I ignore it... I'm just pissed off and I think I have a tendency then, when like I see Java pop up in the corner, I'm like, 'Fuck you, Java.'"*

When participants became confused by repeated requests for Java updates they said they went online and attempted to understand what the updates were doing. Because Java is well known as being vulnerable to security compromise some online forms advise uninstalling it. However, it is necessary for many programs to function correctly, so other forums recommend updating it regularly. Participants searching for information on Java encountered conflicting discussions similar to the rhetoric described above and became confused about what to do. Ashley talked about such an event:

> *"I started to think there was a problem, and so then I started looking more, and actually reading to try to decide, do I wanna install the update? You know, some things, [you should] install the update because it will make it better. Other things are like... Just take it off your computer completely."*

Because it runs invisibly in the background, our participants only saw that Java updates cost them time. They did not understand that updating Java made other programs potentially run better, or that not updating Java made them vulnerable to attackers. Java is the second most common source of security compromise on Windows computers, and one of the least updated programs with only 6% of Windows computers running the latest version (Microsoft, 2012).

In the survey we asked participants what parts of the Java update process were automated, and which required user involvement. Most participants with Java installed (93%) correctly thought that it automatically checked for new updates but did not automatically install the updates (Table 6). However, out of the 32 participants with Java installed only 9 (28%) had fully updated it. No one with Java installed believed that the computer would automatically install the updates for them. This suggests that the 72% of participants who did not update Java did so by choice. Participants were also confused as to whether they had Java installed or not, 2 participants reported Java being installed when it was not, and 7 reported it as not being installed when it was.

| *Auto Java update* | *Java State (Logs)* | | | |
|---|---|---|---|---|
| *behavior (Survey)* | Not installed | Not updated | Updated | Total |
| Not installed | 3 | 7 | - | 10 |
| Auto checks for updates | 2 | 14 | 9 | 25 |
| Auto installs updates | - | - | - | 0 |
| Not automatic | - | 2 | - | 2 |
| Total | 5 | 23 | 9 | 37 |

Table 6
*Java software update survey answers versus observed state of Java software (Number of Subjects).*

**If it Ain't Broke, Don't Fix It!.** Participants explained that if their software was working and fulfilling their needs, they saw no need to make changes. They were reluctant to expend effort and risk problems just to change the behavior of functional software.

As described above, participants felt updating software is potentially fraught with uncertainty. Most updates provide little to no information to end users about what will occur when they click the "Install" button. In addition to the risk of user interface changes, there is also a risk that a needed feature will be removed from the software, or that the software will stop functioning entirely. When faced with the choice to update users have three options: blindly accept, research, or deny. Accepting the update carries the potential cost of installing an update that has unwanted components, and researching the updates costs time and mental energy, so some participants chose the least risky option of deny.

Participants talked about using software until it became non-operational, and then either updating the program or deleting it. Andrew talked about not updating software:

> *"Many times I do not update. Just for regular software unless I feel that this software now is not working properly. Otherwise, I'll keep it simple."*

Participants also made a distinction between "regular" software which didn't need to be updated and security software such as anti-virus. Nick talked about how he kept security software up-to-date, but avoided other updates:

> *"I feel like if I'm really used to the software I'm using and I think it's meeting my needs I won't upgrade the software. But if it is really important like anti-virus it has to be upgraded."*

***Case: AdobeReader.*** AdobeReader, a PDF viewer, was an excellent example mentioned by our participants of a utilitarian software program that had a single clear function, and no obvious link to security. It is also the 3rd most common vector for computer compromises (Microsoft, 2012). Participants were puzzled about why AdobeReader needed to be updated at all. Justin explained that he never updated AdobeReader because there was no need to do so: “I just don’t see what an update to [AdobeReader] can do. I mean it’s PDF files. Its viewing them...” David explained how he didn’t “listen” to AdobeReader update requests, because the current version met his requirements: “Adobe, current version helps me to read. And so that’s how I decided my requirements.” Participants saw AdobeReader as fulfilling a specific function, and if it was still functioning they saw no reason to change it.

Participants talked about updating AdobeReader when it stopped functioning. Part of Mike’s job involved downloading PDF files from websites and modifying them. If he did not have the latest version of AdobeReader he could have issues: “If I try and download something that is from a more advanced version, it won’t accept it. It’ll just die.” Nicole also discussed not updating programs like AdobeReader, but when the program stopped opening files she would find and download a new program that wasn’t “broken.”

