Privometer: Privacy Protection in Social Networks

Nilothpal Talukder, Mourad Ouzzani, Ahmed K. Elmagarmid, Hazem Elmeleegy, and Mohamed Yakout

Purdue University, West Lafayette, IN, USA

{ntalukde,mourad,ake,hazem,myakout}@cs.purdue.edu

***Abstract*— The increasing popularity of social networks, such as Facebook and Orkut, has raised several privacy concerns. Traditional ways of safeguarding privacy of personal information by hiding sensitive attributes are no longer adequate. Research shows that probabilistic classification techniques can effectively infer such private information. The disclosed sensitive information of friends, group affiliations and even participation in activities, such as tagging and commenting, are considered background knowledge in this process. In this paper, we present a privacy protection tool, called *Privometer*, that measures the amount of sensitive information leakage in a user profile and suggests self-sanitization actions to regulate the amount of leakage. In contrast to previous research, where inference techniques use publicly available profile information, we consider an augmented model where a potentially malicious application installed in the user's friend profiles can access substantially more information. In our model, merely hiding the sensitive information is not sufficient to protect the user privacy. We present an implementation of Privometer in Facebook.**

I. Introduction

Online social networks have paved the way for people to stay connected with their friends, mingle with others having similar interests, and share personal information and experiences. To address privacy concerns, users can use privacy settings and hide sensitive information. However, it has been shown [1][2] that such measures are not sufficient to protect one's privacy due to the friendship relations, group memberships, or even participating in activities like photo tagging and commenting, which can be harvested through 'screen-scraping' [3] or other means. Social ties represented as network data can be exploited by the adversary to predict the value of the sensitive attributes through a wide array of network classification techniques [4], [5].

To promote different aspects of usage, social network platforms are now allowing independent developers to build applications on their platforms, e.g., Facebook API and Orkut OpenSocial API developer platforms. Through these APIs, third party applications have now access to personal information that they may not even need. In May 2008, the technology program 'Click' on BBC [6] demonstrated how a malicious application masquerading as a harmless application harvested personal data without the user even knowing about it. A study by Felt et. al. [7] shows that 90.7% of applications are being given more privileges than they need. Clearly, malicious applications can have access to more information than the information simply obtained by 'screen-scraping'. Moreover, such malicious applications can more effectively infer sensitive information from profiles of users who did not even install them in their profiles.

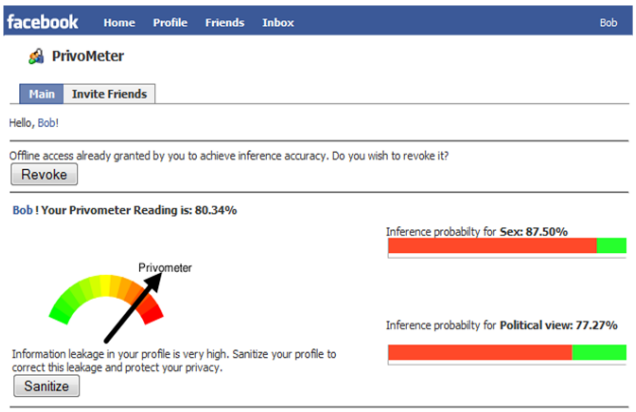
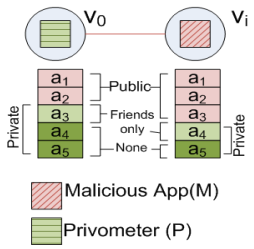
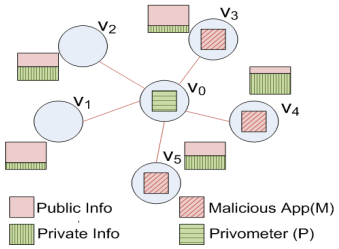


Fig. 1: Privometer on Facebook

To address privacy concerns in social networks including the case where a potentially malicious application is installed in the profiles of the user's friends, we present Privometer[1] (Fig. 1), a novel privacy protection tool for social networks that can (i) measure the amount of sensitive information leakage based on the relationship information and private information of the user's friends and (ii) offer users a unique feature to regulate this leakage through a suggested list of actions referred to as self-sanitization actions.



