Why Johnny Can't Blow the Whistle: Identifying and Reducing Usability Issues in Anonymity Systems

Greg Norcie
Indiana University
gnorcie@indiana.edu

Jim Blythe
Information Sciences Institute
blythe@isi.edu

Kelly Caine
Clemson University
caine@clemson.edu

L Jean Camp
Indiana University
ljcamp@indiana.edu

***Abstract*—Tor is one of the most widely used anonymity networks in the world. The anonymity provided to any individual Tor user is a function of the total number of Tor users. Adoption of anonymity tools, including Tor, can be hindered by a lack of usability. Therefore, enhancing Tor's usability and making it easier for more people to use Tor successfully increases security for all Tor users. To enhance Tor usability we begin by identifying stop-points which may act as barriers to using the Tor Browser Bundle (TBB). We suggest changes based on our analysis of these stop-points. To follow through, in our second study we test whether the changes we recommended to the Tor Project are effective in increasing usability, and find a significant decrease in stop-points across the board. Based on these studies, we both suggest new design heuristics appropriate for improving the usability of anonymity systems and identify design heuristics which are particularly salient in the design of anonymity systems.**

I. Introduction

Communicating privately on the Internet is an increasingly difficult task. People who want to communicate privately, such as whistleblowers, dissidents, journalists, patients with sensitive health conditions, and/or domestic violence victims often use anonymity tools to consume online information without revealing their identities to those who might surveil them.

Tor [17] is a widely used anonymity tool that works by passing traffic through a series of nodes to obfuscate its source. Experts such as Bruce Schneier [1] recommend utilizing Tor to combat government surveillance. Due to Tor's nature it is impossible to precisely count the number of Tor users. However, conservative estimates [3] show that as of this writing (January 2014), upwards of 4 million people are connecting to the Tor network daily.

USEC '14, 23 February 2014, San Diego, CA, USA

http://dx.doi.org/10.14722/usec.2014.23022

Tor uses onion routing [24] to prevent third parties from observing or transparently interjecting themselves into a user's web activity. The term "*onion routing*" refers to Tor's layered encryption, analogous to the layers in an onion, which prevents third parties (including Tor nodes) from observing the content of packets in transit. Tor also prevents the linkage of the source and destination of any packet in the Tor network. Of course, there are attacks that threaten privacy even when using Tor, including subversion via control of malicious nodes and timing attacks. However, the largest challenge facing Tor arguably is not timing attacks, but rather the relatively small number of Tor users. In terms of providing anonymity, more users is better.

As Dingledine and Mathewson [16] point out, there is a strong network effect present in anonymity networks like Tor. In any system like Tor where anonymity is measured as $\frac{1}{N}$, additional users increase the anonymity of the system. The concept of k-anonymity [40], which states that anonymity is dependent on the ability of a person being distinguished from at least k - 1 other people, is a related measure. Thus, reducing usability barriers to the use and adoption of anonymity systems like Tor increases the anonymity of all users of the system and therefore improves the privacy of the entire network. While many papers have examined the technical aspects of Tor's security [35], [33], [32] or ways to avoid privacy issues in general purpose software [31], the scholarship on the usability of Tor[12] is more limited,as is the usability of security and privacy systems, especially anonymity systems in general [44], [4]. Our work is further distinguished in that we do not merely identify usability flaws, but we also suggest solutions, then test these implemented solutions, as well as derive design heuristics for anonymity systems based on our results.

Tor can be run in a variety of configurations. Previous work by Clark et al. [12] presented a cognitive walkthrough of several types of Tor software and offered a set of suggestions about the most usable version of Tor. Clark et al. found that a self-contained, simple-to-deploy browser bundle that required no configuration was the most usable. Clark's discussion of the usability of the Torpark echoed Lederer et al.'s [31] discussion of common privacy pitfalls consisting of common mistakes present in the design of software for non-experts. For example,

the pitfall *"Inhibiting established practice"* may be relevant to general purpose browsers, but since most users do not have established practices for use of anonymity systems, they have no prior knowledge to bring to bear. Lederer et al. noted the pitfalls of *"emphasizing configuration over action"* and *"lacking coarse grained control"* in their work. The Tor Browser Bundle's (TBB) initial design performs well in both of these areas. The TBB requires no configuration and simply requires a user to unzip a binary to run the bundle, thus avoiding these major pitfalls.

Clark et al.'s results suggested that Torpark, a Firefox variant with Tor incorporated into the browser, was the most usable instantiation of Tor. Subsequent to Clark et al.'s study, the Tor Project later released an official TBB, which contains a standalone Firefox variant with Tor built in. However, no subsequent studies until now have examined its usability.

Our work complements the existing work on improving the usability of Tor and extends it by identifying "*stop-points*" in the installation and use of the TBB. Stop-points are places in an interface where a user may find themselves facing a requirement for user action such that the user is unable to proceed [20]. However, a stop-point does *not* necessarily prevent a user from going forward — any point at which the user is going forward, but is confused about the functionality or state of the system is also a stop-point. In a previous examination by Whitten et al. [44], stop-points in Pretty Good Privacy (PGP) were found to be a result of the assumptions about the technical expertise of the user; in particular users were required to understand the concept of cryptographic keys beyond the metaphorical level.

In this paper we describe two studies. In Study 1 we evaluate stop-points during both the installation and the use of the TBB. Based on the results of Study 1, the authors identify changes that could be made to the TBB interface. We then hypothesize a set of design recommendations that could be applied not only to the TBB but that are also generalizable to other anonymity systems. The identification of usability issues, and corrsponding suggested changes as described in Study 1 were presented in preliminary form by Norcie et al [36] at the 5th Workshop on Hot Topics in Privacy Enhancing Technologies, a non-archival venue dedicated to research in progress.

