Will this onion make you cry?
A Usability Study of Tor-enabled Mobile Apps

[Poster Abstract]

Hala Assal
Carleton University
HalaAssal@scs.carleton.ca

Sonia Chiasson
Carleton University
chiasson@scs.carleton.ca

1. INTRODUCTION

In 2013, more than half a million users relied on Tor, the second generation Onion Routing system, daily to protect their privacy [1]. Users hide among other users in the Tor network to achieve anonymity, so the degree of anonymity grows as more users successfully participate in the network [3]. A user who makes mistakes while installing or using Tor software jeopardizes both her own privacy and that of other users in the network [2]. Accordingly, in order to attract more users, ensure their successful participation in the network, and improve privacy, the usability of Tor tools is considered "*a security requirement*" [3].

A poor user interface may lead to dangerous errors, and can leave users unknowingly unprotected with a false sense of security that can be especially harmful [4]. Whitten *et al.* [4] performed both a cognitive walkthrough (CW) and a user study on PGP 5.0 and used their results to introduce general principles for designing usable security UIs that are now widely accepted. Clark *et al.* [2] used CW for studying the usability of different Tor tools on desktop computers. The authors defined a set of representative tasks, then evaluated them against usability guidelines compiled specifically for Tor-enabled tools. To the best of our knowledge, we present the first usability study of Tor mobile apps.

2. TOOLS UNDER STUDY

Orbot is a Tor client proxy app for Android enabling other apps to use the Internet anonymously. If a user has a fully-privileged account (root), Orbot can intercept and route all outbound traffic to the Tor network. Otherwise, users could manually configure individual apps with proxy features to route their traffic through Orbot. **Orweb** is a basic browser built specifically for anonymous browsing through Orbot. The only functionality it provides are browsing and setting a home page. The default homepage displays the status of the connection to the Tor network. Android's Firefox does not allow users to route outbound traffic through a proxy server (e.g., Orbot). **ProxyMob** overcomes this limitation by allowing users to configure proxy settings for a Firefox session on Android devices. To route Firefox traffic, *both* Orbot and ProxyMob must be running.

3. EVALUATION METHODOLOGY

We study the usability of Orbot, Orweb, and ProxyMob using a CW as it focuses on users' goals and knowledge while performing specific tasks. We provide evaluators with personas, core tasks to perform, and task context through scenarios. We also provide the apps' description from the Play Store. After completing the CW, and guided by its results, we evaluate the apps' usability against a set of guidelines, similar to a Heuristic Evaluation.

We perform one CW session per evaluator, where the evaluator acts one persona and carries out three tasks on each app. Sessions were video recorded and the evaluators were encouraged to think aloud and comment on their persona's experience. The three evaluators have background in usable security. Our study was performed on an *unrooted* Nexus 7 tablet running Android v4.3. We evaluate Orbot v12.0.5, Orweb v5.1, and ProxyMob v0.0.10.

Tasks: Each persona performed three main tasks per app: (i) Install (and configure) the required components. (ii) Run the apps and configure web-traffic first to be anonymized to any location different from the real location and secondly to a specific location. (iii) Disable traffic anonymizing and return to a direct connection.

Guidelines: We also evaluated the apps against nine usable security guidelines. The first eight by Clark *et al.* [2]. The ninth was inspired by Yee's *Path of Least Resistance* design principle [5]. The guidelines state that *Users should...*

G1 *be aware of the steps needed to complete a task.*
G2 *be able to determine how to perform these steps.*
G3 *know when they have successfully completed a task.*
G4 *be able to recognize, diagnose, and recover from non-critical errors.*
G5 *not make unrecoverable dangerous errors.*
G6 *be comfortable with the terminology used.*
G7 *be sufficiently comfortable with the UI to keep using it.*
G8 *be aware of the application's status at all times.*
G9 *be guided to take secure actions.*

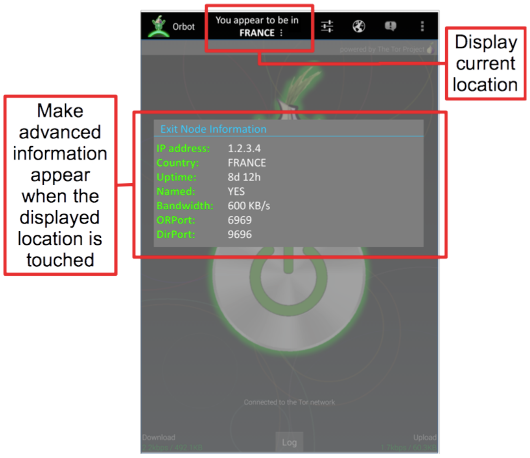
4. RESULTS

We highlight critical usability issues for each app and propose improvements to the design addressing each issue.

4.1 Orbot

Overall, Orbot uses very technical language (e.g., obfuscated bridges, exit nodes) that is not understandable to most users and lacks appropriate detail in many instances. For example, the forged location is not shown, making it difficult to verify proper anonimization. If the user does not manually verify the connection, she could be risking her privacy in case of a malfunction.

