Learning Based Access Control in Online Social Networks

Mohamed Shehab,
Gorrell Cheek & Hakim Touati
College of Computing & Informatics
University of North Carolina Charlotte, NC
{mshehab, gcheek, htouati}@uncc.edu

Anna C. Squicciarini
College of Information Sciences & Technology
Pennsylvania State University
University Park, PA
acs20@psu.edu

Pau-Chen Cheng
IBM T.J. Watson Research Center
Hawthorne, NY
pau@us.ibm.com

ABSTRACT

Online social networking sites are experiencing tremendous user growth with hundreds of millions of active users. As a result, there is a tremendous amount of user profile data online, e.g., name, birthdate, etc. Protecting this data is a challenge. The task of access policy composition is a tedious and confusing effort for the average user having hundreds of friends. We propose an approach that assists users in composing and managing their access control policies. Our approach is based on a supervised learning mechanism that leverages user provided example policy settings as training sets to build classifiers that are the basis for auto-generated policies. Furthermore, we provide mechanisms to enable users to fuse policy decisions that are provided by their friends or others in the social network. These policies then regulate access to user profile objects. We implemented our approach and, through extensive experimentation, prove the accuracy of our proposed mechanisms.

Categories and Subject Descriptors: C.2.0 [Computer Communication Networks]: General-security and protection; D.4.6 [Operating Systems]: Security and Protection-Access Controls; K.6.5 [Management of Computing and Information Systems]: Security and Protection

General Terms: Security

Keywords: Privacy, Social Networks, Supervised Learning

1. OVERVIEW

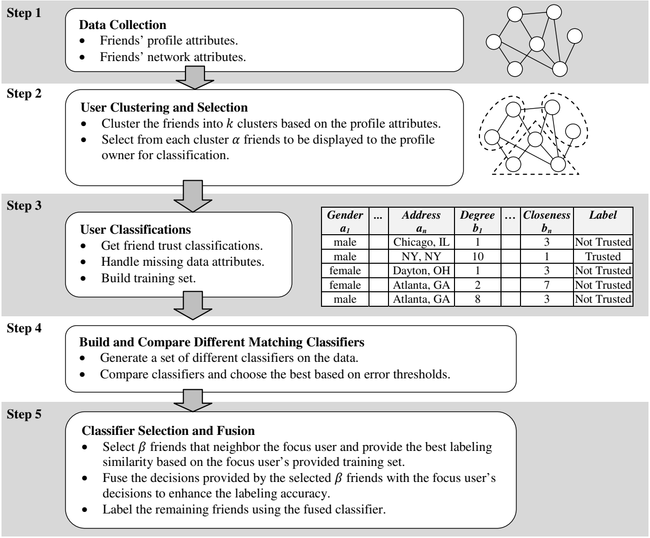
With the growing size and adoption of social networks, users are continuously updating their profiles by adding friends and posting new objects. For example, on Facebook alone, over 2 billions pieces of content (web links, notes, photos, etc.) are shared each week. This coupled with the fact that the average user has 130 friends makes it a challenging effort in managing access to user information. In order to guarantee fine-grained protection, a user has to specify a policy every time an object (e.g., photo) is added to their profile or they establish a new friendship. Maintaining an effective user access control policy can be a very laborious and tedious task. As a result, policies are only partially configured and maintained. Or, they may be all together ignored. This leads to user content not being properly protected and potentially unknowingly made available to unintended recipients.

Instead of asking the user to decide for each of her friends who to give access or not, our proposed approach only requires the user to choose the access rights (or label) of α carefully selected users from her friend's list. Our proposed approach uses *supervised learning* mechanisms to decide on the access control policy settings for the remaining users. (Note: Hereafter, we refer to the profile owner as the *focus user*.) The steps involved in the learning based policy management process are described in Figure 1.

In step 1, for each focus user's friends, the profile attributes A_j are collected and the network attributes B_j are computed based on the generated social graph information. In step 2, the collected attributes $\{A_j, B_j\}$ are used to cluster the focus user's friends into K non-overlapping clusters. The clustering is performed to ensure that the focus user labels representative members of each of the computed clusters. From each of the clusters, we select α representative friends. In step 3, the focus user is then asked to classify (or label) each of the α friends, that is to indicate which of the friends are trusted to access a specific object and which are not trusted. The labeled α friends are added to the training set Θ. For each of the labeled friends, there might be some missing attributes. We use heuristics to estimate such missing attributes. For example, a user's missing age could be estimated as the average of all their friends' ages. In step 4, the training set Θ can be used directly to train a classifier. However, there are several classifier algorithms and it is crucial to select the classifier that is most suited for this specific user instance. So, the mechanism we adopt is to train and tune several classifiers and then compare their performance based on standard cross validation methods such as n-fold cross validation [2].

