Evaluating the Impact of Attacks In Collaborative Tagging Environments

Maryam Ramezani and J. J. Sandvig and Tom Schimoler
and Jonathan Gemmell and Bamshad Mobasher and Robin Burke
Center for Web Intelligence
DePaul University
Chicago, Illinois, USA
{mramezani,jsandvig,tschimoler,jgemmell,mobasher,rburke}@cs.depaul.edu

***Abstract*—The proliferation of social web technologies such as collaborative tagging has led to an increasing awareness of their vulnerability to misuse. Attackers may attempt to distort the system's adaptive behavior by inserting erroneous or misleading annotations, thus altering the way in which information is presented to legitimate users. Prior work on recommender systems has shown that studying different attack types, their properties and their impact, can help identify robust algorithms that make these systems more secure and less vulnerable to manipulation. Unlike traditional recommender systems, a tagging system includes multiple retrieval algorithms to facilitate browsing of resources, users and tags. The challenge is, therefore, evaluating the impact of various types of attacks across different navigation options. In this paper we develop a framework for characterizing attacks against tagging systems. We then propose a methodology for evaluating their global impact based on PageRank. Using real data from a popular tagging systems, we empirically evaluate the effectiveness of several attack types. Our results help us understand how much effort is needed from an attacker to change the behavior of a tagging system and which attack types are more successful against such systems.**

I. Introduction

Collaborative tagging systems such as Delicious, last.FM, and Bibsonomy are valuable components of the Social Web. They allow users, firstly, to organize their own data with a level of freedom not possible in traditional taxonomic filing systems – whether it be web bookmarks, music collections, or academic journal references. Secondly, they provide users a means to openly share this information so that friends and colleagues can easily communicate with each other about their latest discoveries. Finally, they allow anyone to utilize the collective knowledge of others for discovering new resources and perhaps even new friends. These benefits, however, are only as powerful as the system is trustworthy. As with any open adaptive system, maintaining the integrity of a tagging system presents a considerable logistical problem.

There are some who would manipulate the navigational and retrieval mechanisms of the system to further their personal agenda. False profiles, which cannot be readily distinguished from ordinary users, may be injected into the system to bias the network towards or against a particular resource. In a profile injection attack, an attacker uses fictitious identities to give dishonest ratings (either implicitly or explicitly) to a given target in a recommender system. These attacks are typically algorithmically unsophisticated and easily automated [14], [9]. In previous work, we outlined the major issues in building secure recommender systems and proposed approaches to designing robust recommenders and detecting attack profiles [12], [13]. More recently we have extended this work to securing collaborative tagging systems [15], [18]. Tagging systems differ from classic collaborative filtering recommenders in that users generate rich semantics for resources, rather than simply rating them.

Others have also begun to study attacks against collaborative tagging systems. Xu et al. [20] have introduced basic criteria for a good tagging system and proposed a collaborative algorithm for suggesting tags that meet these criteria. They have accounted for spam by assigning a reputation score to each user, based on the quality of the tags contributed by that user. Koutrika et al.[7] have proposed an ideal tagging system where malicious tags and malicious user behaviors are well defined. They propose a trusted moderator who periodically checks if user postings are "reasonable". The moderator also identifies good and bad tags for any resource in the collection. The authors have also defined different strategies of attack, experimenting on the impact of different search algorithms. There is a set of good tags for each resource and if the user does not select from that set to annotate the resource, the tag is classified as spam. This approach, however, does not provide a distinction between a goal-oriented attack or a normal user who might be inclined to select a non-obvious tag for personal reasons. Krause et al. [8] use machine learning approaches to identify spammers in a social bookmarking system. They present features based on the topological, semantic and profile-based information and classify users as spammer or non-spammer. Heymann et al in [2] surveys three categories of potential countermeasures against spam on the social Web based on detection, demotion, and prevention.

Our work, in contrast, has been more concentrated on the practical aspects of how existing systems handle large-scale profile injection attacks. Our first goal is to characterize the various realistic attack scenarios. We believe that studying properties of typical attack strategies will lead to improved detection algorithms and to robust retrieval algorithms. Our second goal, then, is to empirically examine attack strategies using real-world tagging data towards exposing general vul-

nerabilities, eventually detecting and deterring such malicious activity.

In this paper we present a framework for navigating tagging systems and characterize attack types based on those navigation channels. The proposed framework is adapted from our earlier preliminary work [18], in which we defined dimensions of an attack. These dimensions are motivation of attacker, genericity of intended audience, degree of profile obfuscation, size of the attack, navigation context, and the target element. In [15] we described two attack types in detail and studied their impact on the system. In particular, we described the *overload attack* in a tag-resource context (in which a user navigates to resources by selecting a tag) and the *piggyback attack* in a resource-resource context (in which a user selects a resource and is presented with related resources). However, previous techniques for measuring attack effectiveness have been limited. Local metrics are specific to particular retrieval algorithms, and are customized to each attack type. In [15] we proposed hit probability to approximate the probability that a user choosing tags randomly would encounter a given target resource. Although useful, this measure cannot be adapted for other navigation channels or attack types. In this paper we address the need for a metric to evaluate the system-wide impact of attacks.

