Visual Analysis of Photo Policy Misconfigurations Using Treemaps

Yousra Javed
Illinois State University
School of Information Technology
Normal, IL 61790 USA
yjaved@ilstu.edu

Mohamed Shehab
University of North Carolina
Charlotte
Department of Software &
Information Systems
Charlotte, NC 28223 USA
mshehab@uncc.edu

Abstract

Online photo privacy is a major concern for social media users. Numerous visualization tools have been proposed to help the users easily compose and understand policies on social networks. However, these tools do not incorporate the ability to quickly identify and fix unintended photo sharing. We propose a tool that displays the photo albums w.r.t their policy misconfigurations using a Treemap visualization.

Introduction

Photos are one of the most frequently shared content on social media [1]. Braunstein et al. [3] conducted a study using indirect content privacy surveys. Their results indicated the users' privacy concern for their online photos. Being non-experts in access control—the users frequently end up giving access to unintended audience. Unintended sharing corresponds to both over-sharing or under-sharing of information. This suggests the need for techniques that can enable a user to quickly figure out the unintended audience accessing a photo and fix it.

Numerous visualization tools have been proposed to help the users easily compose and understand policies on social networks. PViz is a tool proposed by Mazzia et al. [6] which lets the user understand the visibility of her profile according to various friend groups. Similarly, Anwar et al. [2]

proposed a visualization tool for privacy settings. The social network is displayed as a graph, and mousing over a node in the graph indicates what the person can access in the current user's profile. However, these tools do not incorporate the ability to quickly identify and fix unintended photo sharing.

A Treemap [7] displays hierarchical (tree-structured) data as a set of nested rectangles. Each branch of the tree is given a rectangle, which is then tiled with smaller rectangles representing sub-branches. A leaf node's rectangle has an area proportional to a specified dimension of the data. When the color and size dimensions are correlated in some way with the tree structure, one can often easily see patterns that would be difficult to spot in other ways, such as if a certain color is particularly relevant. A second advantage of Treemaps is that, by construction, they make efficient use of space. As a result, they can legibly display thousands of items on the screen simultaneously.

We propose a tool that displays the photo albums w.r.t their policy misconfigurations using a Treemap visualization. We developed a prototype for Facebook by extracting the privacy settings of photo albums through the Facebook API.

Potential Policy Misconfigurations

We use the following set of misconfigurations proposed by Javed et al. [5]. Each misconfiguration is assigned a sensitivity weight on a scale from 0 to 1 based on the extent of privacy leak that it can cause (Table 1).

M1 An empty friend list has been allowed or denied. For example, if the user allows access to a smart list which contained friends initially, but became empty over the course of time when the respective friends changed their work, education or hometown.

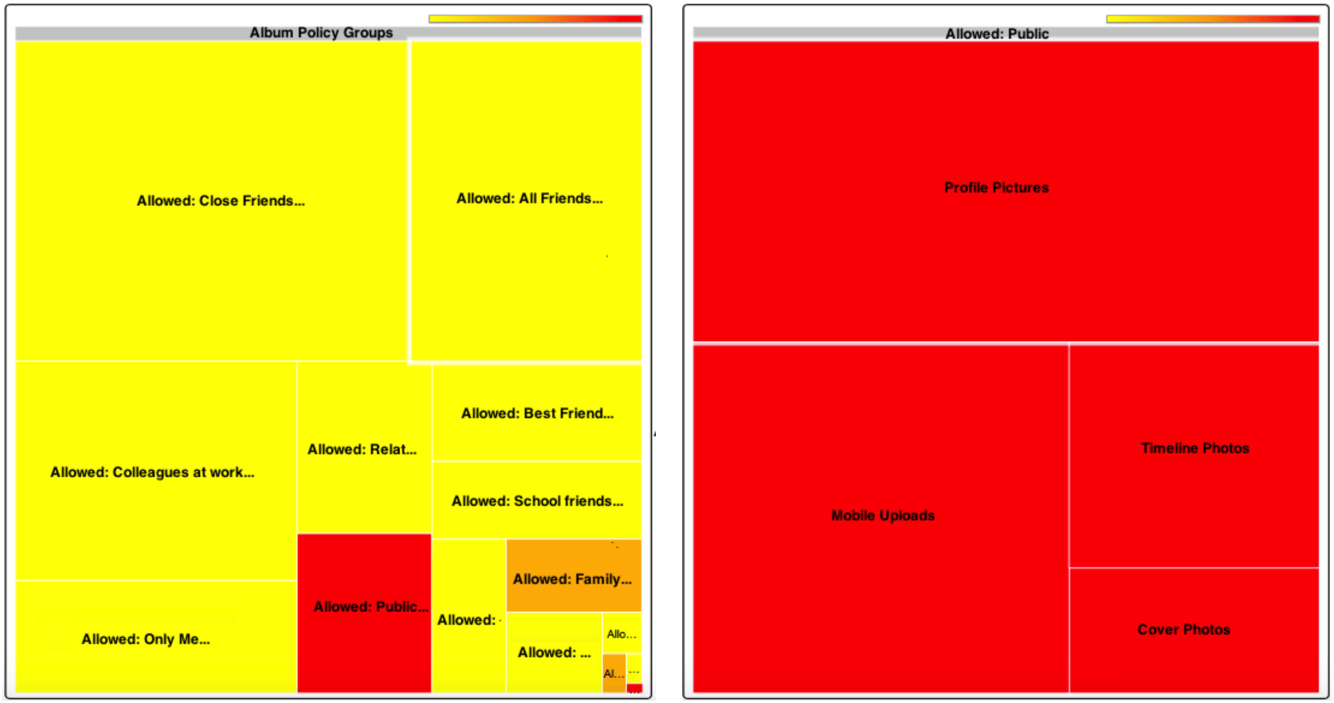
Table 1: Misconfiguration sensitivities

Misconfiguration	Sensitivity
M1. A friend list having no friends is allowed or denied	0.1
M2. One or more friends exist in both the allowed and denied fields	0.2
M3. There are common friends between friend lists	0.4
M4. A friend has been denied explicitly in this album, but is allowed in other albums	0.6
M5. An automatically created friend list which updates without user concern has been used	0.8
M6. The album is visible to people outside the friend network	1

M2 A friend has been explicitly denied access to this photo album, but is allowed access to other albums. This misconfiguration can arise due to the existence of a friend who has been given access to most albums but is denied access to a specific album by adding their name in the denied set of users. The user can have a false perception that this friend has been denied access to all the other albums as well.