After study 1 concluded, we then performed a second study (Study 2), which tests our usability suggestions and verifies that the design recommendations enhanced the usability of the TBB.

In the next section we describe related work. Given the breadth of the domains on which we draw, these descriptions are necessarily brief. Following this overview of related work, we describe the methodology and results of Study 1 followed by those of Study 2. We then discuss the findings of both studies, offer our conclusions, and finish with a discussion of future work.

II. Related Work

There are three bodies of literature that inform this work: laboratory evaluation of usability (particularly for security and privacy), privacy-enhancing technologies, and delay tolerance/decision making.

A. Usability and Usable Security

Usability design heuristics predate the field of usable security by many years. Molich and Nielsen [34] wrote a widely cited set of design heuristics for human-computer interaction. Later, Lenderer et al. [31] described a set of "privacy pitfalls" to be avoided when designing general purpose software.

Starting in the late 1980s [28] and throughout the 1990s [45] many of the seminal works on user-centered security were published. Whitten and Tygar's "Why Johnny Can't Encrypt" [44] described a cognitive walkthrough and lab-based usability test examining the usability of PGP 5.0. Whitten and Tygar concluded that the usability of security software requires a different standard of usability than other software. Specifically, they suggest that it is necessary for users to be aware of the tasks they need to perform, are able to successfully perform said tasks without making dangerous errors, and are comfortable enough with a security interface to continue using it. The major issues Whitten and Tygar noted are lack of incentive alignment, lack of feedback, and inability to recover from errors, also known as the "barn door property", evoking the futility of lockdown after loss of information.

Also in 1999, Adams and Sasse [4] argued that people are not careless with information protection (passwords in their study), but rather they are rationally allocating their own resources. Security requirements that are antithetical to human capacities cannot be met (i.e., choose many passwords impossible to guess and highly random, don't write them down, remember them, and change them often). Similarly, policies that conflict with work procedures or prevent completion of tasks are rejected.

Nearly concurrently, the field of economics of security was studying the issue of incentive alignment. Anderson [5] argued that lack of incentive alignment is a core problem in the design of security technologies, and Camp et al. [9] illustrated that it is economically rational to underinvest in security in the face of significant externalities imposed by proper security practices. These works all converged on the same point: that if users do not feel that security will provide them utility (benefit), they will not strive to improve their security. Thus we can conclude that simply improving usability will not make security usable — the developer must align their interface with users' incentives.

Other analyses of usable security or privacy include Inglesant and Sasse, who found that while individuals do in fact care about security, password policies are too inflexible to match human capabilities [27]. A follow-up study illustrated that graphical passwords had similar difficulties [8].

Maxion and Reeder [38] implemented a laboratory examination of usability of access control utilizing similar methods. As with our work, they determined that individuals who may

believe that they have implemented the correct settings are not consistently correct. Indeed, as few as 25% were able to complete a basic ACL task using XPFP.

While work on usable security is plentiful, work on usable anonymity is sparse.

One of the earliest attempts to improve Tor's usability was a GUI design competition[1] sponsored by the Electronic Frontier Foundation. This design competition resulted in the Vidalia and Torbutton projects, which (along with Tor and Firefox) serve as integral components in what would become the Tor Browser Bundle.

Also, it should be noted that a runner up in the competition called Foxtor [39] correctly pointed out that users may have issues knowing whether they are currently connected to Tor.

As mentioned earlier, based on results from a cognitive walkthrough of several early Tor interfaces, Clark et al. concluded in 2007 that Torpark (a self-contained browser bundle similar to the TBB our lab study evaluates) was the most usable option for novice Tor users [12]. Another related preliminary, non-archival report examined usability issues in the TBB [36], but did not empirically test their recommendations.

Thus, the fundamental challenges of security and privacy from a human-centered perspective are that individuals must be motivated to and capable of adopting the technology. Motivation must address rational investment, as well as the heuristics and biases of human decision-making [23]. Our investigation in usability of Tor is grounded in these previous works. Moving on, we will discuss works from the domain of privacy-enhancing technologies (PETS).

B. Privacy-Enhancing Technologies

The second significant domain of related work is that of PETS. The idea of passing an encrypted message between series of intermediate nodes was first discussed by Chaum [11]. The classic Chaumian mixnet is similar to the Tor network in that each packet is subject to a series of encrypting operations so that no node in the network can determine both the initiator and target of a communication. Similarly, each message has a theoretical requirement for three hops. However, in mix networks packets are delayed and reordered (thus preventing timing attacks at the expense of a latency.)

There have been several high anonymity, high latency systems such as MixMinion [14], Tarzan, [22], and Freenet [13]. However, none of these platforms ever gained popularity. Indeed, many of them never went beyond laboratory demonstration projects. Onion routing was first presented in 1998 [37], but Tor was not invented until 2004 [17]. Before 2000 the majority of anonymizing systems that were used in practice were single-hop proxies, for example, Anonymizer [7]. For cryptographic PETs, Tor is unique in its acceptability and adoption, with the number of users in the hundreds of thousands. The latest instantiation of Tor has usability as a design goal; the TBB combines Tor and the web browser one package. This simplified interface has the potential to expand Tor to a broader user base.