Our contributions in this paper are two-fold. First, we develop a comprehensive framework that characterizes different type of attacks against tagging systems. Second, we propose a methodology for evaluating and comparing the global impact of attack by evaluating the underlying annotation structure and how it changes over time. We use the adapted PageRank introduced in [4], [5] to study the tagging graph and observe the global change in the graph structure when injecting attack profiles. By performing link analysis on the connections between resources, tags and users, we are able to rank their relative importance outside the context of a particular retrieval algorithm. The global metric provides a means to quantify the system's overall vulnerability. We then empirically show the impact of attacks on two real data sets from Delicious and Bibsonomy.

II. Attacks On Tagging Systems

Collaborative tagging systems facilitate the retrieval and discovery of resources within a user-centric browsing environment. Understanding the methods for attacking a tagging system requires analysis of this navigation process. However, there has been little formalization of tagging system output, and most research treats tagging systems solely as retrieval engines, ignoring the flexible browsing environment such sites offer. There is need for a general model of navigation options and system outputs that can help us study the impact of attack.

A collaborative tagging system consists of three generic elements: users, resources, and tags. Formally, the model can be described as a four-tuple $D = \langle U, R, T, A \rangle$, such that there exists a set of users, U; a set of resources, R; a set of tags, T; and a set of annotations, A. Annotations are represented as a set of triples containing a user, tag and resource such that $A \subseteq \{\langle u, r, t \rangle : u \in U, r \in R, t \in T\}$. The model can be viewed as a tripartite hypergraph $G = (V, E)$, where $V = U \cup R \cup T$ is the set of nodes and $E = \{\{u, r, t\} | \langle u, r, t \rangle \in A\}$ is the set of hyperedges [19]. To simplify analysis, we can reduce the hypergraph into three bipartite graphs with regular edges. The graphs model aggregate associations between users and resources (UR), users and tags (UT), and tags and resources (TR) [11].

		Target Element Type		
		Resource	**Tag**	**User**
Navigation Context	**Resource**	Piggyback	Coattail	Pivot Point
	Tag	Overload	Co-Occurrence	Pivot Point
	User	Mole ("Shill User")	Mole ("Shill User")	

Fig. 1. Navigation Channels and Attack Types

An attacker may attempt to influence a tagging community by manipulating the underlying structure through strategic annotation of resources. The goal is to bias the patterns of organization in a way that is advantageous to the attacker. Although the logistics of mounting such an attack are important, success ultimately depends on generating visibility for the attack within the context of social navigation. Therefore, it is necessary to study a tagging system's means of navigation to determine where vulnerabilities lie.

The framework shown in Figure 1 facilitates a common understanding of navigation options available to disparate tagging systems. Each combination of element types R, U, and T represents a specific *navigation channel* for presenting information [18]. Although a particular tagging system may choose to include only a subset of the possible channels, the system's output should map onto this framework.

A navigation channel has a particular context $r_c \in R$, $u_c \in U$, or $t_c \in T$ that refers to the current location of a user who is browsing or querying the system. A channel also defines a ranking algorithm for the elements $R_c \subseteq R$, $U_c \subseteq U$, or $T_c \subseteq T$ considered relevant to that context. For example, a user selects the tag "coffee" (the tag context) and is presented with a list of resources that are tagged with coffee, ranked by popularity or recency. The user may also retrieve other tags related to coffee, or other users who have employed the tag.

The navigation channel framework is also a natural way to categorize potential avenues of attack. We believe attackers will focus on those navigation channels with the greatest positive impact on their intended outcome. An attack against a tagging system consists of one or more coordinated *attack profiles*. Each profile is associated with a fictitious user identity and contains annotations designed to bias the system. An *attack type* is a strategy for building attack profiles. It allows us to classify common patterns of attack, identifying their aims and tactics. We consider here attacks that have the goal of improving the visibility of a target element. Our overall aim is to identify different attack strategies, study their characteristics, and measure their impact on collaborative

tagging environments. The following attack types are based on the navigation channels shown in Figure 1.

Overload: The goal of an overload attack, as the name implies, is to overload a tag context with a target resource so that the system correlates the tag and resource highly. The assumption is that the attacker wants to associate the target resource with some high-visibility tag, thereby increasing traffic to the target. In some cases, the attacker may attempt to promote the resource to a specific subset of users as a focused version of the overload attack.

Piggyback: The goal of a piggyback attack is for a target resource to ride the success of another resource. It exploits the idea of sharing tags among resources and attempting to associate the target resource with some resource context, such that they appear similar. There are two possible implementations of piggyback. The *tag duplication* technique is to pick a number of tags highly correlated to the resource context and annotate the target resource with the same tags, preferably with the same distribution. The *tag overlap* tactic is to pick any number of random tags and annotate both the resource context and the target resource with those tags within the same attack profile.

Co-Occurrence: The goal of co-occurrence is for a target tag to be correlated with another popular or focused tag. An attack consists of annotating any resource with both tags, such that they always occur together. Tagging systems that measure the similarity between tags may increase the rank of the target with respect to the tag context and a user viewing the tag context will have a high chance of seeing the target in the list of related tags. There are two possible implementations of co-occurrence. The *resource duplication* technique is to pick a number of resources highly correlated to the tag context and annotate each resource with the target tag. The *resource overlap* technique is to pick any number of random resources and annotate them with the tag context and the target tag within each attack profile.