M3 There are common friends between friend lists. For example, Facebook gives access to the union of all the friends/friend lists included in “Allowed” field. However, during policy composition, the user might want to allow access to only those friends who exist in all the friend lists in the “Allowed” field, leading to over-sharing of information.

M4 One or more friends exist in both the allowed and denied fields. In Facebook access control, the *deny*



(a) Level 1- Album Policy Groups

(b) Level 2 - Albums Inside an Album Policy Group

Figure 1: Treemap visualization of photo albums w.r.t their policy misconfigurations

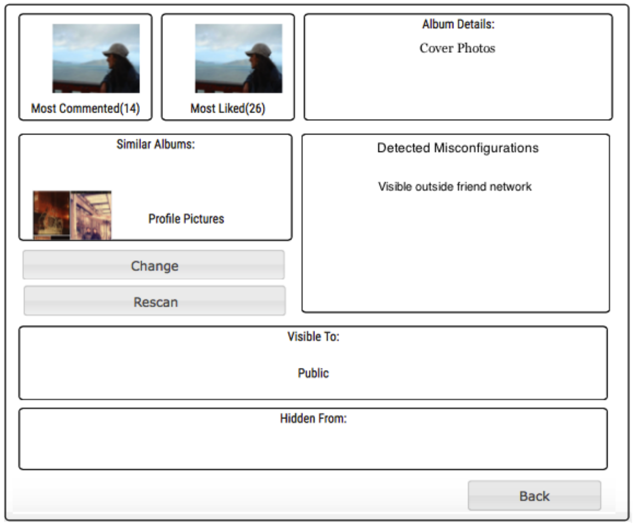
permission takes precedence over the *allow* permission. Therefore, this misconfiguration intends to notify the user of unintended denial of access to a friend.

M5 A smart list, which updates automatically without user consent, has been used. If a user grants access to a smart list, there is a potential chance of sharing information with unintended audience because the user does not control the list.

M6 The photo album is visible to people outside the friend network. This scenario is possible if the chosen privacy settings are Friends of Friends, Friends and Networks, or Public.

Treemap Visualization

We use a Treemap to visualize the photo albums w.r.t their policy misconfigurations.



Album Details: Cover Photos

Similar Albums: Profile Pictures

Detected Misconfigurations: Visible outside friend network

Visible To: Public

Hidden From:

Figure 2: Album Summary Screen

Treemap Levels

There are two levels in our Treemap hierarchy.

- Level 1 - The first tree level displays the albums grouped by their policies. Each rectangle in this level corresponds to a distinct policy. The size of the rectangle is proportional to the number of albums having this policy. The color of the rectangle shows the sensitivity of the misconfiguration detected in the underlying policy, where red represents the highest sensitivity and yellow represents the lowest sensitivity (Figure 1(a)).
- Level 2 - The second level displays all the albums inside an album policy group. Each rectangle corresponds to a single album. The size of the rectangle is proportional to the number of photos inside the album. The color of each rectangle is the same as that of the parent rectangle in level 1 and shows the sensitivity of the misconfiguration detected in the album. Figure 1(b) shows the albums inside the policy group “public". Clicking on a rectangle in this level takes the user to the album’s details screen. This screen displays a summary of the album consisting of its description, policy, detected misconfigurations, most liked and commented photos, and similar albums. There is also an option to fix the album policy and rescan it (Figure 2).

Pilot Study

In order to evaluate our prototype, we are designing a pilot study that answers the following questions:

1. Compared to the existing privacy configurations interfaces on social media, is the Treemap visualization more effective w.r.t users’ speed and accuracy of identifying and fixing album policy misconfigurations?
2. What is the users’ subjective satisfaction of our prototype based on the System Usability Scale (SUS) [4]?
3. Can we use eye-gaze tracking to analyze user attention towards our prototype?

REFERENCES

1. 2014. Simply Measured Facebook Study. (2014). `http://www.slideshare.net/simplymeasured/q3-2014-facebook-study`
2. Mohd M. Anwar and Philip W. L. Fong. 2012. A visualization tool for evaluating access control policies

in facebook-style social network systems. In *Symposium On Applied Computing (SAC)*. 1443–1450.

3. Alex Braunstein, Laura Granka, and Jessica Staddon. 2011. Indirect content privacy surveys: measuring privacy without asking about it. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*. ACM, 15.
4. John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
5. Yousra Javed and Mohamed Shehab. 2013. Access Control Policy Misconfiguration Detection in Online Social Networks. In *Social Computing (SocialCom), 2013 International Conference on*. IEEE, 544–549.
6. Alessandra Mazzia, Kristen LeFevre, and Eytan Adar. 2012. The PViz comprehension tool for social network privacy settings. In *Symposium on Usable Privacy and Security (SOUPS)*. 13.
7. Ben Shneiderman and Catherine Plaisant. 1998. Treemaps for space-constrained visualization of hierarchies. (1998).