[1] http://tor.hermetix.org/gui/index.html

C. Delay Tolerance and Decision Making

As we discuss later in our results section (IV.C), many TBB users reported that the speed of the TBB's connection was disappointing. While there is considerable work on mitigating jitter and latency in Tor (eg. [43]), delay is an inherent price of Tor. Unlike issues such as icon salience, which was remedied by a quick change to the user interface by the Tor Project, there is no easy fix for browser lag in an anonymity network like Tor. Due to the nature of how anonymity networks function — passing traffic through a series of nodes — there will always be delays. As pointed out by Köpsell [29], decreasing latency in an anonymity network increases adoption of said system (and, conversely, increased latency produces an equal drop-off in user adoption.)

If it is not technically possible to eliminate latency, the next logical solution is to reduce the *perception* of latency. To understand how to convince users not to abandon Tor, we must look at the literature on delay tolerance and decision making.

Hertwig et al. described the effects of "framing" on decision making [25]. The researchers found that when given a description of a set of options, along with their associated probabilities, study participants overweighed rare events. However, when participants made decisions based on past experience, the participants *under*weighted rare events. This finding has relevance to security decisions, since most users make security decisions based on prior experience. For example, if a user has never experienced a negative result from clicking an e-mail attachment, they will underweight the risk that an email attachment could contain malicious code.

Security decisions, like all other decisions, are subject to the framing effect, a cognitive bias first described by Tversky and Kahneman [41]. The researchers found that study participants were risk seeking when it came to potential gains, but risk averse to potential losses. For example, when deciding on whether to vaccinate a large population against a deadly disease, the decision can be framed as a potential gain (citing the number of people to be saved by the vaccine), or as a potential loss (citing the number of people who would "certainly die" if the vaccine was not administered). While the basic probabilities remained the same for both scenarios, study participants were much more likely to choose vaccination when it is framed in terms of potential gain (lives saved).

Thus, we theorized when performing our studies that framing can assist us in helping users make better security decisions. This hypothesis is supported by previous literature. For example, Egelman et al. [19] examined whether users would tolerate security delays in a Mechanical Turk task. Turkers were told that they were testing a new web-based PDF viewer. Egelman et al. found that users were much more likely to cheat on the Mechanical Turk task when presented with either a non-security explanation for the delay, such as a simple loading bar, or a vague security explanation, such as changing the loading

bar to simply read "Performing security scan." Conversely, users were less likely to cheat when given a concrete security explanation — that online documents often contain viruses and that the PDF reader was performing a virus scan.

Moving on, we will discuss the methodology and results for our two studies, followed by an overall discussion of these results, closing with conclusions drawn from them and plans for future work.

III. Study 1: Identifying Usability Issues

A. Study 1 Methodology Overview

We recruited 25 undergraduates from a large North American university for Study 1. Students were given lab credit for participating in the study. Students who were not comfortable participating in the study were given the option instead to write a one-page essay on Tor's basic functionality. The entire study is IRB approved.

For Study 1 all participants were seated at a computer running Windows 7 and given an instruction sheet, as well as a short questionnaire where they were instructed to record any usability issues they encountered throughout the study. We also captured users' on-screen actions using screen capture software. The instruction sheet that was handed out provided users with the URL for the Tor Project (https://torproject.org) and instructed users to download the latest version of the TBB, run the TBB, and use the TBB to download a new desktop background for their lab machine. [2]

Prior to beginning the study participants were informed that the study was a usability evaluation and that we were evaluating the technology and not their abilities. Participants were also informed that the instructions they received were purposefully not step-by-step instructions and may therefore seem vague. If they were unclear how to proceed at any time, the participants were instructed to raise their hand so that the researcher could assist them. Participants were also given the definition of "stop-points" presented earlier in this paper. The participants were told that the lab was designed to find stop-points and that participants should raise their hands if they encountered a stop-point that they could not proceed beyond. Upon encountering a stop-point the researcher would then ask the participant to note this stop-point on their questionnaire. A post-task survey queried participants about whether or not they encountered any issues during installation and use. Participants who raised their hands during the study were instructed to note their issue on their post-task survey and then advised how to proceed past the stop-point.

Study 1 Demographics: In addition to collecting data on stop-points, our exit survey also collected demographic information from all participants. Participant ages ranged from 20 to 37, with a median of 22.7 and a mode of 21. Eighty-eight percent were male (22/25). Participants were asked if they had heard of Tor. The users were also asked to rank their familiarity with Tor, as well as their familiarity with computer security on a 1-to-7 scale (1 meaning "not at all familiar" and 7 meaning "very familiar"). While 84% (21/25) of users had heard of Tor, the users were by no means Tor experts. When asked "How familiar are you with Tor?", users responded with an average of 2.13 on a 7-point scale. Users were slightly more familiar with computer security. When asked "How familiar are you with computer security?", users reported an average of 4.5 on a 7-point scale.

[2]TBB v2.2.35-7.1 at the time of Study 1

B. Study 1 Limitations

Like many usability studies, participants in this study are not perfectly representative of the population as a whole. Our participants are more educated, more familiar with computer security, and more male than the general population. Normally, in research we strive to find representative users rather than expert users, because experts have experience that may predispose them to being able to use a system that non-expert users may find difficult to use. In this study, however, our participants were generally computer savvy and specifically security savvy. While this can be viewed as a potential limitation of the study, it may also be viewed as an advantage. Tor users, having sought out an anonymity system, are likely to have more expertise than an average person. Second, we can expect that any difficulties experienced by the expert users would also be experienced by non-experts (indeed, the same argument is made in many seminal works on usable security (e.g. [44]). Therefore, while this participant population is limited in that we will likely not identify all usability issues with the TBB that the population of potential users may encounter, we will likely uncover the *most problematic* usability issues.