(a) Information Visibility in User Profile



(b) Adversary Model with Malicious Application

Fig. 2: User's Privacy Control and Adversary Model

Privometer, installed in a user profile (v_0), can access private information from the friend profiles that are made available to friends only (Fig. 2a). However, a malicious application installed in a friend's profile may try to infer the user's private information that is not revealed to anyone including the user's friends. In Fig. 2b user v_0 is shown to have five friends $\{v_1, ..., v_5\}$ having various amount of private and public infor-

[1]Privometer can be accessed at http://apps.facebook.com/privometer.

mation in their profiles. The users v_3, v_4 and v_5 have installed an application M. M has access to all private information of users v_3, v_4 and v_5, even though they have not made it publicly available. v_0 has not added M to her profile; yet, her private information can be more accurately inferred by M due to this additional knowledge. Privometer assumes that the malicious application runs an inference algorithm in the background using known inferences models. Since Privometer is unaware of the inference model the adversary might use, it considers a list of the best known models [2] and determines sensitive information leakage using any of these models. Then it selects the model which most accurately infers the user's sensitive attributes; in other words, the model that causes the most damage to the user in terms of leakage. Finally, based on the combined probability of sensitive attribute inference using the best chosen model, Privometer presents to the user a measure of her privacy, a ranking of her friends based on individual contributions to privacy leakage, and self-sanitization actions to lessen this leakage.

Previous work on sensitive attribute inference [1][2][8] use publicly available data only. Due to the lack of sensitive information availability in public profiles, researchers took resort to applying learning methods on synthetic data. He et. al. [1] considered a hypothetical attribute and assumed homogeneous Conditional Probability Table (CPT) for immediate friends of a user in a Bayesian Network Structure. In the network, a friend is considered a child node of the user (parent node). The assignment of a node's value is made based on parent's value and node's assigned probability conditioned on parent's value. Since our work considers a malicious platform application as adversary and the augmented model, the adversary obtains additional knowledge of some users' sensitive information (obtained through the social network platform API) in addition to their public profile information. Through this data collection approach we can build real-world data set and avoid generating synthetic data. Felt et. al. [7] proposed a data hiding scheme for third party developers to be implemented by the social network platform called privacy-by-proxy to preserve anonymity of user data. However, this approach have not been considered by any platform due to the potential negative influence on the social network's growth and performance.

Our contributions in this paper are:

- To the best of our knowledge, Privometer is the first functional prototype of a privacy measuring tool to be implemented on a social network platform.
- We provide a realistic insight to the sensitive attribute inference problem by considering applications installed on the user's friend profiles as a potential adversary that can access more than publicly available profile information.
- We introduce the notion of self-sanitization to provide users with control over information leakage from their profiles.

II. Privometer Framework

We show the basic Privometer framework in Fig. 3. The basic building blocks of Privometer are Inference Model, Friends Rank, and Self-Sanitization. The Inference model determines the probability of individual sensitive attribute inference in a user profile based on her friendship relations and friends' attribute values. The information leakage is represented as a combined probability of inference, and the best inference model is chosen based on the maximum leakage value. The Friends Rank component finds the amount of match of sensitive attribute values between the user and friends. The friends are then ranked based on this matching. The self-sanitization component considers that a high ranked friend would cause more leakage than a low ranked friend. It suggests to the user a list of actions to 'sanitize' (lessen) the information leakage in their profiles.

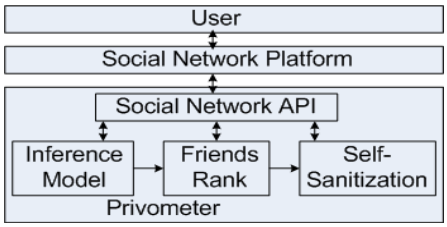


Fig. 3: Basic Framework for Privometer

We developed Privometer as a Facebook platform application using Facebook PHP Client API to fetch information from user profiles. Privometer has two different modes of operation, namely, 'online' and 'offline'. For performance reasons, the 'online' mode takes into consideration only partial friendship relations to limit computation. In 'online' mode, the inference is performed based on the most frequently occurring attribute value in friends' profiles (Fig. 4). The 'offline' mode requires the user to allow Privometer to collect information even when the user is not logged in. The training is performed on this data with various classification techniques. The current implementation considers only neighbor relations (immediate friends) and uses the 'network-only Bayes classifier' [9] to measure the probability of inference (or information leakage) due to friendship links. In this paper, we do not consider profile information beyond immediate friends. The self-sanitization recommendations take into account how much individual influence the friends have on the leakage of user's sensitive attributes. An example of self-sanitization action is to ask a friend to totally hide the matched sensitive attribute. When the user logs in, the measure of information leakage and self-sanitization recommendations are ready for her to view.