Coattail: The goal of coattail attack is for a target tag to be correlated with a particular resource context. The resource context may be popular or focused, depending if the intended audience is generic or specific, respectively. An attack is created by annotating the resource context with the target tag in every attack profile.

Mole: The goal of mole (or "shill user") is to create profiles intended to build trust within a targeted audience. The audience may be general, or more likely, a focused user segment. Over time, the attack profiles annotate resources relevant to the targeted audience in such a way as to mimic a domain expert. At some point after the attack profile has established trust, the intended target resource or tag is injected into the profile, hoping that other users in the segment will simply assume it is also relevant to them.

Pivot Point: The goal of pivot point is to create a strong association between an attack profile and its intended audience by correlating it with resources and/or tags that are relevant to the targeted user segment. A mole attack may utilize a pivot point in order to establish the attack profile as an expert in the particular user segment. The defining characteristic of a pivot point attack is an indirect link to the actual target element – the attacker wants to raise the visibility of the attack profile itself, which in turn contains the target resource or tag.

III. Evaluating Impact of Attacks

We are interested in evaluating both the local impact of an attack on a single navigation channel as well as capturing the overall global impact of an attack. By studying localized tagging system output, as defined by a navigation channel, we can see how an end user of the system is actually affected by attack. By studying the global changes to the underlying annotation structure, we are able to gain insight into how attacks propagate through the tagging system.

A. Measuring the Local Impact of Attack

From the attacker's perspective, an attack is successful if it generates the desired visibility for the targeted element within the intended navigation channel. Therefore, it is necessary to have localized metrics showing how end users that are browsing the channel are affected by the attack. We can track these channel-specific effects by looking at the rank of the target item before and after the attack.

Although the average rank treats differences at the top and bottom of the list identically, from the attacker's point of view a difference between a rank of 10 and a rank of 20 is far more significant than the difference between a rank of 110 and 120. For this reason, we measure the difference in reciprocal rank before and after attack. Let r be the rank of the target item before the attack and r' be the rank afterwards. Rank improvement is given by $Imp = \frac{1}{r'} - \frac{1}{r}$ and is relative to the navigation channel, making the local metric specific to the implemented retrieval algorithm.

The retrieval algorithm of a navigation channel selects and ranks the elements considered relevant to that context. Relevance may be displayed in different ways between contexts, such as "popular tags", "recent tags", "recent resources", "active users", "related tags", etc. Generally, results are ranked by popularity or recency, but there is no limitation.

While research on retrieval models in tagging systems has introduced new algorithms in [5], [10] , we focus on the vector space model [17] adapted from the information retrieval discipline to work with social tagging systems. We assume that retrieval is based on the Tag-Resource channel using the reduced TR bipartite graph; however, this framework may be easily modified to support retrieval in any navigation channel by using an appropriately defined bipartite graph.

A resource can be represented as a tag vector $\vec{r} = [w_{t1}, w_{t2}, \cdots, w_{tn}]$ such that w_t is the weight of a particular tag $t \in T$. Vector weights may be derived through many methods, including frequency or recency. The *tag frequency*, *tf*, for a tag, $t \in T$, and a resource, $r \in R$ is the number of users who have annotated the resource using the tag. We

define *tf* as: *tf(t,r)* $= |\{\langle u, r, t\rangle \in A : u \in U\}|$. When a single tag is used as a query, as we assume here for simplicity, the resources are listed in order of their tf values because using $tf.idf$ will result in the same ranked list.

Three navigation channels are symmetric in nature – the Resource-Resource, Tag-Tag and User-User channels. Each has navigation context and target belonging to the same element type, and the outputs of these symmetric channels are based on similarity. For example, Delicious shows a list of tags that are related to a particular tag. In the same way, it is possible to list similar resources and users. In our experiments we use cosine similarity to find related resources and tags. This allows us to evaluate the local effects of Piggyback and Co-Occurrence attacks, respectively.

B. Measuring the Global Impact of Attack

Although local metrics based on navigation channel output are useful for evaluating how end users are affected by attack, they are limited because they only measure attack effectiveness relative to specific navigation channels and retrieval algorithms. A global metric that observes changes in the underlying network of annotations allows us to study attack effectiveness in a more holistic manner. Annotations belonging to an attack create links between the target element and other contextual elements. This propagates to subsequent nodes via other annotations, and so on. Because there are any number of paths to the attacked element that are beyond the attacker's control, there may be unexpected effects of an attack across navigation channels.

Hotho et al have introduced two algorithms, Adapted PageRank and FolkRank, that generate a global ranking of tagging system elements [4], [5]. The algorithms are variations of PageRank and score all elements by exploiting the underlying graph structure. They have used the PageRank-based approach in the context of retrieval and recommendation; in contrast we are proposing the use of Adapted PageRank for evaluating the global impact of attack. We believe PageRank can accurately measure global impact because it introduces a notion of page authority that is independent of content and based solely on the graph structure. In the context of tagging systems, a resource that is annotated by authoritative users with authoritative tags can be considered to be authoritative itself. An attack exploits this model because it links the target element to contextual elements and creates a mutual reinforcement of authority that is measurable by PageRank.

