Establishing Darknet Connections:
An Evaluation of Usability and Security

John Bethencourt   Wai Yong Low   Isaac Simmons   Matthew Williamson

{bethenco,ids}@cs.cmu.edu, {wlow,mcwillia}@andrew.cmu.edu

Introduction

In many applications, hosts in a peer to peer network may wish to maintain their anonymity or the privacy of their queries. In some applications, an even stronger guarantee is desirable: hosts would like to prevent others from determining whether they participate in the network at all. Darknets, or friend-to-friend networks, are one approach to preventing the discovery of hosts within a peer to peer network [1]. In such a network, hosts only form Internet connections with and directly communicate with a small set of hosts whose operators are known and trusted a priori. That is, each user only connects to her friends, trusting that her friends will not reveal her identity or existence in the network.

Several current peer to peer networks employ this concept; however, establishing the trusted connections between nodes is a difficult process. As an example, in order for someone to join a W.A.S.T.E. network, they must generate a key pair and manually exchange (large) public keys with a friend already within the network. Little guidance is provided during these tasks, suggesting many users will have difficulty completing them and understanding their security implications. Motivated by the difficulty users are likely to encounter in establishing trusted connections to friends in a darknet, this project is an investigation into the usability and security of three potential methods for doing so which are inspired by existing darknet client software [2].

Methods for Establishing Connections

We assume a user Alice is already a member of a darknet and her friend Bob wishes to connect to the darknet through her. We will consider the process of connection establishment to be complete when Alice and Bob each have the IP address and port at which the other can be contacted in the future and a public key which can be used to encrypt communications to the other. If both the users involved already have public keys signed by a CA trusted by both of them, the problem is trivial. However, the vast majority of users do not already have a public key certificate, have no simple way of obtaining one, and likely would not want to bother if they did. Thus, we assume no such infrastructure is available.

To begin, the users will have to communicate some information through some out-of-band channel with some assumed level of security. Specifically, the client software may

generate an "invitation code" encoding IP address, port, and public key fingerprint in a short, printable string for convenience. Assuming base64 encoding, 35 characters will suffice for an invitation code with an address, port, and a 160-bit SHA-1 hash of the public key. Several possible channels (e.g., email, telephone calls, transport on a USB flash drive) exist for the initial communication of such a code, with varying security and convenience. The three general methods for using such a channel to establish a trusted connection between friends in a darknet that we consider in this work are 2-way key exchange, 1-way key exchange, and "conversational", as described below.

2-way key exchange.

Alice generates an invite code as described above and gives it to Bob, when then generates a second code and returns it to Alice. Provided the out-of-band channel preserves the *integrity* of the invite codes, Alice and Bob can be certain that they do in fact end up with each other's public keys.

1-way key exchange.

Alternatively, Alice may generate a single use invitation for Bob and record it in a list of pending invitations with his name before sending it out. When he connects to Alice, he may send the invitation he is using along with his public key encrypted for Alice, and Alice may accept his key if the invitation has not previously been used. In this case, if we assume that the out-of-band channel both protects the *integrity* of the invite code and maintains its *secrecy* until Bob can connect, then we will arrive at a secure state.

Conversational.

As a middle ground between the (apparent) high security and low convenience of a 2-way key exchange and the low security and high convenience of a 1-way key exchange, we consider an additional method that we term "conversational". In this method, Alice makes a single user invitation for Bob as before. However, once he connects, rather than immediately accepting the connection, Alice asks him some questions that only he is reasonably likely to be able to answer. If he answers correctly, Alice accepts the connection. If we assume that the out-of-band channel preserves only the *integrity* of the invite code, then Bob may be assured of Alice's public key. Then if we assume the answers to the questions asked by Alice would only be known by Bob, Alice may also be assured of Bob's public key.

User Study Design

In order to evaluate the usability of these three methods of connection establishment, we have designed and piloted a

comparative user study. In the study, users complete a set of tasks related to joining and establishing trust on a network using a software mockup of a darknet client that we developed for the study. A mockup is used in order to reduce the degree to which we are evaluating the specific user interfaces of existing darknet clients and instead focus on the differences inherent to the connection establishment methods. The tasks required for this experiment can be completed by pressing an "add friend" button, which presents the user with a dialog that allows them to invite a friend, or to enter an invitation code sent to them by a friend. When inviting a friend, users are given an "invitation code" and told to give it to their friend.