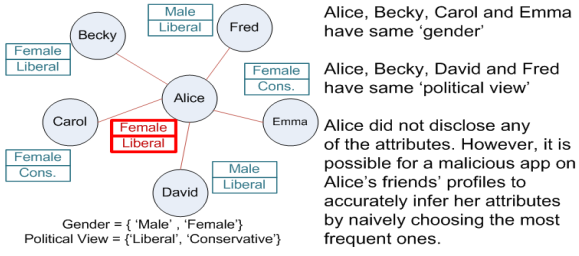


Fig. 4: Inference by Friendship Relations

III. Inference Model

We represent the friendship relations of a Privometer user as star graph[2] as shown in Fig. 2b. We call it a friendship link graph, denoted by $G = (V, E)$. In the graph, $v_0 \in V$ is connected to every other nodes in the graph. An edge $e_j \in E$ represents that v_j is a friend of v_0's. From Fig. 2a, Privometer, which is installed on v_0's profile, has access to v_i's attributes that are either public or made available to friends only. The sensitive attributes can take on a set of possible values; for example, attribute a_k can take values from a finite set $A_k = \{\alpha_{k1}, ...\alpha_{kl}\}$.

We consider, a malicious application has access to all these attribute values except for Alice's ones (shown as bold periphery in Alice's case). Alice, Becky, Carol and Emma have the same value for 'gender', and Alice, Becky, David and Fred have the same value for 'political view'. So, even though Alice did not disclose any of the attributes, it is possible for the malicious application to accurately infer the sensitive attributes from her friendship links.

The information leakage is represented as a combined probability of sensitive attribute inference from the information available in immediate friends' profiles. Fig. 4 shows an example of inference by looking at the most frequent attribute values in friends' profiles. Privometer determines the probability of individual attribute inference first. Then, based on these values and relative sensitivity of attributes, it obtains the combined probability (described later in the section). In Fig. 4, let us consider that the attribute 'political view' is more sensitive than 'gender' and both of them happen to be private in Alice's profile. In that case, David would cause more sensitive information leakage than Carol for Alice.

In the inference model, the sensitive attribute a_k of v_0 is considered a random variable $v_0.\tilde{a_k}$. G captures the friendship relations, and hence the inference is conditioned on G. For a general probabilistic inference model Υ, v_0's inferred attribute:

$$v_0.\tilde{a}_k = \arg_{\alpha_{kt}} \max P_\Upsilon(v_0.a_k = \alpha_{kt}|G)$$

In the Privometer prototype, we implemented the network-only Bayes Classifier inference model (nBC) described by Chakrabarti et. al.[9]. Any other inference model can be easily added to Privometer in a plug and play fashion. Since, the value for $v_0.\tilde{a_k}$ is dependent on $v_i.a_k$'s distribution, where $i \neq 0$, the probability of inference for $v_0.\tilde{a_k}$ in this model is:

$$P_{nBC}(v_0.\tilde{a_k} = \alpha_{kt}|N_0) = \frac{P(N_0|v_0.\tilde{a_k}=\alpha_{kt}).P(v_0.\tilde{a_k}=\alpha_{kt})}{P(N_0)}$$

$$= \frac{P(N_0|v_0.\tilde{a_k}=\alpha_{kt}).P(\alpha_{kt})}{P(N_0)} \text{ where, } 1 < t < |A_k|$$

Let us consider that $v_i.a_k$'s are the sensitive attribute value observed at v_i's. N_0 represents the collection of all known attribute values for a_k of all v_0's friends, i.e. $v_i.a_k$'s, $i \neq 0$. Since, $v_i.a_k$'s are independent of each other, we can assume conditional independence for the term $P(N_0|v_0.\tilde{a_k} = \alpha_{kt})$ and further reduce it.

$$P(N_0|v_0.\tilde{a_k} = \alpha_{kt}) = \frac{1}{Z} \prod_{i=1}^{|N_0|} P(v_i.a_k|v_0.\tilde{a_k} = \alpha_{kt})$$

[2]Facebook platform API is limited to immediate friends' information only

Here, Z is the normalization constant. $P(N_0)$ is the same for all possible $v_i.a_k$ values, and hence considered constant [4][9]. We do not need to compute $P(N_0)$ explicitly since we are normalizing $P(N_0|v_0.\tilde{a_k})$ with Z. The nodes' prior probability, $P(\alpha_{kt})$, is simply the frequency distribution of v_i's for all a_k values (without considering any relations). The term, $P(v_i.a_k|v_0.\tilde{a_k} = \alpha_{kt})$ (called the 'Influence Strength' in He et. al.'s work [1]) describes how v_0 influences its friend $v_i, i \neq 0$ on attribute a_k. The higher the value is, the higher the probability that v_0 and $v_i, i \neq 0$ will have the same value for attribute a_k.

