A Framework for the Analysis of Attacks Against Social Tagging Systems*

J.J. Sandvig, Runa Bhaumik, Maryam Ramezani, Robin Burke, Bamshad Mobasher

Center for Web Intelligence
School of Computing, DePaul University
Chicago, Illinois, USA
{jsandvig,mramezani,rbhaumik,rburke,mobasher}@cti.depaul.edu

Abstract

Social tagging systems provide an open platform for users to share and annotate their resources such as photos and URLs. Due to their open nature, however, these systems present a security problem. Malicious users may try to distort the system's behavior by inserting erroneous or misleading annotations, thus altering the way in which information is presented to legitimate users. This paper addresses the problem of modeling attacks against social tagging systems and evaluating their impact on the systems' behavior. Gaining a fundamental understanding of the nature and impact of such attacks will hopefully lead to more secure and robust social Web applications. We present the dimensions that characterize an attack and outline a framework to model the attacks based on various navigation channels and target elements. Using our framework we classify and identify different types of potential attack strategies against a social tagging system. We implement two of our attack models and evaluate their impact on retrieval algorithms commonly used by tagging systems.

Introduction

Social tagging systems have become popular tools for organizing content. A tagging system allows users to annotate resources with one or more personalized labels, also known as "tags". The primary benefit for a user is the ability to classify information in a natural way. There is typically no limit to the number of tags that may be assigned to a resource and there is no strict hierarchy of tags. This freedom from predefined navigational and conceptual hierarchies has resulted in tagging systems being described as "folksonomies".

Many different tagging systems are available, each specializing in a particular type of resource. Some popular examples include del.icio.us[1] and Flickr[2]. Del.icio.us is a Web site that allows users to bookmark URLs and view them from any connection. Flickr allows users to upload, share and manage pictures. Other applications specialize in music, blogs, or journal publications.

Although users often tag resources for personal benefit, the emerging patterns of organization can contribute to the common good (Golder and Huberman 2006). As a result, tags can be used to enhance social navigation. Social tagging systems allow users to peruse other users' personal tags. It is commonly possible to browse another user's tagged resources directly, or to browse all resources with the same tag.

Recent work has established that adaptive Web applications, such as collaborative recommender systems, can be manipulated via "profile injection attacks". In a profile injection attack (sometimes called "shilling"), an attacker uses fictitious identities to insert biased implicit or explicit ratings into a recommender system (Burke et al. 2005). Such profiles may be generated manually by an attacker or an automated agent. These attacks do not require a great deal of knowledge about the details of the recommender system or its algorithms (O'Mahony et al. 2004; Lam and Reidl 2004; Burke, Mobasher, and Bhaumik 2005; Mobasher et al. 2005).

Tagging systems are also dependent on public input, and are therefore susceptible to profile injection attacks. Attackers may use misleading tags to confuse others or to achieve some goal, such as promoting a product or brand. A real-world example of a spam attack can be seen in Figure 1, where a user has succeeded in promoting his profile to three of the top bookmarks in the del.icio.us site. Spamming has also forced Spurl.net [3] to disable certain functionality. In addition, Ma.gnolia [4] has noted that over a 3 month period, twice as many spam bookmarks were created as legitimate bookmarks.

For further illustration, consider the example shown in Figure 2 of a tagging system that allows users to annotate URLs. A subset of tag assignments are displayed for users (User1 - User6). Suppose a user is searching for the resource that is most related to the tag "coffee". Prior to attack, the system will display the resource "Starbucks" based on the number of occurrences of the tag. Now suppose another coffee shop, Jonbucks, wishes to promote the resource "Jonbucks" to a segment of users interested in coffee. Attack profiles (Attack1 - Attack3) are created, assigning the tag "coffee" to "Jonbucks". After the attack, the system now displays Jonbucks as the most related resource to coffee based