To study the overall effectiveness of an attack we measure the Adapted PageRank of an attacked item, then calculate the improvement in a similar manner as defined in section III-A. In this case, the rank of the target item refers to the relative rank of the element based on its PageRank score across all elements. We spread the weight using the following equation:

$$\vec{x} = dW\vec{x} + (1-d)\vec{v} \tag{1}$$

To calculate the weighted matrix, notice that G' contains three types of edges $E' = E_{u,r} \cup E_{u,t} \cup E_{r,t}$. The weight of each edge u, r is $|\{\langle u, r, t\rangle \in A : t \in T\}|$, each edge u, t is $|\{\langle u, r, t\rangle \in A : r \in R\}|$, and each edge r, t is $|\{\langle u, r, t\rangle \in A : u \in U\}|$. The damping factor d determines the influence of $\vec{v}$, which is typically defined as $v = [1, \cdots, 1]^T$to make the system randomly jump to another link from time to time but it may be personalized with user preferences. In all our experiments except the focused attacks we consider d as 1 since we are not interested in the random jumps of the system. In the focused attacks, the preference vector is set to target specific group of users. To ensure that the transition matrix is column stochastic, we then divide each weight by the total number of annotations the column participates in.

IV. Experimental Evaluation

In this section we present results showing the impact of several different attack types based on section II. For each attack type, we generate a number of attack profiles and insert them into the system database, testing the effects of different attack sizes and number of selected tag contexts.

A. Data Description

Our analysis is performed on two separate tagging systems: Delicious and Bibsonomy. Delicious is a popular bookmarking service boasting a very large user base. Bibsonomy focuses on bookmarks as well as scientific publications [3]. Since tagging systems often contain a great deal of noise in the data, a p-core [1] of the data sets were taken such that each user, resource and tag appear in at least p annotations where an annotation is defined as a user, resource and all tags that user applied to that resource. The density of the data is also increased, as a portion of the long tail is removed.

The Delicious data set was gathered from 10/20/2008 to 12/15/2008. The most popular tags in the system were collected. For each of these popular tags the most recent users and their neighbors are collected. Full user histories were then collected. Due to memory constraints, 20% of the users was randomly selected. Finally, a 20-core was taken in order to reduce noise and increase density. The final dataset contains 7664 users, 15612 resources and 5746 tags.

The Bibsonomy dataset is available online by the system administrators. Again a p-core was taken to reduce noise and increase density. However, since the Bibsonomy data set is far smaller and less dense than that offered by Delicious it cannot support a 20-core. Instead a 5-core was selected. The final dataset contains 401 users, 2014 resources and 1754 tags.

One of our goals is to determine whether tagging distribution of the target object influences attack effectiveness. It has been observed that the distribution of the number of users who tagged a URL follows a power law, in which a relatively small number of URLs are tagged with high frequency while all the rest occur with low frequency. Since we remove the long- tail part of the data by applying p-core, we divide the remaining part into two partitions and run experiments on each partition independently to explore if the different parts of the distribution show different behaviors.

We use the coefficient of variation ($CV = stdev/mean$) to determine the partition boundaries as described in [16].In

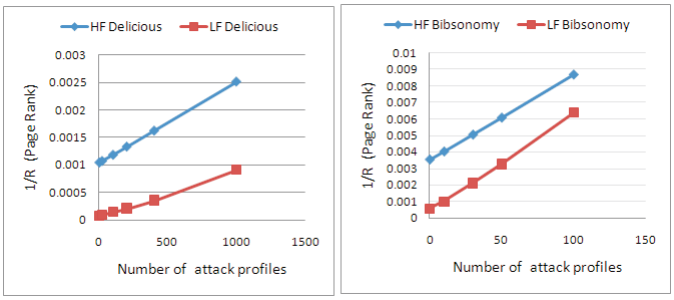

Fig. 2. Global impact of popular overload attack with one popular tag in each attack profile

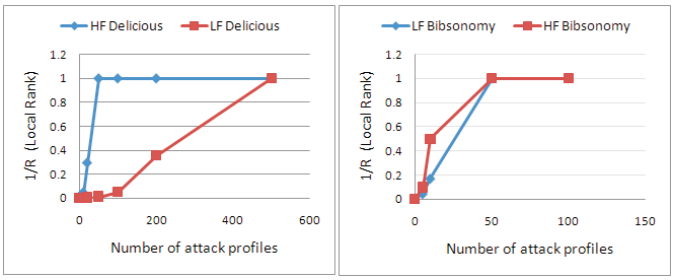

Fig. 3. Local impact of popular overload attack with one popular tag in each attack profile

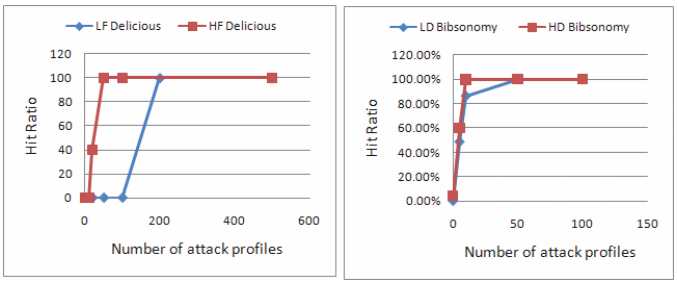

Fig. 4. Hit Ratio for popular overload attack with one popular tag in each attack profile

Bibsonomy data resources with frequency of 20 and higher are located in the high frequency part and the cut-off for high frequency resources in Delicious is 224.