Document viewers, and AdobeReader in particular, are the third most common source for computer compromises on Windows computers (Microsoft, 2012). Participants thought of document viewers as simply displaying static information; however, they are actually similar to web browsers in that they change the stored information into something visible to the user—a process which can result in security issues.

In the survey we asked participants what part of the update process for Adobe products were automated, and which required user involvement. Most participants with Adobe Reader installed also believed that Adobe Products automatically checked for new updates but did not automatically install the updates (Table 7). Starting with version 10.0.1 Adobe Reader changed its default setting from automatically checking for updates but not installing them to automatically checking for and installing the updates. Of the 22 participants with an auto-installing version of Adobe Reader, 21 had fully updated software. Of the remaining 9 participants with software that checks for updates but does not automatically install it only 1 was fully updated.

| *Adobe product update* | *Adobe Reader State (Logs)* | | | |
|---|---|---|---|---|
| *behavior (Survey)* | Not installed | Not updated | Updated | Total |
| Not installed | 1 | - | 1 | 2 |
| Auto checks for updates | 6 | 7 | 19 | 32 |
| Auto installs updates | - | 1 | - | 1 |
| Not automatic | - | 1 | 1 | 2 |
| Total | 7 | 9 | 21 | 37 |

Table 7
*Adobe product software update survey answers versus observed state of Adobe Reader (Number of Subjects).*

## Discussion

Our subjects had a number of misunderstandings about what their computers were doing with respect to software updates. Also, our subjects frequently were not able to execute on their intentions about whether and when to install software updates. We speculate that these challenges may be the result of trying to remove the human from security decisions. We also observe that improving usability may actually backfire.

***Learning Through Decisions.*** In designing security technologies, there is a tension between removing human decisions to automate security, and allowing the user the flexibility to make important choices (Cranor, 2008). The current version of Windows Update represents a compromise; most of the decisions about updates are made by the computer, removing the human from decision making. Many updates are downloaded and installed automatically, and Windows eventually automatically installs all downloaded updates even when they require a reboot. Some human decisions remain, particularly when they impact use of the computer, such as rebooting.

Removing the human from Windows Update decisions, however, seems to have had an unintended side effect: users now find it difficult to understand what the computer is doing, and to correctly implement their part of the updates process. Having to make decisions as part of a security mechanism helps the user to learn how that mechanism works, what decisions are appropriate, and how to correctly execute those decisions. This learning may be direct, coming from feedback within the system. Or, this learning may be indirect learning, with the user seeking out the knowledge necessary to make better decisions. Windows Updates has successfully automated so many security decisions that many users don't learn how to make intelligent security decisions about software updates. Instead, they struggle at understanding what their computer is doing, and often fail to execute even when they do make a decision.

The invisibility of security issues is also apparent in the choices users make when deciding not to update installed software. It is not clear to users how programs such as Adobe Reader and iTunes relate to the security of their computers leading users to focus on more visible issues such as the user interface. The lack of association with security also makes it challenging for users to associate the cause of a security breach (opening a PDF file with an outdated copy of Adobe Reader) and the effect (loss of credit card information).

This is important when some, but not all, security-relevant decisions can be automated. Removing the user from most of the decisions makes it more difficult for the user to intelligently make the remaining decisions that cannot be fully automated.

***Designing Update Systems.*** There is a fundamental tension here between learning and understanding what the computer is doing, and improving security by forcing the user to behave securely. It isn't clear which is a better strategy. Consider just the results in this paper: if usability were improved and users were able to accurately execute on their intentions, some users would end up less secure but many would end up more secure. The net effect on security isn't clear; it is possible that ignorance and inefficacy might be better for security than learning and usability.

There is also a tension here among the users. Some users want to trust the computer and software companies to make good decisions for them. For these users, automating good decisions is valuable. However, other users want control over their computer, and rebel against the feeling of being forced into doing things they don't agree with or that impact their workflow.

The software industry is currently struggling with these tensions. Windows Update is clearly moving toward automating as much of the software update process as possible. A wide variety of other system applications are following. Firefox, for example, now automatically downloads and installs updates with virtually no user intervention (Mozilla, 2014). Java is also moving toward automatically installing updates, and Adobe is moving to a subscription model with automatically installed updates and upgrades. Apple's iOS 7 and OSX Mavericks now allow users to turn on a setting to automatically install updates to all software installed via the official App Stores.

However, some end-user "apps" and most business applications are moving to a much more explicit, user-driven update model. Some smartphones, for example, require the user to explicitly check for updates and choose to install them.