With respect to ecological validity of task, because normal users of Tor are aware that they are engaging in a security task, we did not attempt to hide the nature of the task from participants, as might be done in, for example, a study on anti-phishing techniques.

There are a few other small limitations to this study. First, our sample was skewed heavily towards undergraduate males. While there is evidence that *privacy concerns* differ along gender lines [26], the authors are not aware of any evidence that *usability* issues in security differ along gender lines. For example, males and females fall equally for phishing attacks [15]. However, this skewed gender ratio remains a small limitation. Additionally, our sample sizes, (N = 25 for Study 1, N = 27 for Study 2), could have been larger. Both of these factors may affect the generalizability of our results.

C. Free-Response Coding Methodology

As mentioned previously, in addition to demographic questions, we asked participants to respond to the free-response question: *"Did you encounter any problems when installing or using the Tor Browser Bundle?"* In this section we will detail how we moved from a set of free-response answers to a set of mutually exclusive categories for usability and then discuss our methodology for assigning these responses to categories.

Category Generation: In Study 1 the 25 study participants reported a total of 41 stop-points in their answers to the free-response question. Some participants reported multiple issues.

Two coders independently coded the answers to these questions, assigning each complaint to one of of seven mutually exclusive categories. Categories were generated post-hoc after a holistic evaluation of the free-response questions. These categories were subsequently used to code the responses from Study 2. The categories of usability issues we discovered were as follows.

A.) Long launch time: The user noticed a lag between clicking the icon to start the TBB, and the TBB window opening.

B.) Browsing Delay: Browsing through the TBB had a noticeable lag.

C.) Download Clarity: User wasn't sure where on website to download the TBB.

D.) Window Discriminability: User wasn't sure which window was the TBB and which was a normal browser.

E.) Archive confusion: Problems unzipping the TBB package.

F.) Icon Salience: Problems finding the icon to start the file ("Start Tor Browser").

G.) Security Measure Confusion: Security measures taken by the TBB (such as redirecting from Google CAPTCHA to DuckDuckGo) confused users.

Interrater Reliability: To ensure coder agreement was not due to random chance, final intercoder agreement was calculated using Cohen's Kappa [10] for Study 1 and using Fleiss' Kappa for Study 2 [21]. Both of these formulas are methods of calculating observer agreement of categorical data that accounts for agreements due to chance. We used Cohen's Kappa for Study 1 because we only had two coders. For Study 2, additional research team members were able to assist with coding, necessitating a measure of reliability that allowed for multiple coders (i.e., Fleiss' Kappa).

Overall intercoder agreement between the two coders for Study 1 was Cohen's Kappa = .72, and overall intercoder agreement between the four coders for Study 2 was Fleiss' Kappa = 0.82. Kappas of .61 - .80 are considered 'substantial', and kappas between .80 and 1 are considered 'almost perfect' [30], showing that in both studies coding agreement was reliable. After the first pass of coding, there was 100% coder agreement.

In the next section we will detail the prevalence of each of category of usability issue we identified. We discuss the design implications of these findings, and present a set of design heuristics based on them in the Discussion section.

D. Study 1 Results

Results from Study 1 are shown in Table I. Sixteen users (out of 25) reported a total of 41 individual issues; some users reported multiple issues. As we can see from Table I, the majority of the issues users encountered in Study 1 centered around launch time, browser delay, and window discriminability.

Next we discuss the methodology and results of Study 2, then discuss the implications of both studies.

TABLE I
TYPE OF TOR PROBLEMS ENCOUNTERED, PRESENTED AS PERCENTAGE OF USERS EXPERIENCING AN ISSUE. SOME USERS EXPERIENCED MULTIPLE ISSUES, THUS CATEGORIES DO NOT NECESSARILY SUM TO 100%."*" INDICATES STATISTICAL SIGNIFICANCE.

Category	% Study 1	% Study 2	p-value	t-value	df
Popup Peeves*	N/A	44%	$< .001$	4.385	50
Long launch time*	50%	0%	$< .001$	-5.303	50
Browsing delay	24%	19%	.346	-.475	50
Window discriminability*	16%	0%	$< .001$	-2.224	50
Archive Confusion*	16%	4%	.002	-1.507	50
Icon salience	12%	7%	.271	-.552	50
Security Measure Confusion*	12%	0%	$< .001$	-1.882	50
Download Clarity*	12%	0%	$< .001$	-1.882	50

IV. Study 2: Verifying Our Recommendations

A. Implementing our Recommendations

Our second study aimed to verify that the issues we identified were solved by the design recommendations we made. After Study 1, the Tor Project made several changes to the TBB based on our preliminary results. For example, the Tor developers reduced the amount of time it took for the TBB to launch in order to reduce complaints of "long launch time." The Tor developers also created a new Tor Browser Bundle icon in order to solve problems with window discriminability.

One issue that was not addressed by these changes was Tor's speed. One of the main issues reported in study 1 were complaints about the TBB's latency. While there is considerable work on mitigating jitter and latency in Tor (eg. [43]), delay is an inherent price of Tor, and Tor has traditionally been slower than a typical Internet connection [18]. Building on Egelman et al.'s work with Mechanical Turk users [19], the authors hypothesized that when users are given *realistic* expectations about Tor's speed, they will not attribute this lack of speed to an error. In fact, this type of information may instead help users develop a more accurate mental model. Thus, instead of becoming frustrated, users may instead picture their packets traversing several nodes as they wait, thus gaining a sense of security from the delays sometimes introduced by Tor.