We show the impact of each attack type using the local and global metrics. For measuring the global effect we look at the change in the PageRank of the target item in the overall network. So, the global rank includes all users, tags, and resources. The local measures include hit ratio and change in the local rank in each related channel using a particular retrieval algorithm. Note that the global rank is independent of the navigation channel or the retrieval algorithm while the local rank is based on retrieval algorithm in each navigation channel.

B. Overload Attack

As mentioned in Section II, the goal of an overload attack is to associate a set of tags with a target resource so that the system retrieves the target resource when one of the tags is given as a query. We implement two variants of this attack: *popular* and *focused*.

1) Popular Overload Attack: We have two sets of experiments for popular overload attack. In both cases, the target resource is randomly selected from either the high or low frequency partition. In the first, we choose a single tag at random from the 50 most popular tags in the system, testing different numbers of attack profiles. Figure 2 shows the global impact of this type of attack for both data sets. We can see the same trend in both data sets, i.e., that increasing the number of attack profiles increases the reciprocal rank for the target resource. As expected, the high frequency resources have higher reciprocal rank compared to the low frequency resources before the attack; however, we see that the rate of change in reciprocal rank is slightly proportional to the initial value; highly ranked items rise faster with increased attacks than lower ranked items. The LF partition in Bibsonomy has a higher rate of increase compared to LF in delicious which suggests that in a smaller tagging system, it is easier for low frequency resources to get higher ranks compared to a popular system such as Delicious. Same results can be found in local rank improvements that show it needs much more effort to get high ranks for resources located in the low frequency part of the data in Delicious,

Figure 3 shows the local impact of attack based on the retrieval algorithm (tf, in this case). We can see that the impact of the attack is already reflected in the local rank after only small attacks. Figure 4 shows the impact of the attack from the point-of-view of an actual user querying the system; we show the hit ratio of the target for a query consisting of a single popular tag and considering the top 10 results.

For our second round of experiments, we keep the attack size constant and vary the number of popular tags included in each profile. Figure 5 depicts the global effect of varying the profile size from 1 to 10 tags. Our PageRank metric allows us to directly compare the effect of attack size vs. profile size. We see that adding more popular tags to each attack profile can be as effective as adding more attack profiles. For example, at attack size 50, associating 2 popular tags to the target resource has more or less the same impact as adding 100 attack profiles with 1 popular tag each. Note that it is easier and less costly for an attacker to associate more popular tags to target resource per profile than it is to add more attack profiles.

2) Focused Overload Attack: In a focused attack, the attacker is targeting a particular group of users. Such an attack might contain profiles that associate the target URL with tags related to a specific tag. To measure the global effect of this attack, we bias the preference vector toward the focused tag to see the impact on the system for users who are interested in that specific tag. We use the approach taken in [6] to set the weights for the preference vector. We give higher weights to the focused tag in $\vec{v}$ in equation 1. While each user, resource, and tag gets an initial preference weight of 1, the focused tag gets an additional preference weight, $1+|T|$, where $|T|$ is the number of unique tags in the data. We set parameter $d = .85$.

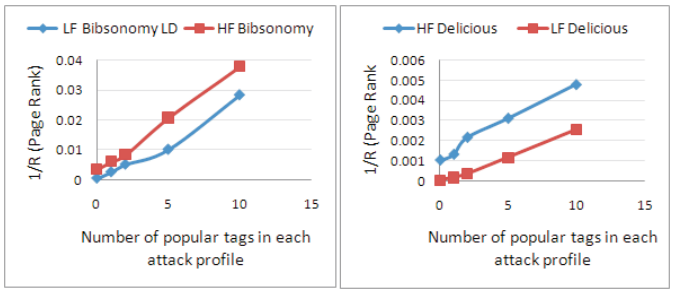

Fig. 5. Global impact of popular overload attack with n popular tag in each attack profile with 50 attack profiles for Bibsonomy and 200 for Delicious

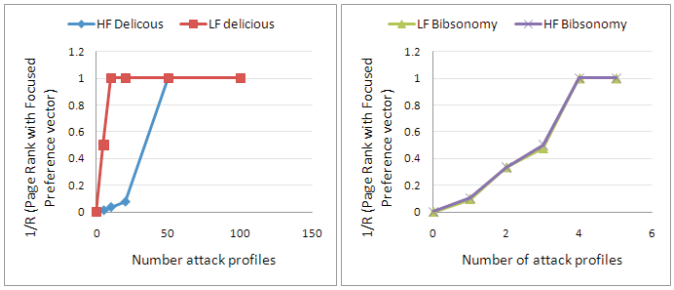

Fig. 6. Global impact of focused overload attack

We use random tags from the low density partition as our focused tags.

