Eliminating Stop-Points in the Installation and Use of Anonymity Systems: a Usability Evaluation of the Tor Browser Bundle

Greg Norcie, Kelly Caine, L Jean Camp

Indiana University

Abstract

Broad-based participation in anonymizing technology can be hindered by a lack of usability. Since systems such as Tor give $\frac{1}{n}$ anonymity, the existence of stop-points is of particular concern because wider adoption of Tor equals better anonymity for all users of the system. Stop points are places in an interface at which an individual is at risk from being presented with a situation which may prevent them from moving forward with installation or use. Sixty four percent of users in our study encountered at least one stop-point while using the Tor Browser Bundle. While each stop-point may be minor in isolation, the cumulative effect is great and has the potential to harm the overall anonymity of the system. In this paper we enumerate these stop points and detail specific design changes that may address them. We also provide more generic but generalizable recommendations for the design of anonymity systems. Taken together, these suggestions may enhance usability and thus anonymity of individual users as well as system level anonymity.

1 Introduction

Tor is an anonymity network that uses onion routing [11] to prevent third parties from observing a user's web connection. The term onion routing refers to Tor's layered encryption (analogous to the layers in an onion), which prevents third parties from observing the content or linking the source and destination in the internet traffic of a participant in the Tor network.

As Dingledine and Mathewson[10] point out, there is a strong network effect present in anonymity networks like Tor. In any system where anonymity is $\frac{1}{n}$, additional users increase the anonymity of the system. Thus, barriers that reduce the number of people who adopt an anonymity system like Tor, reduce the anonymity all users of the system receive. Therefore, any increase in the usability of Tor that reduces barriers to adoption and increases the anonymity Tor provides. While many papers have examined the technical aspects of Tor's security [21][19][18], relatively few papers have examined Tor's usability (see [25] and [6] for notable exceptions). Our work is further distinguished by the dual focus on both flaws in the Tor interface, as well as their resolutions.

Tor can be run in a variety of configurations. Previous work [6] offered a cognitive walkthrough (but no subsequent lab study) of Tor, and suggested that Torpark, a Firefox variant with Tor incorporated into the browser was the most usable. Tor later released an official "Tor Browser Bundle" (TBB) which similarly contains a standalone Firefox variant with Tor built in, a design change backed up by the data collected by Clark et al.

Our work complements and extends existing work on improving the usability of Tor by identifying"stop-points" in the installation and use of the Tor Browser Bundle. 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. [14]

However, a stop point does *not* necessarily prevent a user from going forward — any point at which the user is going forward while confused about the functionality or state of the system is also a stop point.

In this paper, we evaluate stop points during both the installation and the use of the Tor Browser Bundle, in a lab study with 25 participants. Based on this study, we utilize our results to specify immediate changes that can be made to the Tor Browser Bundle Interface. We also present a set of design recommendations which can be applied not only to the Tor Browser Bundle, but are also generalizable to other anonymity systems.

In the next section we describe related work. We place our study in the larger context of three main bodies of work: usable security, privacy enhancing technologies, and design heuristics. Given the breadth of these domains, these descriptions are necessarily brief. Following this overview of related work we describe our method in Section 3. In Section 4 we enumerate our findings. We then explicate these findings and their implications in Section 5, including both a general discussion that places the results in the context of related work and suggests design recommendations both specifically for Tor as well as for anonymity systems generally. Finally, we outline limitations of our study and then conclude in Section 7.

2 Related Work

There are three bodies of literature that inform this work. The three domains on which we draw for this work are laboratory evaluation of usability (particularly for security and privacy); privacy enhancing technologies; and design heuristics.

2.1 Usable Security in the Laboratory

In Whitten and Tygar's "Why Johnny Can't Encrypt" [25], one of the first usable security papers published, Whitten and Tygar performed a cognitive walkthrough and laboratory study examining the usability of PGP 5.0. Whitten and Tygar state that security software is usable if users are 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. Based on their evaluation of PGP 5.0, Whitten and Tygar note several problematic properties of computer security.

The major issues Whitten and Tygar noted are that users are unmotivated to protect their information, that it is difficult to produce abstractions for many security functions, that there is a lack of feedback when performing security tasks. Whiten and Tygar also describe what they call the "barn door property" - that data, once lost, cannot be reclaimed (evoking the futility of locking a barn once the animals have escaped). Whitten and Tygar also draw on economics of security literature (eg [24]) to apply the "weakest link property'. The weakest link property states that a security system is only as strong as it's weakest link. Often the weakness in a security system is the lack of usability in the interface, or the failure of an interface to match the mental model of the users. These findings have been reified in later work including [8] and [4], respectively.