We hypothesized that users of Tor are likely willing to sacrifice some speed for better privacy. Simply taking steps to inform users that the TBB may take a while to open and that such delay is normal could substantially alter a user's perception of the TBB. As Molich et al. point out with their feedback principle [34], the typical user assumes that if a program fails to respond within a certain time frame, that either a process has run in the background, or an error has occurred. By providing an informative dialog instructing users

to wait for the browser window to open, the confusion Tor users experience can be avoided.

To test this hypothesis, the authors created a browser extension which informed users when delays occurred, along with a message that such delays were normal and a sign Tor was operating securely. If a delay greater than 10 seconds was detected, our custom extension displayed a warning stating that the delay was normal and was in fact a property of Tor's security. Users in our study were directed to a locally hosted mirror of the Tor project, which, aside from having it's windows zip file replaced with our custom version, matched the Tor Project's page in every manner.

Using the updated TBB and our custom extension, we conducted a second study to test whether the changes we recommended, as well as our browser extension, would indeed increase the usability of the Tor Browser Bundle.

B. Study 2 Methodology

We recruited 27 undergraduate students for Study 2 from the same course (but a subsequent semester) and followed the same study procedure using the same machines with the following exceptions: the instruction sheet provided the URL for a locally hosted Tor mirror that served our own version of the Tor Browser Bundle v2.3.25-2 with a custom extension that detected delays and informed the user that these delays were normal. Our hypothesis was that reframing lag as normal would make users more tolerant of it. Figure 1 shows a sample pop-up.

We then proceeded to code the responses from participants in Study 2 in the same manner as Study 1, with one difference: interrater reliability for Study 2 was calculated using Fleiss's Kappa [21], as described above. In Study 2, we also added the category 'Popup Peeves' to capture complaints directed at the pop-ups generated by our custom browser extension. Since our extension was not present in Study 1, Table I does not show any popup peeves for Study 1.

Study 2 Demographics: Similar demographic information was collected from all users on the second exit survey. Study 2's sample was skewed even more heavily male than Study 1 - 3% (1/27) were female. Participants ranged in age from 20–32, with a mean of 22.

Our second study's population had similar familiarity with Tor and information security to Study 1's participants. When asked "How familiar are you with Tor?", users responded with an average of 2.3 on a 7-point scale. Users were slightly more familiar with computer security. When asked "How familiar are you with computer security?", users reported an average of 3.9 on a 7-point scale.

Moving on, we will elucidate the results specific to Study 2, then in the discussion section we will discuss the implications of these results.

C. Study 2 Results

Twenty-seven users reported a total of 34 individual issues. It should be noted that the category "Pop Up Peeves" was added in study 2, and refers to complaints about the messages

Fig. 1. Example of pop up informing users in Study 2 that a delay has occurred

we added to the TBB in Study 2 informing users when delays occurred. This extension was not present in Study 1, hence no users reported issues with it in that study.

Referring back to Table I, we can note that overall, there was a numerical decrease across the board in usability issues during Study 2. Notably, the reductions in "Window Discriminability" and "Long Launch Time" were both statistically significant, implying that changes to the TBB increased usability.

While the reduction in Browsing Delay was not statistically significant, 60% (3/5) of the users expressing negative opinions about browsing delays also complained about usability issues with our warning pop-ups. Specifically, these three users all noted that the number of pop-ups (5 in one instance) seemed excessive. Users may have perceived the lag caused by clearing these pop ups as latency on the part of the Tor Browser.

Now that we have established what problems have been experienced by our participants in both Study 1 and Study 2, we will discuss the implications of these findings in our discussion section.

V. Discussion

Based on our results from Study 1, we hypothesized a set of design implications for the TBB, as well as a set of general design recommendations for all anonymity systems. Then based on those suggestions, several changes were made to the TBB. We will detail those changes, and then reflect on how these changed the TBB browsing experience.

As we can see from Table I, a higher percentage of people in Study 2 reported no problems than in Study 1, indicating an increase in usability. Additionally, we observed a decrease in the total number of problems reported (41 in Study 1 to 34 in Study 2), even though we had more participants in Study 2 than Study 1.

A. Issues and Solutions

After Study 1 was concluded, we had collected 41 issues from 25 users, as described in Table I — with some users listing multiple usability issues. As described in our methods section, we then coded these responses into mutually exclusive categories. We elucidate these issues below, along with our

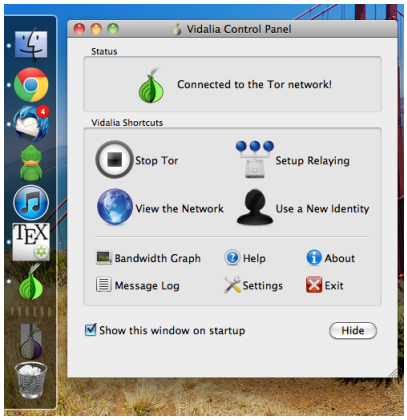

Fig. 2. Vidalia reporting a connection to the Tor network, even though the Tor Browser window has not yet opened.

hypothesized solutions. We then reflect on the efficacy of the solutions implemented for each issue in turn.

Issue 1: Long Launch Time: Many users noticed a long delay between clicking "Start Tor Browser Bundle" and the TBB opening. A typical scenario would be that the user would click on "Start Tor Browser Bundle". At this point, Vidalia (the graphical controller for Tor, an interface which confused users) appeared. Many users incorrectly assumed after 30 seconds or so that all their Internet traffic was anonymized and proceeded to open Firefox or Internet Explorer.