The results from both datasets show that the focused overload attack can drastically change the behavior of the system for the target users. As shown in Figure 6, even very small attacks can change the rank of the target resource to 1. The hit ratio and rank change using the tf retrieval algorithm show similar results, which implies that focused attacks can have an extreme effect on the behavior of the system for a specific target group. Since the drastic impact of this attack is clearly shown using PageRank, we have not shown the local measures such as hit ratio and local rank due to space limitations.

In Figure 7, we show an example of the impact of a focused attack. In this example our focused tag which is randomly selected from the low frequency partition is "projektmanagement". We can see that before the attack the resource "http://openproj.org/" is the most highly associated resource with this tag. The right side of the table shows the ranking after the attack. The target resource in this attack is "http://www.lambdaprobe.org/d/index.htm" and we can see that not only does the target resource get the highest rank with respect to the focused tag, but also the other resources and tags which are highly associated with the focused tag will change. Tags "monitoring" and "tomcat" happen to get higher rank because they are the tags which are associated with the target resource. As shown in this example, using PageRank can help us understand the impact of attacks in the overall network which was not possible using previous approaches such as hit ratio or hit probability.

Before the attack	After Attack
tag : projektmanagement	tag: projektmanagement
http://openproj.org/openproj	tareget resource: http://www.lambdaprobe.org/d/index.htm
user: 329440	tag : tools
tag : tools	tag:monitoring
user: 244700	tag: software
http://www.backpackit.com/	tag: design
user: 205480	tag: java
http://openproj.org/	tag:tomcat
tag: software	tag:web

Fig. 7. An example of change in the network with focused attack for focused tag " projektmanagement"

C. Piggyback Attack

To evaluate the Piggyback Attack we implement two variations of the basic strategy as discussed above: in the *tag duplication* technique, for each "user" a certain number of tags are chosen from the popular resource's profile and applied to the target; in the *tag overlap* technique, both the target and the popular resource are tagged a predetermined number of times using tags selected at random.

We perform our experiments on the delicious dataset, using targets from the high and low frequency groups. In addition to inverse rank based on PageRank, we also measure cosine similarity between the target resource and the chosen popular resource. For each experiment, we vary only the number of fake profiles: the number of annotations per attack profile is set at 6; which we found to give optimal effect for attacks of this kind[15]. Ten independent runs of each experiment are performed – two targets are chosen at random for each of the five most popular resources in the dataset – and the average taken.

As shown in figure 9, when choosing targets from the low frequency set, we see a divergence of the inverse rank for the two techniques. The *tag overlap* strategy is notably more effective at raising the target's PageRank (admittedly only at the severest levels of attack). One explanation is that in the case of *tag duplicate*, the attack has the side effect of reinforcing the already-high authorities of the tags in the popular resource's profile. The target only receives a small portion of this increased authority in back-propagation; the popular items' "outflow" is diluted among their many neighbors. When random tags are used, a more exclusive connection is generated between the popular resource and the target. This is not a direct connection, of course; it has to flow through various tag and user nodes, but these nodes have a highly concentrated flow back to the target, since they generally do not have many connections elsewhere.

For targets in the high-frequency group (see figure 8), there is a much smaller penalty for choosing popular tags. The target resource is already well-connected, so its authority grows at the same pace as the popular resource and its tags. Still, we see that *tag overlap* does perform best at the highest levels of attack. There certainly seems to be a benefit to injecting "novel" annotations into the network.

An interesting comparison is made when looking at the localized metric of cosine. *tag duplicate* causes the cosine between target and popular resource to jump dramatically after

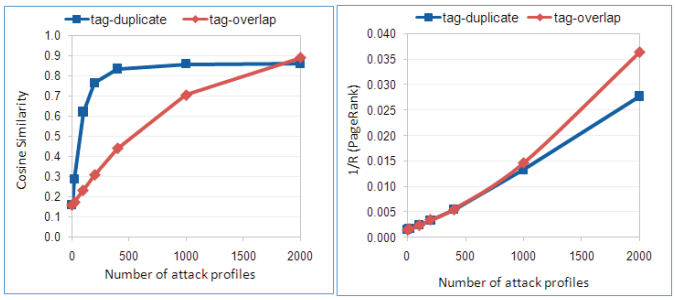

Fig. 8. Local (cosine, left) and global (PageRank, right) impact for two variants of the Piggyback attack, with a target from the high frequency partition

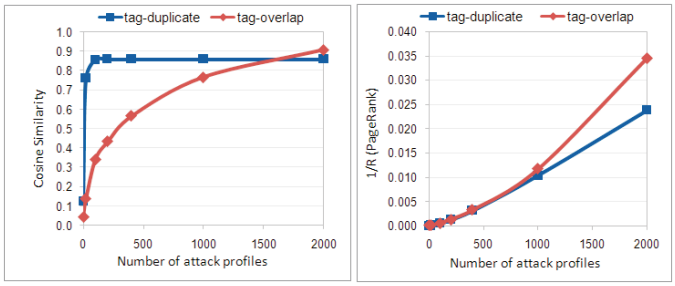

Fig. 9. Local (cosine, left) and global (PageRank, right) impact for two variants of the Piggyback attack, with a target from the low frequency partition