In step 5, the knowledge accumulated by other users in the social network can be utilized to further enhance the classifier's accuracy. It is important in this step to seek classification advice from other friends who classify users similarly to the focus user. This is referred to as the *selection process* where β other user classifiers are selected based on their accuracy in labeling the focus user's training set Θ. In the *fusion process*, the decisions of the selected β classifiers are combined with the decisions of the focus user's classifier to classify the remaining focus user's friends.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.



Step 1

Data Collection
- Friends' profile attributes.
- Friends' network attributes.

Step 2

User Clustering and Selection
- Cluster the friends into k clusters based on the profile attributes.
- Select from each cluster α friends to be displayed to the profile owner for classification.

Step 3

User Classifications
- Get friend trust classifications.
- Handle missing data attributes.
- Build training set.

Gender a_1	...	*Address* a_n	*Degree* b_1	…	*Closeness* b_n	*Label*
male		Chicago, IL	1		3	Not Trusted
male		NY, NY	10		1	Trusted
female		Dayton, OH	1		3	Not Trusted
female		Atlanta, GA	2		7	Not Trusted
male		Atlanta, GA	8		3	Not Trusted

Step 4

Build and Compare Different Matching Classifiers
- Generate a set of different classifiers on the data.
- Compare classifiers and choose the best based on error thresholds.

Step 5

Classifier Selection and Fusion
- Select β friends that neighbor the focus user and provide the best labeling similarity based on the focus user's provided training set.
- Fuse the decisions provided by the selected β friends with the focus user's decisions to enhance the labeling accuracy.
- Label the remaining friends using the fused classifier.

Figure 1: Learning based policy management process

2. EXPERIMENTAL RESULTS

We performed experiments on both the Last.FM and Facebook platforms. A data set was collected by crawling the Last.FM social network. This data set was used for empirical testing of the our learning based approach for access control in social networks. We assessed how different classifiers performed on the Last.FM data set and how classifier fusion mechanisms influenced the overall results. In addition, we developed a proof-of-concept interactive Facebook application that collects user training data, i.e., requests the user to label certain friends as either trusted or non-trusted.

In our Last.FM experiments, the following profile attributes were obtained: Age, Gender, and Home Country. In addition, each focus user's social graph was built and a series of network metrics were computed on their respective social graph, e.g., Degree, Betweenness, Closeness, etc. Nine different classifiers were trained: Naive Bayes, BayesNet, Radial Basis Function Network, K Star, AD Tree, Support Vector Machine, Naive-Bayes Tree, Random Forest and Decision Table. The classifiers were tested by labeling each friend in the test set as either trusted or non-trusted based on the users' profile attributes and network metrics. The results were compared against the labels in the training set Θ. The true positive, true negative, false positive, and false negatives for each classifier were recorded. The classifier with the highest accuracy is selected, which is denoted as f^*_Θ.

Following the section of f^*_Θ, additional classification advice is sought from the focus user's friends. For each focus user, β friends (between 10-40) were selected from the focus user's friend set who classify users similarly as the focus user. These β additional classifiers are fused with the focus user's classifier f^*_Θ to label the remaining friends as either trusted or non-trusted. In our experiments, we adopted the following classifier fusion algorithms: *group voting*, *most confident*, and *group confidence product* [1].

Figures 2 shows the results generated for the different classifiers using a training set (α) equal to 20% and 10 selected friends (β). From these results, we are able to obtain approximately 70% accuracy without fusing additional friends' classifiers with the focus user's classifier f^*_Θ. However, by fusing the focus user's friends' classifiers, accuracy improved to around 83%. Based on these results, it is evident that our fusion base approach improves the classification result, with the voting based approach leading the other fusion mechanisms. Furthermore, the AD Tree classifier provides the highest accuracy results. We conducted numerous other experiments whose results are very promising.

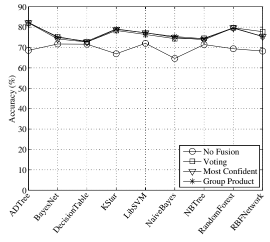


Figure 2: Classifier Type vs. Accuracy

3. CONCLUSION

We presented a supervised leaning based mechanism that assists users in composing and managing their access control policies in online social networks. Our approach assists users in deciding trust and access control settings for each of their friends. Moreover, we incorporated knowledge from others in the social network to enhance the supervised learning results. Finally, we demonstrated the feasibility of our approach by implementing it on two different social networking sites. In our experiments, we used a large data set from Last.FM giving us a more precise understanding of the varying trends and user behaviors. We also implemented our framework's user labeling as a Facebook application.

4. REFERENCES

[1] J. Kittler, M. Hatef, R. P. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[2] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, second edition, 2005.