As mentioned earlier, the lag between Vidalia opening and the browser window opening had been significantly reduced for Study 2. As Table I shows, this led to statistically significant reduction in complaints about long launch time. We also theorize that this may have contributed to a statistically significant decrease in complaints about window discriminability, since users in Study 1 sometimes were confused by a long launch time, believing that all Internet traffic was anonymized, and then opened a non-anonymized browser window to attempt the task.

Furthermore, based in part on preliminary results from Study 1 shared with several Tor developers, the Tor Project began to develop a new design of the TBB which would eliminate the need for Vidalia, leaving just one window. This change went live in TBB version 3.0, released in alpha in June 2013[3]. While our study occurred prior to this change, we believe that eliminating Vidalia will reduce complaints of both "Long launch time" and "Window discriminability" even further.

Issue 2: Browsing Delay: Many users noted that browsing with Tor was slower than browsing over a typical Internet connection. As mentioned earlier, some latency is an inherent price of Tor. However, our extension led to a numerical decrease in complaints of "browsing delay."

Our extension could be improved. Users complained about the number of pop-ups they experienced using the TBB with our custom extension. Our delay threshold may have been too low, and if several tabs experienced delays in rapid succession, multiple pop-ups could occur for the same delay. Even so, we did see still a slight reduction in complaints related to browser delay when using our extension. Furthermore, as noted in our results, 60% of users experiencing browsing delay complaints also had issues with excessive pop-up warnings. In the future, capping the number of pop-ups to only once per browsing session may further eliminate these complaints.

[3]https://blog.torproject.org/blog/announcing-tor-browser-bundle-30alpha1

Fig. 3. The new Tor Browser Bundle icon.

Despite this small setback, our results were very promising. Note that there no increases in usability issues in any category (and statistically significant decreases in several categories.)

Issue 3: Window Discriminability: Several users in Study 1 (many of whom also experienced a long launch time) had trouble discriminating which window was the TBB and which was a normal browser window. It should be noted that the TBB is, at its heart, just a rebranded portable Firefox build. Study 1 tested TBB v2.2.35-7.1, which used the same icon as a normal Firefox installation. Thus, if a user had Firefox open prior to opening the TBB, it was extremely easy to use a non-Torified window when performing study tasks since the icons were identical. Confusing a non-Torified window for a Torified window could result in a user believing his or her web traffic is anonymous when it is not. The seriousness of this confusion is worth pointing out. If a whistleblower is attempting to communicate with a journalist and believes her Internet traffic is anonymous when it is not, the consequences could range from jail time in some countries to physical harm in others.

After Study 1 the Tor Project changed their program icon from the Firefox icon to its own custom icon. As noted in Table I, changing the TBB's icon caused a large and statistically significant drop in complaints about window discriminability.

Issue 4: Icon Salience: Some users were unclear how to start the TBB, or thought that the TBB would start automatically, not realizing that the "Start Tor Browser Bundle" would open an anonymized browser. This could lead to serious errors, such as when one participant assumed after unzipping the TBB that all subsequent traffic was anonymous and proceeded to attempt to complete the study tasks using an unanonymized system browser.

To solve this issue we proposed that the TBB could place an icon on the desktop/dock. The TBB could note at some point between downloading and installation that the user must click "Start Tor Browser Bundle" to begin. Alternatively, the TBB could launch automatically after installation.

Neither the Tor project nor our researchers made any changes to attempt to mitigate this issue, mainly because so few people (12% in Study 1) experienced this issue. The authors' efforts were centered on a cluster of three usability

issues: long launch time, window discriminability, and browsing delay, which together made up a majority (66%) of the usability issues reported.

Issue 5: Download Clarity: Some users were unsure which package to download and/or accidentally downloaded the wrong operating system's version of the TBB.

We proposed that the download page could provide larger logos for each operating system (as opposed to simply stating the OS in text), along with larger, bolded text describing which operating system a given package is for. As with Issue 4 our researchers were focused on the cluster of issues that caused a majority of usability issues and thus did not make any changes to the Tor project's download page when establishing our mirror.

Issue 6: Security Measure Confusion: Some security measures that the TBB takes, such as redirecting from Google searches to DuckDuckGo, and disabling certain types of active content confused users in Study 1 who did not understand why a given action had been redirected or a pop-up box had been generated.

As a result of our results in Study 1 suggested that prior to performing any redirects, the TBB could provide a jargon-free explanation of *why* a security measure is being taken. For example, before redirecting to DuckDuckGo, a pop up could appear and state: *"Google keeps a record of your history. Using DuckDuckGo will allow you to search anonymously."*

Since decision science and the field of risk communication are vast, complex fields, and because security measure confusion was a small issue (present in 12% of complaints in Study 1), we chose to save the rewording of security dialogs for future work. No users experienced this issue in Study 2.

Issue 7: Archive Confusion: Some users expected a guided 'wizard' installer and did not realize they had to click on "Start Tor Browser Bundle" once unzipping had occurred, leading to confusion.

This issue is not necessarily a problem with Tor, but as we discuss later in our design heuristics, installation of the TBB is a prerequisite for *using* the TBB. While the TBB developers cannot control the usability of the host operating system, we suggested that a prominent note could be made on the download page that users will need to unzip the TBB prior to using it.

Since this was a minor issue (effecting 12% of Study 1 participants) we did not attempt to change the Tor download page on our mirror, instead directing our efforts to more serious usability issues such as "Browsing Delay'".

No users experienced "Archive Confusion" in Study 2 (despite no changes in the UI occurring.) Furthermore, after Study 2 occurred, but prior to submission of this paper, the Tor Project released TBB 3.0, which now provides a wizard-style graphic installer (along with several other improvements.) This design change occurred after preliminary results from Study 1 were shared with members of the Tor Project. While we did not examine this change in our study, based on a brief cognitive walkthrough of the installer, the authors feel that said installer will help reduce user confusion.