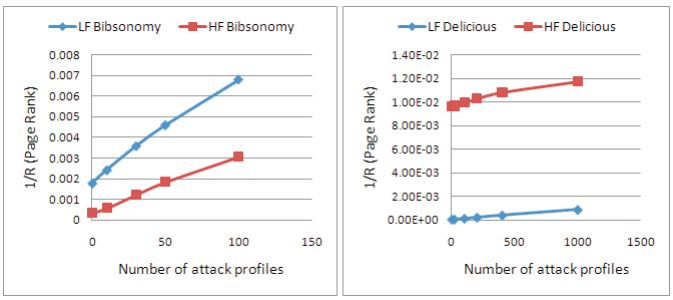

Fig. 10. Global impact of Co-occurrence attack with 1 popular tag and 1 resource in each attack profile

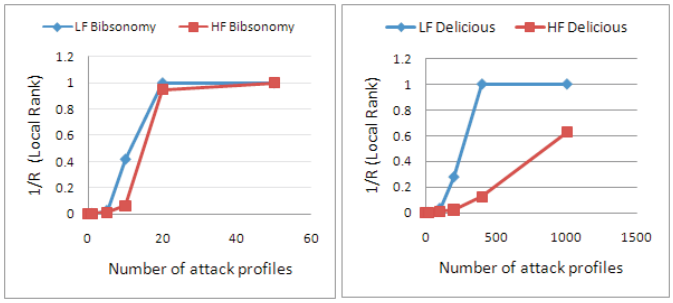

Fig. 11. Local impact of Co-occurrence attack with 1 popular tag and 1 resource in each attack profile

only a small attack, yet it levels out and never surpasses a certain upper bound. Meanwhile, *tag overlap* requires a greater effort, but appears to grow boundlessly, subject to the size of the attack.

D. Co-Occurrence Attack (Tag Push)

The goal of co-occurrence is for a target tag to be correlated with another, more popular tag. An attack consists of annotating any resource with both tags, such that they always occur together. The Co-occurrence attack profile consists of n popular tags, n resources – randomly selected from the high density partition – and the target tag. Basically, this approach is similar to the *tag overlap* variation of the PiggyBack attack, with the difference that in this case the target is a tag and the goal is to associate it with popular tags. We have three sets of experiments for this attack. In all experiments the target tag is selected at random from either partition of the data. Our first experiment varies the number of attack profiles. An attack profile consists of one popular tag, one random resource, and the target tag. Figure 10 shows the change in the overall rank of the target tag. We see from the results that this attack is not as successful for the delicious dataset as it is for bibsonomy. This may be due to the fact that the popular tags in delicious are well established and it is not easy for an attacker to generate a high rank for some random tag in the overall network. However, the local results reported in Figure 11 show that this attack can be effective in the context of "related tags". We use cosine similarity to find similar tags to the popular tag. Figure 11 shows that even small attacks can change the related tag rank in favor of the target tag. In bibsonomy, for example, adding 10 attack profiles makes the target tag most similar to the popular tag.

Our next experiment is to change the number of popular tags in each profile while keeping the number of resources and attack size fixed. In this experiment, each attack profile consists of n popular tags, one random resource and the target tag. The left side of Figure 12 shows the results for this type of attack. As we can see, there is no correlation between the number of popular tags and the rank of the target tag. This result is to be expected since changing the number of popular tags in one attack profile will not help the target tag get a higher rank, but it will help the associated resource to get a higher rank as the target tag occurs only once in each attack profile. Our third experiment changes the number of resources to be tagged. Thus, each attack profile consists of one popular tag, n resources, and the target tag. The results for this experiment can be seen in the right side of Figure 12. We see that the number of resources can effect considerably the rank of the target tag. This means that one attacker can associate a target tag with a popular tag over many resources and easily get a high rank for the target tag.

E. Summary of the Results

Our results show that one of the most effective attacks is the focused attack, particularly because it does not require a huge effort by an attacker to mount and because it would be difficult to detect. Attackers can easily change the global and local behavior of the system for a target group by adding sufficient number of attack profiles. The results from popular overload attack show that in a large and popular system such

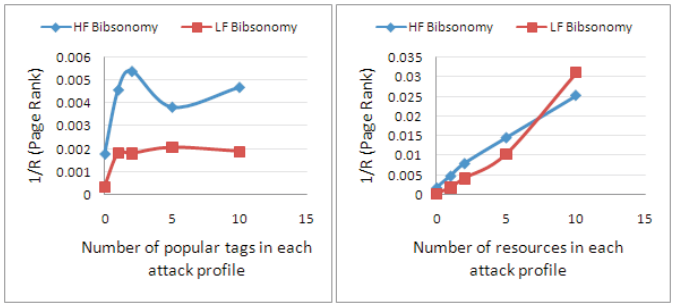

Fig. 12. Global impact of Co-occurrence attack by changing the number of popular tags (left chart)/resources (right chart) with 50 attack profiles

as Delicious an attacker would require a huge effort to push an unpopular resource by associating it with only one popular tag. However, by associating the target resource with a group of popular tags, the rank of the target resource can be raised significantly. Our results from the piggyback attack shows that tag duplicate technique is more successful in changing the local behavior of the system (i.e., with respect to the specific retrieval algorithm) while tag overlap approach is more successful in changing the global rank in large attacks. Results from co-occurrence attack show that it doesn't need a huge effort from an attacker to change the local behavior of the system. However, in a large system like Delicious in which the popular tags are well established, it is almost impossible for an attacker to have a global impact. So, while attackers can easily associate the target tag with a popular tag, they cannot enhance the position a the target tag into the the group of most popular tags.