Whitten and Tygar concluded that interfaces must be designed to guide and inform security decisions. Our work draws on this but targets on specific points of potential failure, and lessso on macro observations as described by Whitten et al.

Maxion and Reeder implemented a similar laboratory examination of usability of access control. 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 [23].

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 and not designed for humans

— security policies are too inflexible to match their capabilities [16]. A follow-up study illustrated that graphical passwords had similar difficulties [3].

Engelman et al. [13] examined whether users would tolerate security delays in a Mechanical Turk task asking them to test a new web based PDF viewer, reading a document and reporting the frequency that a word appeared in a particular document, and then presenting them with a delay. Engleman et al. found that 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.

2.2 Privacy Enhancing Technologies

The idea of passing an encrypted message between series of intermediate nodes was first discussed by Chaum [5]. 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 but increasing latency) as opposed to there being a stable path for any session.

There have been several high anonymity, high latency systems such as MixMinion [9]. Zero Knowledge Systems' "Freedom" was released in beta in early 2000. Tarzan provided transport-layer anonymization using a decentralized, distributed Chaumian mix. [15] Freenet offered anonymous and resilient publishing using distributed content and shared keys; however, the initiator of a request could be identified. [7] Free Haven offered anonymous storage as well as protecting against traffic analysis. [7]

Yet none of these platforms ever gained popularity. Indeed, many of them never went beyond laboratory demonstration projects. Onion routing was first presented in 1998. [22] The second generation onion router work was published in 2004, when Dingledine was with Free Haven.[11] Before 2000 the majority of anonymizing systems that were used in practice were single hop proxies, e.g. [2]. 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 a more usable version of Tor combines the proxy with Tor along with a browser in one package called the Tor Browser Bundle (TBB). This simplified installation and interface has the potential to expand Tor to a broader user base.

2.3 Design Heuristics

Usability design heuristics predate the field of usable security by many years. Molich and Nielsen [20] wrote a widely cited set of design heuristics for human-computer interaction. Whitten and Tygar [25] expanded on this work, pointing out that secure systems have additional usability requirements. Users of privacy enhancing technologies need to be aware of the security tasks they need to perform. The user must then be able to perform these task(s) without making any dangerous errors, and then be comfortable enough with the interface to continue using it.

Clark, Oorschot, and Adams performed a cognitive walkthrough of several early Tor interfaces, but did not follow up with a lab study.The authors concluded that Torpark (a self contained browser bundle similar to the TBB our lab study evaluates) was the most usable option for novice Tor users. [6]

3 Methodology

3.1 Recruitment and Procedures

We recruited 25 undergraduates from a large midwestern university. Students were given lab credit for participating in the experiment. Students who were not comfortable participating were given the option to instead write a one page essay on Tor's basic functionality. Our study was approved by the Indiana University Institutional Review Board.

All participants were seated at VMware image of 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. Users also had their on-screen actions using screen capture software. The instruction sheet that was handed out provided users with the URL for the Tor Project (`http://torproject.org`). Users were then directed by the instruction sheet to download the Tor Browser Bundle (TBB), run the TBB, and use the TBB to download a new desktop background for their lab machine.

Before being given their instructions, users were informed that the experiment was a usability experiment, and thus the normal rules for a lab did not apply. Normally, students in a lab are expected to complete their tasks with minimal aid from the instructor, except for clarification of instructions, working through any complexities on their own. Before the experiment, users were briefed that their instructions were purposefully vague, and that if they were unclear how to proceed at any time, they should raise their hand so that the experimenter 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. A post-task survey also asked users if they had encountered any issues. Users who raised their hands were instructed to note their issue on their post-task survey, and then told how to proceed past the stop-point. Finally, we recorded each user's screen, and were able to go back and examine their recordings if a user's textual response contained ambiguity.

3.2 Ecological Validity

Normally, in usable security experiments, experimenters strive to find non-expert users. However, it is the author's belief that typical users of Tor are moderately security savvy. Furthermore, since our users were security savvy, it actually strengthens the authors' argument that the interface is less usable than is desirable if even security experts make errors when using it.

Furthermore, since normal users of Tor are aware that they are engaging in security task, we did not attempt to hide the nature of the task from participants as one might do in say, a phishing study.