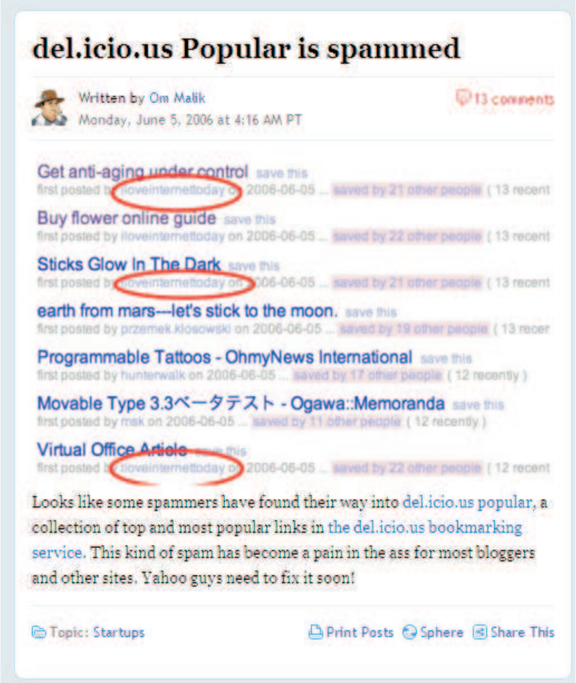
*This work was supported in part by the National Science Foundation Cyber Trust program under Grant IIS-0430303.

[1] del.icio.us

[2] www.flickr.com

[3] www.spurl.net

[4] ma.gnolia.com



del.icio.us Popular is spammed

Written by Om Malik
Monday, June 5, 2006 at 4:16 AM PT

13 comments

Get anti-aging under control save this
first posted by iloveinternettoday on 2006-06-05 ... saved by 21 other people (13 recent

Buy flower online guide save this
first posted by iloveinternettoday on 2006-06-05 ... saved by 22 other people (13 recent

Sticks Glow In The Dark save this
first posted by iloveinternettoday on 2006-06-05 ... saved by 21 other people (13 recent

earth from mars---let's stick to the moon. save this
first posted by przemek.klosowski on 2006-06-05 ... saved by 19 other people (13 recer

Programmable Tattoos - OhmyNews International save this
first posted by hunterwalk on 2006-06-05 ... saved by 17 other people (12 recently)

Movable Type 3.3ベータテスト - Ogawa::Memoranda save this
first posted by msk on 2006-06-05 ... saved by 11 other people (12 recently)

Virtual Office Article save this
first posted by iloveinternettoday on 2006-06-05 ... saved by 22 other people (12 recent

Looks like some spammers have found their way into del.icio.us popular, a collection of top and most popular links in the del.icio.us bookmarking service. This kind of spam has become a pain in the ass for most bloggers and other sites. Yahoo guys need to fix it soon!

Figure 1: The spam on Del.icio.us page

on number of occurrences. Although Starbucks is still a reasonable result for the system to display, the attack profiles have created a bias toward Jonbucks.

The primary contribution of this paper is a framework for the analysis of attacks against social tagging systems. We first present a set of attack dimensions that establish a context for our analysis. We next discuss six attack types, based on a navigation channels and attack targets within a tagging system. These attack types represent in abstract the strategies that could be employed by attackers in order to manipulate the output of the system. Finally, we present some experimental results on two attack types on popular tags and on popular resources, quantifying the system's vulnerabilities using several proposed evaluation metrics.

Related Work

Recently, collaborative tagging has exploded as a trend in information systems to manage online resources. Users benefit from social tagging systems in several ways. The user may manage a collection of resources for later retrieval or discover resources tagged by other users. Furthermore, social tagging helps users not only to identify interesting resources, but also interesting groups of users.

Mika (Mika 2007) has modeled the networks of folksonomies at an abstract level, representing such systems as a tripartite graph with hyperedges. The set of vertices is partitioned into three (possible empty) disjoint sets corresponding to set of actors (users), set of concepts (tags) and the set of annotated objects (resources). In a social tagging system users tag objects with concepts, creating ternary associations between the user, concept and the object.

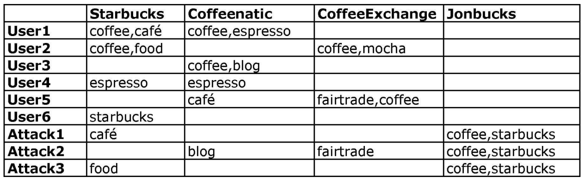
As tagging systems are becoming more popular, researchers have started to explain and characterize the tagging phenomenon (Macgregor and McCulloch 2006; Golder and Huberman 2006). The most significant formal study of tagging systems appeared in the work of Golder and Heberman (Golder and Huberman 2006). The authors studied the information dynamics in "collaborative tagging systems" specifically, the del.icio.us system. The authors discussed how tags have been used by individual users over time and how tags for an individual resource stabilizes over time. They also discussed two semantic difficulties: polysemy (when a single word has multiple related meanings) and synonymy (when different words have the same meaning) of tagging systems. Macgregor and McCulloh provide an overview of the phenomenon and explore reasons why both social tagging as well as ontologies will have a place in the future of information access (Macgregor and McCulloch 2006). From a system's perspective, Sen et al. studied how personal tendencies and community influences affect the way users tag items on a movie recommender Web site (Sen et al. 2006).