We recommend displaying the user's forged/real location


Orbot
Make advanced information appear when the displayed location is touched
Display current location
powered by The Tor Project
Connected to the Tor network
Download
Log
Upload

You appear to be in FRANCE

Exit Node Information

IP address:	1.2.3.4
Country:	FRANCE
Uptime:	8d 12h
Named:	YES
Bandwidth:	600 KB/s
ORPort:	6969
DirPort:	9696

Figure 1: Orbot with our proposed modifications

when Orbot is activated/deactivated. Advanced connection information would be available for interested users by tapping on the displayed location (Fig. 1). Having location information readily available in the main screen is intuitive, eliminates the need for manually checking the connection, and makes the user constantly aware of the app's status. This design potentially decreases the chances of a user erroneously assuming she is anonymous.

We recommend using more natural language. For example, on the *Exit Node*, we could provide the user with a list of available nodes. The list should be filterable by country, bandwidth, etc. Upon choosing an exit node, the user should be presented with feedback about her new location.

A second problem is that the link between Orbot and Orbot-enabled apps is unclear. A user deactivating Orbot may end up with a non-operational Orbot-enabled app. Thus when deactivating Orbot, the user should be alerted of apps that are configured to use Orbot and prompted to return to the apps' non-Torified mode if possible.

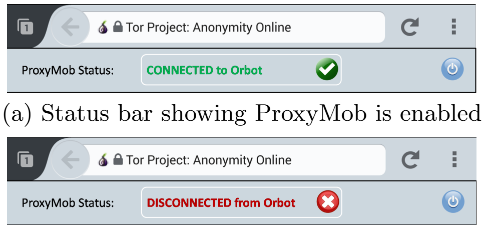
4.2 Orweb

Orbot's wizard prompts users to download Orweb directly with a user-chosen browser. The user is prompted to modify their device settings to allow installation from unknown sources, which makes the user more vulnerable to downloading malicious software. We recommend directing the user to Orweb's Play Store page instead.

4.3 ProxyMob

The most critical issue with ProxyMob is that it is not visible to users once installed, so users have no indication of whether ProxyMob is enabled or functioning properly. Due to ProxyMob's invisibility, a user may mistakenly assume her Firefox traffic is anonymized just by activating Orbot. On the other hand, if a user wrongfully assumes deactivating Orbot is sufficient to return to normal browsing, Firefox is remains in a non-operational state.

To make it more visible, we propose placing a ProxyMob status bar beneath Firefox's address bar. As shown in Fig. 2, the status bar informs the user whether the add-on is enabled (i.e., the user is connected to Orbot) or disabled (i.e., disconnected from Orbot). The status bar has a power button which the user can use to enable or disable the add-on. The ideal status bar would also include information about the user's location.


Tor Project: Anonymity Online

ProxyMob Status: CONNECTED to Orbot ✓

(a) Status bar showing ProxyMob is enabled

Tor Project: Anonymity Online

ProxyMob Status: DISCONNECTED from Orbot ✗

(b) Status bar showing ProxyMob is disabled

Figure 2: The proposed ProxyMob status bar

At installation, Orbot presents a list of browsers to use for downloading ProxyMob. However, installing the add-on is only successful if it was downloaded from Firefox. We recommend automatically downloading ProxyMob through Firefox since this is the only viable option.

5. CONCLUDING REMARKS

Our study of Tor-enabled mobile apps revealed a number of usability issues that may intimidate users, as well as endanger their privacy and security. In general, software interfaces should shield users from underlying technical details. More specifically, Tor users should be able to fully benefit from Tor-enabled tools regardless of their knowledge of Tor's complicated infrastructure. Since the degree of privacy offered by the Tor network depends on the number of participating users, the usability of Tor-tools has become a fundamental requirement to attract more users. As is the case with any software interface, a Tor tool should not impose a cognitive load on the user; it should be easy to use and understand, and guide the user through the steps required to protect her privacy. Such tools should also speak the user's language and avoid using unintelligible technical terms. In addition, a good security software interface should be informative, providing the user with visual cues and comprehensible feedback that allows her to be consistently aware of the tool's status and protect her from wrong assumptions about her security and privacy.

Accordingly, we proposed modifications that address the unintuitive feel of the apps, the apps' technical language, and the insecure options that risks users' security and privacy. As an extension to the work presented in this poster, we plan to perform a user study to test the usability of our proposed modifications to Orbot, Orweb and ProxyMob.

6. REFERENCES

[1] Tor project: Anonymity online. `https://www.torproject.org`. [Accessed Nov-2013].

[2] Clark, J., Van Oorschot, P. C., and Adams, C. Usability of anonymous web browsing: an examination of TOR interfaces and deployability. In *Symp on Usable Privacy and Security*, ACM (2007), 41–51.

[3] Dingledine, R. Tor: The second-generation onion router. In *USENIX Security Symp* (2004).

[4] Whitten, A., and Tygar, J. D. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *USENIX Security Symp* (1999).

[5] Yee, K.-P. User interaction design for secure systems. In *Information and Communications Security*, LNCS. Springer Berlin Heidelberg, 2002, 278–290.