The mockup may be started in one of three modes, causing it to employ any one of the above methods of connection establishment when the user chooses to invite a friend to the network or receives an invitation. Under the 1-way key exchange condition, users must simply enter or send the invitation codes and the task is completed. Under the conversational interface, after entering the code, users are presented with a question / answer dialog where they must chat with the other user and both are given the option to accept or reject the trust negotiation. During the experiment, an experimenter on another computer plays the role of the other user. In the 2-way key exchange condition, there is no conversation, but the user must both accept and send an invitation code for each task instead of only requiring one-way transmission.

The users are provided with printed sheets containing a back story which explains a hypothetical scenario in which they wish to disseminate documents while maintaining their anonymity, motivating the use of a darknet. The back story describes a darknet suitable for this purpose at an intuitive level, in a manner similar to how a user may be introduced to a real life darknet by another non-technical user that nevertheless understands the basic premise of a darknet. The sheets go on to instruct the user to perform several tasks involving joining the network by accepting invitations and generating new invitations to bring others online. The instructions describe the goal of the task from a high-level and provide no direction specific to the trust establishment method currently in use by the software mockup. The user is also provided with an email client (a GMail account) to be used in sending and receiving invitation codes.

As they work on completing the tasks, users are encouraged to explain what they are thinking. Camtasia is used to record screen events during the test for later evaluation and timing data. After a user completes all the tasks for under a particular trust establishment method, a survey is given to evaluate their impressions of the software. In addition to questions about the usability of the software, the survey asks the user questions about their perception of its level of security.

Pilot Study Results

Currently, the study has been piloted on six participants. All were college students with an average age of 22.5. All were relatively technically savvy and had prior experience using peer to peer software; however, none had prior experience with darknet software or were familiar with the concept of a darknet. Due to the small scale of the initial pilot study, we had each participant go through all the tasks three times, once for each of the trust establishment methods. They were
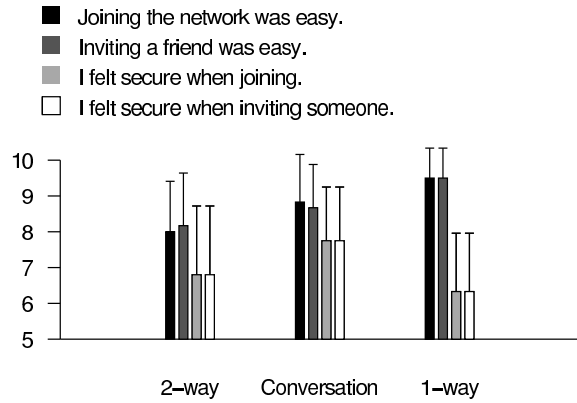

Figure 1: A sampling of results from the pilot study.

each presented with the three conditions in a different order (each of the six permutations was exercised exactly once) to cancel out learning effects. All users were able to successfully complete all tasks within the allotted time. The task completion times were significantly higher for the 2-way key exchange as compared to the other two methods. This is to be expected since there are inherently more steps involved in this process. Additionally, the conversational interface took slightly longer than the 1-way exchange. Again, this is to be expected since the conversational tasks are a strict superset of the tasks required in the 1-way situation.

Figure 1 displays a small sampling of the results obtained from the surveys administered as a part of the pilot. The four statements shown were presented to each participant after they completed all of the tasks using a single trust establishment method. The statements were answered with an integer from 1 to 10, with 10 indicating the highest level of agreement. As expected, the users found the 2-way key exchange to be the least easy to use and the 1-way key exchange to be the most easy to use, with the conversational interface falling somewhere in-between. This trend was observed in all ease of use metrics and was marginally significant on most questions across the six subjects we measured.

In metrics related to perceived security, users rated the 1-way key exchange significantly lower than the other two conditions. More surprisingly, users felt more secure with the conversational interface than with the 2-way key exchange. This trend was also reflected in all questions measuring perceived security. Although the precise security properties of the 2-way key exchange depend on the out-of-band channel used to transmit the keys, in most cases it may be considered more secure than the conversational method of trust establishment. Understanding this gap between the security properties of the system and user perceptions is an interesting area of continued work.

References

[1] P. Biddle, P. England, M. Peinado, and B. Willman. The Darknet and the Future of Content Protection. *ACM Workshop on Digital Rights Management*, 2002.

[2] B. Popescu, B. Crispo, and A. Tanenbaum. Safe and Private Data Sharing with Turtle: Friends Team-Up and Beat the System. *Security Protocols Workshop*, 2004.