V. Conclusions and Future Work

In this paper, we introduced an effective measurement, based on PageRank, to evaluate the global impact of attacks against social tagging systems. We modeled three attack types: Overload (popular and focused), Piggyback, and Co-occurrence, experimenting with two real-world datasets. Our results show that tagging systems are quite vulnerable to attack. The results concerning focused overload attacks showed that a goal-oriented attack which targets a specific user group can be easily injected into the system. While producing this attack does not require a great deal of effort or knowledge from an attacker's perspective, it may be more difficult to detect this kind of attack since it resembles the natural behavior of a person who is interested in a specific topic. In our future work, we will model other attack types and compare their impacts on the system. We plan to introduce the notion of "recency" into our experiments, since this is a commonly used factor in commercial social tagging systems. Understanding the attacks and their effects on social tagging systems will help us discover more robust tagging systems and also guide us to find algorithms for detection and prevention of attacks.

References

[1] V. Batagelj and M. Zaveršnik, "Generalized cores," *Arxiv preprint cs/0202039*, 2002.

[2] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting spam on social web sites: A survey of approaches and future challenges," *IEEE Internet Computing*, vol. 11, no. 6, pp. 36–45, 2007.

[3] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "Bibsonomy: A social bookmark and publication sharing system," in *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, 2006, to appear.

[4] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "Folkrank: A ranking algorithm for folksonomies," in *Proc. FGIR 2006*, 2006. [Online]. Available: http://www.kde.cs.uni-kassel.de/stumme/papers/2006/hotho2006folkrank.pdf

[5] ——, "Information retrieval in folksonomies: Search and ranking," in *The Semantic Web: Research and Applications*, 2006, pp. 411–426.

[6] R. Jäschke, L. B. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme, "Tag recommendations in folksonomies," in *Knowledge Discovery in Databases: PKDD 2007*, ser. Lecture Notes in Computer Science, vol. 4702. Berlin, Heidelberg: Springer, 2007, pp. 506–514.

[7] G. Koutrika, F. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina, "Combating spam in tagging systems," *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pp. 57–64, 2007.

[8] B. Krause, A. Hotho, and G. Stumme, "The anti-social tagger- detecting spam in social bookmarking systems," in *Proc. of the Fourth International Workshop on Adversarial Information Retrieval on the Web*, 2008.

[9] S. Lam and J. Reidl, "Shilling recommender systems for fun and profit," in *Proceedings of the 13th International WWW Conference*, New York, May 2004, pp. 393–402. [Online]. Available: http://www2004.org/proceedings/docs/1p393.pdf

[10] A. Lui, "Web information retrieval in collaborative tagging systems," in *Proceedings of the WI 2006. IEEE/WIC/ACM International Conference on Web Intelligence*, 18-22 Dec 2006, pp. 352 – 355.

[11] P. Mika, "Ontologies are us: A unified model of social networks and semantics," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 1, pp. 5–15, 2007.

[12] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Towards trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Transactions on Internet Technology*, vol. 7, no. 4, p. 23, 2007.

[13] B. Mobasher, R. Burke, R. Bhaumik, and J. J. Sandvig, "Attacks and remedies in collaborative recommendation," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 56–63, May/June 2007.

[14] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre, "Collaborative recommendation: A robustness analysis," *ACM Transactions on Internet Technology*, vol. 4, no. 4, pp. 344–377, 2004. [Online]. Available: http://portal.acm.org/citation.cfm?id=1031114.1031116

[15] M. Ramezani, J. J. Sandvig, R. Bhaumik, R. Burke, and B. Mobasher, "Exploring the impact of profile injection attacks in social tagging systems," in *Proceedings of WebKDD*, 2008.

[16] A. Riska, V. Diev, and E. Smirni, "Efficient fitting of long-tailed data sets into hyperexponential distributions," in *IEEE Globecom Conference, Internet Performance Symposium, Taipei, Taiwan, November 2002. IEEE Catalog Number: 02CH3798C*, 2002.

[17] G. Salton, A. Wong, and C. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[18] J. Sandvig, R. Bhaumik, M. Ramezani, R. Burke, and B. Mobasher, "A framework for the analysis of attacks against social tagging systems," in *In Proceedings of The 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, Held in conjunction with the 23nd National Conference on Artificial Intelligence - AAAI 2008 , July 13-17, 2008 - Chicago, Illinois, USA*, 2008.

[19] C. Schmitz, A. Hotho, R. Jaschke, and G. Stumme, "Mining association rules in folksonomies," *Data Science and Classification: Proceedings of the 10th IFCS Conference, Ljubljana, Slovenia, July*, 2006.

[20] Z. Xu, Y. Fu, J. Mao, and D. Su, "Towards the semantic web: Collaborative tag suggestions," in *Collaborative Web Tagging Workshop in conjunction with the 15th WWW Conference*, Edinburgh, Scotland, May 2006.