B. Design Heuristics For Anonymity Systems

In the previous subsection we described seven stop-points and potential solutions based on the usability issues discovered in Study 1, and the verification that our suggested fixes reduced these issues in Study 2. Each design recommendation was discussed in the context of the coding category that documented it. We found that vast majority of the issues were created by long launch times, browsing delay, and window discriminability.

Based on our experience hypothesizing and testing the above issues and solutions in Study 1 and the decrease in usability issues seen in Study 2, we can now arrive at a set of general design recommendations that generalize to other anonymity systems.

Heuristic 1: Installation precedes operation: Even the most well-designed user interface is useless if the user never reaches it. The authors of anonymity software should strive to assist users who are installing the software. Download pages should try and make educated guesses as to what operating system a user is running, and provide users with simple heuristics for determining their operating system. For example, next to a link to download the Windows version of an anonymity software package, the page could state "If your computer is not a Mac, you probably want this version."

Heuristic 2: Ensure users are aware of trade-offs: Today's users have come of age in a time of widespread broadband adoption. Delays longer than a few seconds may cause users to question whether their connection is faulty. While ideally anonymity software should strive to deliver content with as little latency as possible, users of anonymity software are usually willing to trade speed for privacy. However, the software must provide feedback to the user to let them know that a given operation has not failed. Just as a user is willing to accept a slower connection via a crowded Internet cafe wifi network, a user is willing to accept a delay in exchange for anonymous communication. Informing users that delays are normal may increase tolerance of delays, but these messages must be limited, or they may become a design flaw in their own right.

Heuristic 3: Say why, not how: Sometimes an anonymity system must take a security measure, such as redirecting away from a site which may leak identity information, or disabling browser features such as cookies or Javascript. Users desire to be told *why* a given security precaution is being taken. These explanations should avoid technical jargon and use real-world metaphors whenever possible. Users who wish to understand at a deeper level should be given the option to drill down to a more detailed technical explanation.

Heuristic Examples: We can generalize our heuristics to any system that allows users to 'hide in the crowd' and which aims to maximize adoption of a given anonymity system by minimizing user irritants.

For example, the anonymous operating system TAILS [2] requires users to create a bootable live USB or live DVD, which may hinder adoption by less savvy individuals — recall that participants in Study 1 occasionally expected a "guided

wizard" installation process. Such users may also find the idea of creating bootable DVDs/USBs as confusing and/or unhelpful.

Along similar lines, traditional single hop proxy software is equally confusing. Users must seek out list of proxy addresses, and navigate confusing browser dialogs in order to tunnel their traffic through said proxy. In the event that the proxy goes down (a common occurrence), users must navigate a series of extremely technical errors and dialog boxes to fix the issue.

VI. Conclusions

Based on our studies, we have discovered a number of usability issues in the TBB. As Back et al. succinctly state, *''In anonymity systems usability, efficiency, reliability and cost become security objectives because they affect the size of the user base which in turn affects the degree of anonymity it is possible to achieve."* [6].

We noted that long launch time, browsing delay, and window discriminability were the issues most often cited by participants. Based on these issues we presented a set of three design heuristics to help minimize usability issues in anonymity systems.

We note that "*installation precedes operation*" — if installation of an anonymity system frustrates the user, they may never reach the UI, no matter how well designed it is. We also suggested that makers of anonymity systems ensure users are aware of the speed trade-offs in anonymity systems and set appropriate expectations. With our "*Say why, not how.*" heuristic, we encouraged developers to explain why security measures that impact the user experience are taken and that these explanations avoid technical jargon.

Finally, based on our results in Study 2, we validated our design heuristics, showing that by applying these three design heuristics, developers can help make the TBB (or any similar anonymity system) more usable and thus more secure.

Taken together, these recommendations may improve the usability of Tor and other anonymity systems, which may improve the security and privacy of online communication for a variety of users including journalists, whistleblowers, and any others who wish to communicate anonymously. Even small changes in the usability of such a system have the potential to have a disproportionately large impact in terms of potential consequences for a user. Where the consequences of poor usability of most consumer software may be frustration, additional work, and or stress, the potential consequences of lack of usability of anonymity systems may lead to censorship, surveillance, and in very extreme cases, physical harm.

VII. Future Work

While this work examines a number of possible changes to the TBB (and other anonymity systems), it leaves several open research questions. For example, the researchers did not attempt to redesign security warnings that appear during the use of the TBB.

Second, many design decisions in our delay-detecting extension were not empirical. Variables such as the threshold to display a notice that lag is occurring, and how often per browsing session to display such warnings could have their optimal values determined experimentally.

Finally, while Study 1 identified several complaints about security warnings (as reflected in the category "*Security Warning Confusion*" in Table 1), further work could be dedicated to examining how to redesign the various warning dialogs present in the Tor Browser Bundle to nudge more clearly non-expert users towards safe decisions.

VIII. Acknowledgments

The research is supported by DHS Contract N6600112C0137, NSF Grant 1250367, and DARPA FA8750-13-2-0023, with additional funding from Google and Microsoft. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF, DHS, DARPA, Indiana University, Clemson University, ISI, or any other funding organization.

The authors would also like to thank Roger Dingledine, Mike Perry, Tom Lowenthal, and all the other Tor Project members who provided valuable feedback on early versions of this work, and made changes to Tor based on it.

References

[1] How to remain secure against the nsa - schneier on security. https://www.schneier.com/blog/archives/2013/09/how_to_remain_s.html.