Privometer determines the inference probabilities of all sensitive attributes using known inference models (e.g., relational and collective classifiers [2]). For a model Υ, the inferred attributes are $\{v_0.\tilde{a_k}, 1 \leq k \leq q\}$. Privometer records the success and failure of the inferred attributes as a vector, called attribute matching vector, $\tilde{\psi}$ for model Υ. The k-th component, $\tilde{\psi}^{(k)}$ denotes the success or failure of inference for a_k:

$$\tilde{\psi}^{(k)}_\Upsilon = \begin{cases} 1 & \text{if } v_0.\tilde{a}_k = v_0.a_k \\ 0 & \text{otherwise} \end{cases}$$

The combined probability of inference for model Υ is then:

$$S_\Upsilon = \sum_{k=0}^{q} \omega^{(k)} \wp_\Upsilon \tilde{\psi}^{(k)}_\Upsilon$$

where $\wp_\Upsilon = \max_i P(v_0.\tilde{a}_k = \alpha_{kt}|G)$, $\omega^{(k)}$ is the relative sensitivity vector for attributes a_k, $\omega^{(k)} = [0, 1], 1 \leq k \leq q$, and $\sum_{k=0}^{q} \omega^{(k)} = 1$. And, $S_\Upsilon = [0, 1]$ (is normalized by ω).

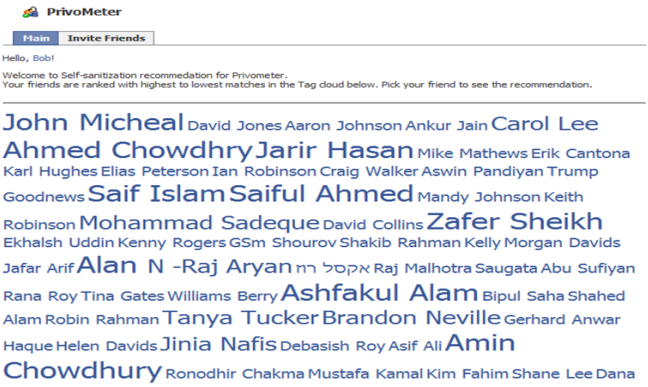
Privometer picks the best inference model based on the maximum combined probability of inference: $\tilde{S_\Upsilon} = max S_\Upsilon$. Self-sanitization recommendations are then provided based on this best chosen inference model.

IV. Self-sanitization Recommendations

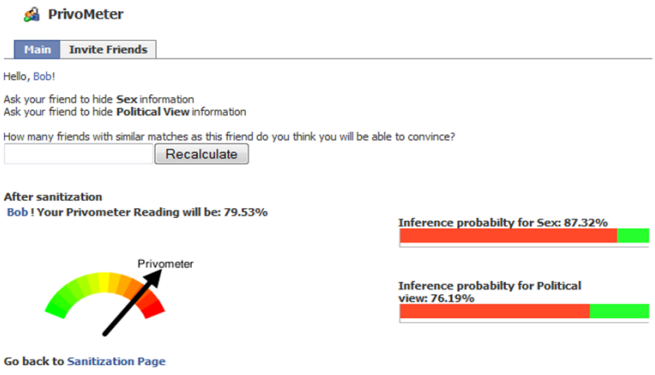
Privometer offers suggestions for effectively controlling the amount of sensitive information leaked from a user profile due to friendship relations. Since the actions to sanitize (lower) the leakage in the user profile are offered as a choice to the user, we call them self-sanitization recommendations. This provides an additional protection that is complementary to user privacy settings. As we mentioned earlier, merely hiding sensitive attributes through privacy settings is not sufficient to protect privacy from malicious applications installed in friends' profiles. For example, in Fig. 2a, even if user v_i has not made a_4 and a_5 publicly available, they can be used by M to infer v_0's a_4 and a_5.

We denote $\tilde{S}^i_\Upsilon$ as v_0's friend v_i's individual contribution ($i \neq 0$) to sensitive attribute inference in the best chosen inference model Υ. We also represent ψ_i as v_i's matching vector that records the matches between v_0 and v_i's attributes. Privometer determines v_i's individual contributions to information leakage as: $\tilde{S}^i_\Upsilon = \sum_{k=0}^{q} \omega^{(k)} \psi^{(k)}_i \tilde{\psi}^{(k)}$.
The weighted version is: $\tilde{S}^i_\Upsilon = \frac{1}{\sum_{i=1}^{|N_0|} \gamma_i} \sum_{k=0}^{q} \gamma_i \omega^{(k)} \psi^{(k)}_i \tilde{\psi}^{(k)}$
where, γ_i is the weight of the link between v_0 and $v_i, i \neq 0$.