3.3 Sample Information

Demographic information was collected from all users during the exit survey. The sample was 88% male (22/25). Participant ages ranged from 20 to 37, with a median of 22.7 and a mode of 21. Participants were asked if they had heard of Tor. They 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.

3.4 Coding

Each post-task survey asked, in addition to demographic questions, two free response questions:

1. "Did you encounter any problems when installing the Tor Browser Bundle?"
2. "Did you encounter any problems when using the Tor Browser Bundle?"

The 25 study participants reported a total of 41 stop-points in these two free response questions. Two coders independently coded the results 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:

A.) Long launch time: The user noticed a lag between clicking the icon to start the Tor Browser Bundle, 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 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.

Final intercoder agreement was calculated using Cohens Kappa, a method of calculating observer agreement of categorical data that accounts for agreements due to chance. Overall intercoder agreement between the two coders was Cohens Kappa = .72. Kappas of .61 - .80 are considered substantial. [17] After the first pass of coding, there was 100% coder agreement.

Now we will move onto the heart of our paper. Details on which of the issues that these categories describe were most prevalent are discussed in section 4 We then discuss the design implications of these findings, and present a set of design heuristics based on them in section 5.

4 Results

As reported in 3 our users were mostly moderately security savvy college students. Eighty percent of our users (20/25) were college aged (18-22) and all were currently enrolled an at least one computer science or informatics class. The study participants reported being familiar with computer security on a 7 point likert scale with 7 point likert scale with 1 being "Not at all familiar" and 7 being "Very familiar", the average participant scored a 4.5 when asked "How familiar are you with computer security? However,while 84% (21/25) of participants had heard of Tor, the average familiarity was only 2.13 on a 7 point likert scale. Table 1 summarizes the demographics of our participants.

Gender	
Male	22
Female	3
Age	
18–22	20
22+	5
Class Year	
Freshman (1)	0
Sophomore (2)	1
Junior (3)	9
Senior (4)	15
Other	0
Heard of tor?	
Yes	21
No	4
Familiar w/ Tor?	
1 (Not at all familiar)	11
2	5
3	7
4	0
5	2
6	0
7 (Extremely Familiar)	0
Familiar w/ computer security?	
1 (Not at all familiar)	0
2	0
3	1
4	9
5	10
6	3
7(Extremely Familiar)	0

Table 1: Results summary

We found that 36% of users (9/25) reported having no problems installing or using the Tor Browser Bundle. The remaining 16 users reported a total of 41 individual issues. As we can see from Table 2, the majority of the issues users encountered centered around launch time, browser delay, and window discriminability.

Category	Description	N	%
Long launch time	The user noticed a lag between clicking the icon to start the Tor Browser Bundle, and the TBB window opening.	13	40.6%
Browsing delay	Browsing through the TBB had a noticable lag.	6	18.8%
Window discriminability	User wasn't sure which window was TBB and which was a normal browser.	4	12.5%
Archive Confusion	Problems unzipping the TBB package.	4	12.5%
Icon salience	Problems finding the icon to start the file ("Start Tor Browser")	3	9.4%
Security Measure Confusion	Security meausures taken by the TBB (such as redirecting from Google CAPTCHA, to DuckDuckGo) confused users.	3	9.4%
Download Clarity	User wasn't sure where on website to download the TBB	3	9.4%

Table 2: Type of problems Tor problems encountered

Now that we have established what problems have been experienced by our participants, we will discuss the implications of these findings in section 5, and present a set of design principles based on them.

5 Discussion

Based on our results, we will present two sets of design implications. First, we will present a set of design recommendations specific to the Tor Browser Bundle. Then, we will generalize our results to provide a generalized set of design heuristics for creators of anonymity systems.

5.1 Tor Design Issues and Solutions

Issue: Long Launch Time (13/41)

Many users noticed a long delay between clicking "Start Tor Browser Bundle" and the Tor Browser Bundle 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, whose interface 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.

Solution(s):

Users of Tor are likely willing to trade speed for 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

Tor Browser Bundle. As Molich and Nielsen point out, systems need to provide feedback to users. [20] 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.

Issue: Browsing Delay (6/41)

Many users noted that browsing with Tor was slower than browsing over a typical internet connection.