[2] Tails: The amnesiac incognito live system. https://tails.boum.org/.

[3] Tor project metrics portal: Users. https://metrics.torproject.org/users.html.

[4] A. Adams and M. Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, 1999.

[5] R. Anderson. Why information security is hard-an economic perspective. In *Computer Security Applications Conference, (ACSAC)*. IEEE, 2001.

[6] A. Back, U. Möller, and A. Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Information Hiding*, pages 245–257. Springer, 2001.

[7] J. Boyan. The anonymizer. *Computer-Mediated Communication (CMC) Magazine*, 1997.

[8] S. Brostoff, P. Inglesant, and M. Sasse. Evaluating the usability and security of a graphical one-time pin system. In *Proceedings of the 24th BCS Interaction Specialist Group Conference*. British Computer Society, 2010.

[9] L. J. Camp and C. Wolfram. Pricing security. In *Economics of Information Security*, pages 17–34. Springer, 2004.

[10] J. Carletta. Squibs and discussions assessing agreement on classification tasks: The kappa statistic. *Computational linguistics*, 22(2):249–254, 1996.

[11] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[12] J. Clark, P. Van Oorschot, and C. Adams. Usability of anonymous web browsing: An examination of tor interfaces and deployability. In *Proceedings of the 3rd symposium on Usable privacy and security*, pages 41–51. ACM, 2007.

[13] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing Privacy Enhancing Technologies*, pages 46–66. Springer, 2001.

[14] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pages 2–15. IEEE, 2003.

[15] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006.

[16] R. Dingledine and N. Mathewson. Anonymity loves company: Usability and the network effect. In *Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006), Cambridge, UK, June*, 2006.

[17] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association.

[18] R. Dingledine and S. Murdoch. Performance improvements on tor or, why tor is slow and what were going to do about it. 2009.

[19] S. Egelman, A. Acquisti, D. Molnar, C. Herley, N. Christin, and S. Krishnamurthi. Please continue to hold an empirical study on user tolerance of security delays. 2010.

[20] H. Elmore. Designing translucent security: Insights from a usability evaluation of pgp desktop. Master's thesis, Indiana University, 2009.

[21] J. L. Fleiss and J. Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33(3):613–619, 1973.

[22] M. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 193–206. ACM, 2002.

[23] V. Garg and J. Camp. Heuristics and biases: Implications for security design. *Technology and Society Magazine, IEEE*, 32(1):73–79, 2013.

[24] D. Goldschlag, M. Reed, and P. Syverson. Onion routing. *Communications of the ACM*, 42(2):39–41, 1999.

[25] R. Hertwig, G. Barron, E. Weber, I. Erev, et al. Decisions from experience and the effect of rare events in risky choice. *Psychological science*, 15(8):534, 2004.

[26] M. G. Hoy and G. Milne. Gender differences in privacy-related measures for young adult facebook users. *Journal of Interactive Advertising*, 10(2):28–45, 2010.

[27] P. Inglesant and M. Sasse. The true cost of unusable password policies: password use in the wild. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 383–392. ACM, 2010.

[28] C.-M. Karat. Iterative usability testing of a security application. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 33, pages 273–277. SAGE Publications, 1989.

[29] S. Köpsell. Low latency anonymous communication–how long are users willing to wait? In *Emerging Trends in Information and Communication Security*, pages 221–237. Springer, 2006.

[30] J. Landis and G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977.

[31] S. Lederer, I. Hong, K. Dey, and A. Landay. Personal privacy through understanding and action: five pitfalls for designers. *Personal and Ubiquitous Computing*, 8(6):440–454, 2004.

[32] K. Loesing, S. J. Murdoch, and R. Dingledine. A case study on measuring statistical data in the Tor anonymity network. In *Proceedings of the Workshop on Ethics in Computer Security Research (WECSR 2010)*, LNCS. Springer, January 2010.

[33] P. Mittal, A. Khurshid, J. Juen, M. Caesar, and N. Borisov. Stealthy traffic analysis of low-latency anonymous communication using throughput fingerprinting. In *Proceedings of the 18th ACM conference on Computer and Communications Security (CCS 2011)*, October 2011.

[34] R. Molich and J. Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33(3):338–348, 1990.

[35] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of tor. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, SP '05, pages 183–195, Washington, DC, USA, 2005. IEEE Computer Society.

[36] G. Norcie, K. Caine, and L. J. Camp. Eliminating Stop-Points in the Installation and Use of Anonymity Systems: A Usability Evaluation of the Tor Browser Bundle. In *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETS)*, 2012.

[37] M. Reed, P. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *Selected Areas in Communications, IEEE Journal on*, 16(4):482–494, 1998.

[38] R. Reeder and R. Maxion. User interface dependability through goal-error prevention. In *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, pages 60–69. Ieee, 2005.

[39] S. Romanosky and C. Kuo. Foxtor: Anonymous web browsing. Tor GUI Competition, 2006.

[40] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[41] A. Tversky, D. Kahneman, et al. The framing of decisions and the psychology of choice. *Science*, 211(4481):453–458, 1981.

[42] H. Varian. System reliability and free riding. *Economics of Information Security*, pages 1–15, 2004.

[43] C. Viecco. *Improving Anonymity Networks via End-to-End Design*. PhD thesis, Indiana University, 2011.

[44] A. Whitten and J. Tygar. Why johnny cant encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th USENIX Security Symposium*, volume 99, 1999.

[45] M. E. Zurko and R. T. Simon. User-centered security. In *Proceedings of the 1996 workshop on New security paradigms*, pages 27–33. ACM, 1996.