Almost all software on PCs eventually requires software updates, and many of these updates are security relevant. Each software vendor makes choices about how to distribute these updates. Our results suggest that automating updates similar to Windows Update or Firefox will lead to more uniform update installations, but will also result in many users not understanding what is happening on their computers and not being able to change things when they want to. On the other hand, manually installing updates may lead to better understanding about updates and greater feeling of control, but will also likely result in lower levels of security and compliance.

# References

Adams, A. & Sasse, M. A. (1999). Users are not the enemy. *Communications of the ACM*, *42*(12), 41–46.

Besnard, D. & Arief, B. (2004). Computer security impaired by legitimate users. *Computers & Security*, *23*(3), 253–264.

Bilge, L. & Dumitras, T. (2012). Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the acm conference on computer and communications security* (pp. 833–844). New York, NY, USA.

Brewer, D. D. (2002). Supplementary interviewing techniques to maximize output in free listing tasks. *Field Methods*, *14*(1), 108–118.

Cranor, L. F. (2008). A framework for reasoning about the human in the loop. In *Usability, psychology, and security (upsec).*

Dillman, D. A., Smyth, J. D., & Christian, L. M. (2009). *Internet, mail, and mixed-mode surveys: the tailored design method* (3rd). Hoboken, NJ: Wiley.

Dourish, P., Grinter, R. E., Delgado De La Flor, J., & Joseph, M. (2004). Security in the wild: user strategies for managing security as an everyday, practical problem. *Personal and Ubiquitous Computing*, *8*(6), 391–401.

Edwards, W. K., Poole, E. S., & Stoll, J. (2007). Security automation considered harmful? In *Proceedings of the new security paradigms workshop, nspw* (pp. 33–42).

Furnell, S. (2005, June). Why users cannot use security. *Computers & Security*, *24*(4), 274–279.

Gkantsidis, C., Karagiannis, T., & Vojnovic, M. (2006, August). Planet scale software updates. In *Acm sigcomm computer communication review* (pp. 423–434). New York, New York, USA: ACM.

Kaemer, S. & Carayon, P. (2007). Human errors and violations in computer and information security: the viewpoint of network administrators and security specialists. In *Applied ergonomics* (Vol. 38, pp. 143–154).

Kainda, R., Fléchais, I., & Roscoe, A. W. (2010). Security and usability: analysis and evaluation. In *International conference on availability, reliability, and security, ares* (pp. 275–282). IEEE.

LaRose, R., Rifon, N. J., & Enbody, R. (2008, March). Promoting personal responsibility for internet safety. *Communications of the ACM*, *51*(3), 71–76.

LaRose, R., Rifon, N., Liu, S., & Lee, D. (2005). Understanding online safety behavior: a multivariate model. In *The 55th annual conference of the international communication association*. New York City.

Marconato, G., Nicomette, V., & Kaaniche, M. (2012). Security-related vulnerability life cycle analysis. In *Risk and security of internet and systems (crisis), 2012 7th international conference on* (pp. 1–8).

Microsoft. (2012, January). Microsoft Security Intelligence Report, Volume 13.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2013, April). *Qualitative Data Analysis*. A Methods Sourcebook. SAGE Publications, Incorporated.

Mozilla. (2014). Update firefox to the latest version. Retrieved from https://support.mozilla.org/en-US/kb/update-firefox-latest-version

Onwuegbuzie, A. J. & Leech, N. L. (2007). Validity and qualitative research: an oxymoron? *Quality & Quantity*, *41*(2), 233–249.

Symantec. (2013). Internet Security Threat Report, Volume 18.

Thaler, R. & Sunstein, C. (2008). *Nudge: improving decisions about health, wealth, and happiness*. Yale University Press.

Vaniea, K., Rader, E., & Wash, R. (2014). Betrayed by updates: how negative experiences affect future security. In *Proceedings of the acm conference on human factors in computing (CHI)*. Toronto, Canada.

von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003). CAPTCHA: using hard ai problems for security. In *Eurocrypt '03* (pp. 294–311).

Wash, R. (2010). Folk models of home computer security. In *Proceedings of the symposium on usable privacy and security (soups)*.

West, R. (2008). The Psychology of Security. *Communications of the ACM*, *51*(4), 34–41.

Wikipedia. (n.d.). Windows Update. http://en.wikipedia.org/wiki/Windows_Update; last retrieved September 17, 2013.

Yee, K.-P. (2002). User interaction design for secure systems. In *International conference on information and communications security, icics* (pp. 278–290).

Zurko, M. E. (2005). User-Centered Security: Stepping Up to the Grand Challenge. In *21st annual computer security applications conference (acsac'05)* (pp. 187–202). IEEE.