Analysis of the Web of Trust

James Parker∗

Department of Computer Science, University of Maryland
College Park, MD 20742 USA

November 12, 2013

1 Introduction

The Web of Trust is a network that aims to establish the identity of its users in a decentralized manner. It is decentralized because there is no central entity that controls the network and is responsible for verifying the identity of users. Instead, each user verifies the identity of other people in the network who the original user knows. As a result, the Web of Trust is a social network where users interact to confirm the identity of other users in the graph. This is advantageous for establishing trust since trust can be distributed across many users in the graph instead of a single entity.

The most recent Web of Trust network data I could find is for February 7, 2012. Each vertex is the user’s identity which includes the user’s name, email address, and public cryptographic key. Each edge is where one user verifies another user’s identity. Technically, the user is digitally signing the other user’s identity. Originally, this dataset 46,791 vertices and 473,292 edges. Unfortunately, NodeXL could not handle a network this big on my machine so I wrote a Ruby script to randomly sample 10,000 vertices. This reduction left 20,325 edges in the network.

2 Headlines

2.1 The Web of Trust fails to maintain decentralization.

In practice, Web of Trust users typically recognize certain users as completely trusted and will then trust identities verified by their trusted users. As a result, most users converge on the same trusted users, placing a significant amount of trust in these few users. Figure illustrates this problem. This visualization is an overview of the entire Web of Trust network.

∗jprider@cs.umd.edu
http://www.lysator.liu.se/~jc/wotsap/download.html
studied. The vertices are laid out using the Fruchterman-Reingold algorithm. Both the vertex color and size depend on betweenness centrality. Blue represents high centrality while orange represents the converse. To emphasize the relatively low number of high centrality nodes, the vertex layout order depends on betweenness centrality.

As seen in the figure, there are relatively few large, blue vertices while most of the nodes are bright orange. This shows there is a significant imbalance of trust between most users and a few central users. This finding is at odds with the goal of the Web of Trust to distribute trust evenly across all of its users. To prevent this convergence on the trusted few, users need to be made aware of this problem and should prefer to place trust in users with lower centrality.

2.2 Average users rely on the trusted few.

In Figure 2 I begin to manipulate the position of vertices based on different attributes. The vertex position on the horizontal axis indicates the number of incoming edges while the vertical axis indicates the number of outgoing edges. Color and size is again used to show betweenness centrality. Average users with lower centrality can be seen as orange, and the trusted few are blue.

The position of the vertices appears to satisfy a linear trend line, however if you look at the trend line only associated with average users, the slope is fairly flat. This suggests that they are receiving more verifications than they are giving. In order to balance the number of incoming and outgoing edges, average users rely upon the trusted few to verify others. This puts further power in the hands of this smaller set of people. To reduce this effect, average users should make an effort to verify each other’s identities.
Figure 2: Web of Trust network where the vertex position on horizontal axis indicates the number of incoming edges and the vertical axis indicates the number of outgoing edges.

Figure 3: Web of Trust network grouped by cluster.
2.3 Many clusters in the Web of Trust are completely disconnected from the rest of the network.

Figure 3 shows the Web of Trust network grouped by cluster. Once again, vertex color and size depend on betweenness centrality. Edges between each group are combined.

The visualization shows that there a large number of smaller clusters in the network are completely disconnected from the rest of the network. This can be a problem if users want to establish a trusted identity for users in the smaller clusters. Since there is no path from these smaller clusters to the larger clusters, there is no way to establish this trusted identity over the network. Now that these smaller clusters have been identified, users can attempt to bridge these disconnected clusters to establish a unified network.

3 Critique

In my experience in using NodeXL, I found the software to be an excellent tool to create insightful visualizations while dealing with large datasets. In particular, it offers informative feedback by continuously updating the excel workbook and the graph visualization whenever changing the display options. By caching metrics in the workbook, speed was manageable with a dataset this large while the continuous updating occurred.

While this tool is great, there are a few areas that could be improved. The first issue I had is that when I attempted to compute metrics on the original dataset, Excel crashed. Here NodeXL failed to offer simple error handling by failing to catch this error and recover gracefully. I also thought that NodeXL could strive for better consistency by offering an option to export the visualization to a pdf format, which would integrate better with other software.