The weights can be the number of wall posts exchanged, number of photos tagged, number of mutual friends, etc.



(a) Leakage-based Tag Cloud of Friends



(b) Sanitization Recommendations

Fig. 5: Privometer Friends Rank and Sanitization page

Privometer ranks friends based on individual contributions to information leakage. From the amount of match, Privometer provides the user with a list of actions to help bring down the information leakage. Privometer can also show how the combined probability will be affected if a specific recommendation action is carried out by the user and her friends. An example of self-sanitization actions is requesting a friend to totally hide a sensitive attribute from his profile.

V. Implementation of Privometer

We consider a use case scenario where 'sex' and 'political view' are the user's sensitive attributes. Privometer reading in Fig. 1 shows the amount of combined leakage found for these two attributes from the available attribute values of the friends. We consider 'political view' (weight 0.7) to be more sensitive than 'sex' (0.3). We classify 'political view' as 'liberal', 'conservative' and 'other' based on simple string matching. 'sex' is categorized to 'male', 'female' and 'none' (not specified). The Friends Rank page (Fig. 5a) shows a list of friends in a 'Tag Cloud' visual representation. The relative size of the fonts denotes the relative amount of leakage caused by the friends. Selecting one of the friends from the cloud will take the user to the sanitization action page for a specific friend (Fig. 5b). In this page, the action - 'requesting the friend to hide the matched sensitive attribute from public' is suggested. The Privometer reading shown in this page represents the new amount of leakage if the sanitization action is performed by the user's friend. It is also possible to check the changes in privacy meter reading if a number of friends with the same matches carries out the sanitization action.

VI. Conclusions and Future Directions

In this paper, we described Privometer, a tool that measures privacy leakage in social networks including information obtained by malicious applications installed in the user's friend profiles. Privometer ranks friends based on their individual contributions to privacy leakage and suggest self-sanitization to lessen this leakage accordingly.

The work presented in this paper is only the start for a novel and practical approach to inform users about their privacy in social networks and help them protect it. There are several possible extensions: Privometer is currently assumed to be installed on a single user profile, restricting available information to immediate friends only. If Privometer is installed on some or all the user's friends, we could have access to more information and apply collective classification techniques [10] for more sensitive attribute inference. Furthermore, while implementing Privometer, we only considered friendship links for sensitive attribute inference. To achieve a more accurate inference, we plan to extend the data model to consider group affiliations, page subscriptions, tagging activities, and so on. We could also weight the links in the data model with more information like the number of messages exchanged, mutual friendships, and similar tastes, to denote strong and weak relationships for inference. Finally, self-sanitization actions can be more diverse to include for example unsubscribing from group affiliations, removing applications, and not engaging in some specific activities (e.g., photo tagging) in social network.

References

[1] J. He, W. Chu, and Z. Liu, "Inferring privacy information from social networks," *Intelligence and Security Informatics*, pp. 154–165, 2006.

[2] E. Zheleva and L. Getoor, "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles," in *Proc. of ACM World wide web*, 2009, pp. 531–540.

[3] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *Proc. of ACM SIGCOMM conf. on Internet measurement*, 2007, pp. 29–42.

[4] S. A. Macskassy and F. Provost, "Classification in Networked Data: A Toolkit and a Univariate Case Study," *Journal of Machine Learning Res.*, vol. 8, pp. 935–983, 2007.

[5] J. Neville and D. Jensen, "Leveraging relational autocorrelation with latent group models," in *Proc. of international workshop on Multi-relational mining*, 2005, pp. 49–55.

[6] BBC. (2008) Identity at risk on facebook. [Online]. Available: http://news.bbc.co.uk/2/hi/programmes/click_online/7375772.stm

[7] A. Felt and D. Evans, "Privacy Protection for Social Networking Platforms," *Web 2.0 Security and Privacy*, 2008.

[8] R. Heatherly, M. Kantarcioglu, B. Thuraisingham, and J. Lindamood, "Preventing Private Information Inference Attacks on Social Networks," University of Texas at Dallas, Tech. Rep. UTDCS-03-09, 2009.

[9] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks," in *Proc. of ACM SIGMOD*, 1998, pp. 307–318.

[10] L. Getoor and C. P. Diehl, "Link mining: a survey," *SIGKDD Explor. Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.