Solution(s):

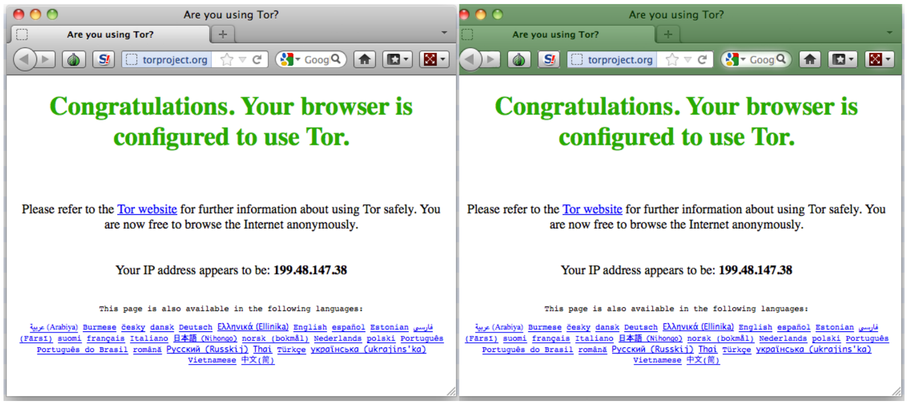
Inform users (on the download page, during installation, and/or on the initial home page) that Tor may be slower than traditional connections, due to its traffic flowing through a series of intermediary nodes. Tor has traditionally been slower than a typical internet connection[12]. Building on Egelman's work with Mechanical Turk users,[13] the authors theorize that 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, insted 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.

Issue: Window Discriminability (4/41)

Some users (many of whom also experienced a long launch time) had trouble discriminating which window was the Tor Browser Bundle and which was a normal browser window. This caused users to (erroneously) use a non-protected Firefox session to perform study tasks.

Solution(s):

The Tor Browser Bundle could consider using a more distinct program icon. The TBB could also change the browser's chrome to visually separate it from other browser windows with custom colors and icons matching those found on the TBB's default homepage (`https://check.torproject.org`), as shown in figure 1.



Congratulations. Your browser is configured to use Tor.

Please refer to the Tor website for further information about using Tor safely. You are now free to browse the Internet anonymously.

Your IP address appears to be: **199.48.147.38**

This page is also available in the following languages:

عربية (Arabiya) Burmese česky dansk Deutsch Ελληνικά (Ellinika) English español Estonian فارسی (Fārsī) suomi français Italiano 日本語 (Nihongo) norsk (bokmål) Nederlands polski Português Português do Brasil română Русский (Russkij) Thai Türkçe українська (ukrajins'ka) Vietnamese 中文(简)

Congratulations. Your browser is configured to use Tor.

Please refer to the Tor website for further information about using Tor safely. You are now free to browse the Internet anonymously.

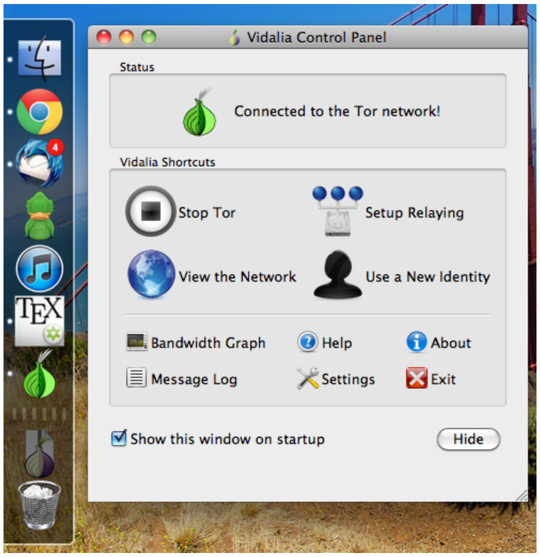
Your IP address appears to be: **199.48.147.38**

This page is also available in the following languages:

عربية (Arabiya) Burmese česky dansk Deutsch Ελληνικά (Ellinika) English español Estonian فارسی (Fārsī) suomi français Italiano 日本語 (Nihongo) norsk (bokmål) Nederlands polski Português Português do Brasil română Русский (Russkij) Thai Türkçe українська (ukrajins'ka) Vietnamese 中文(简)

Figure 1: TBB interface before (left) and after (right) enhanced discriminability changes

Finally, the Tor Browser Bundle could hiding the Vidalia control panel, since having two applications with two separate browser windows and two separate icons often confuses users. Finally, if user accidentally closes the Tor Browser, Vidalia continues to inform the user that they are connected to the Tor network, as pictured in figure 2. This caused several users who accidentally closed the Tor Browser to erronously believe that *all* browser traffic was being anonymized.



Connected to the Tor network!

☑ Show this window on startup

Figure 2: Vidalia reporting a connection to the Tor network, even though the Tor Browser window has been closed.

Issue: Icon Salience (4/41)

Some users were unclear how to start the Tor Browser Bundle, or thought that the Tor Browser Bundle would start automatically. The participant would not realize that the “Start Tor Browser Bundle” This could lead to serious errors, such as when one participant assumed after unzipping the TBB that all traffic was now anonymous, and proceeded to attempt to complete the study tasks using an unprotected system browser.

Solution(s):

the Tor Browser Bundle could place an icon on the desktop / dock. The Tor browser bundle could note at some point between downloading and installation that the user must click ”Start Tor Browser Bundle” to begin. Alternatively, the Tor Browser Bundle could launch automatically after installation.

Issue: Download Clarity (4/41)

Some users were unsure which package to download and/or accidentally downloaded the wrong operating system’s version of the Tor Browser Bundle.

Solution(s):

The download page could provide larger logos for each operating system, along with larger, bolded text describing which operating system a given package is for.

Issue: Security Measure Confusion (3/41)

Some security measures that the TBB takes, such as redirecting to Google searches to DuckDuckGo, and disabling certain types of active content confused non-technical users who did not understand why a given action had been redirected or a pop up box had been generated.

Solution(s):

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."*

Issue: Archive Confusion (3/41)

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.

Solution(s):

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 Tor Browser Bundle. While the TBB developers cannot control the usability of the host operating system, a prominent note could be made on the download page that users will need to unzip the Tor Browser Bundle prior to using it.

5.2 Summary of Design Implications for Tor

In the previous section we described seven stop points and potential solutions based on our results in section 4. 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.

5.3 Potential Design Heuristics For Anonymity Systems

Based on the above issues, we can arrive at a set of general design recommendations that may generalize to other anonymity systems. Any system that allows users to "hide in the crowd" can benefit from these heuristics, which aim to maximize adoption of a given anonymity system. Earlier we described a set of specific solutions — now we present more general heuristics applicable to a wide body of 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 like 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.

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.

6 Limitations

There are several limitations to this study. First, our sample was skewed heavily towards undergraduate males. Our sample size (N = 25), could have been larger. Both of these factors may affect the generalizability of our results. Also, while this paper provides several design heuristics, these principles have not been tested in the laboratory, and future work is needed to verify that indeed enhance the usability of any anonymity systems. For example, a future study should create a new version of the Tor Browser Bundle incorporating this paper's design recommendations, and test whether participants actually find the interface more usable when these changes have been implemented.

7 Conclusions

Based on our survey, we have discovered a number of usability issues in the Tor Browser Bundle. 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."[1].

We noted that the 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 said that "installation precedes operation" precedes operation, since if the 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. Finally, with our "Say why, not how." heuristic, we encourage developers to explain why security measures which impact the user experience are taken, and that these explanations avoid technical jargon.

By applying these design heuristics, developers can help make the Tor Browser Bundle (or any similar anonymity system) more usable, and thus, more secure.

8 Future Work

While this work presented a set of heuristics for designing anonymity systems, as well as a few diagrams giving examples of possible changes to the Tor Browser Bundle, we neglected to present a unified design incorporating our suggestions.

Furthermore, while have presented a number of suggestions, but we have not experimentally verified them. Thus future work could continue along two lines. First, we could develop a new Tor Browser Bundle interface, taking into account the findings in this paper. After developing this new interface, we could verify via lab study that our interface changes were effective in increasing the usability of the Tor Browser Bundle.

References

[1] 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.

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

[3] 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*, pages 88–97. British Computer Society, 2010.

[4] L. J. Camp. Bringing mental models to privacy and security. *IEEE Technology And Society Magazine*, 28(3):37–46, 2009.

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

[6] 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.

[7] 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.

[8] L. Cranor and S. Garfinkel. *Security and Usability*. O'Reilly Media, Inc., 2005.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] H. Elmore. Designing translucent security: Insights from a usability evaluation of pgp desktop. 2009.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

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

[21] 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.

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

[23] 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.

[24] H. Varian. *Microeconomic analysis*, volume 2. Norton New York, 1992.

[25] 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.