Chi and Mytkoswicz (Chi and Mytkowicz 2007) have analyzed del.icio.us and found that the efficiency of social tagging decreases as the communities grow; that is, tags are becoming less and less descriptive and consequently it becomes harder to find a particular item using them. Simultaneously, it becomes harder to find tags that efficiently mark an item for future retrieval. These results indicate that it is very important to take into account user attention in terms of observed tagging activity. Niwa et al. (Niwa, Doi, and Honiden 2006) have proposed a recommendation system based on the affinity between users and tags, and on the explicit site preferences expressed by the user.

Hotho et al. (Hotho et al. 2006) have introduced two algorithms for ranking search results in folksonomies. The first algorithm "Adapted PageRank", a modification of PageRank (Kleinberg 1999), has been proposed to provide a ranking scheme in folksonomies. This modified algorithm provides one global ranking. The basic notion is that a resource tagged with important tags by important users becomes important itself. The same holds true, symmetrically, for tags and users. However, their experimental results showed that adapted PageRank did not work very well. The other proposed ranking algorithm is the FolkRank, which computes a topic-specific ranking. Their results showed that FolkRank performs better than Adapted PageRank.

We have previously studied profile injection attacks against collaborative filtering recommenders (Mobasher et al. 2007). We examined the effects of attack types that require varying degrees of knowledge about the system, and proposed several responses to attack. The first was to develop more robust alternatives to standard collaborative filtering algorithms. We demonstrated that a number of model-based and hybrid algorithms offer substantial improvement over standard algorithms. Another proposed response is to detect and defeat attackers before they cause harm. We used a supervised classification approach to identify and respond to profile injection attacks.

Researchers have begun to study attacks on social tagging



	Starbucks	Coffeenatic	CoffeeExchange	Jonbucks
User1	coffee,café	coffee,espresso		
User2	coffee,food		coffee,mocha	
User3		coffee,blog		
User4	espresso	espresso		
User5		café	fairtrade,coffee	
User6	starbucks			
Attack1	café			coffee,starbucks
Attack2		blog	fairtrade	coffee,starbucks
Attack3	food			coffee,starbucks

Figure 2: An hypothetical example of promoting a resource.

systems. Xu et al. (Xu et al. 2006) 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. Reputation scores have been used for identifying good candidate tags for a particular document, i.e., for automatic tag selection.

Koutrika et al. 2007 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.

Heymann et al. surveyed three categories of potential countermeasures: those based on detection, demotion, and prevention (Heymann, Koutrika, and Garcia-Molina 2007). Although many of these countermeasures have previously been proposed for email and Web spam, the authors found that their applicability to social Web sites differs.

Social Tagging Systems

In the broadest sense, folksonomies consist of three generic elements: users, resources, and tags. The relationships between the elements and their evolution over time defines the social tagging space. A social tagging system provides the supporting infrastructure that allows users to annotate resources in the system.

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 $A \subseteq \{\langle u, r, t \rangle : u \in U, r \in R, t \in T\}$.

A tagging system 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 (Schmitz et al. 2006). This tripartite graph is complicated and difficult to understand. However, we can reduce such a hypergraph into three bipartite graphs with regular edges. These three graphs model the association between users and resources (UR), users and tags (UT), and tags and resources (TR) (Mika 2007). For example, the bipartite graph TR links tags to resources, and each link is weighted by the number of times users annotated that resource with that tag.

A tagging model provides an explicit structure for codifying tacit knowledge possessed by a system’s user community. Individual users assign personal meaning to resources via tags. Collectively, the relative proportion of unique tags assigned to a resource tend to stabilize over time, indicating both imitation and shared knowledge within the community (Golder and Huberman 2006).

An attacker may attempt to influence a tagging community by manipulating the underlying structure through strategic annotation of resources. Although the logistics of mounting such an attack are important, success ultimately depends on generating visibility for the attack target. Therefore, it is also necessary to study the means of navigating a social tagging site to determine where vulnerabilities lie. In the following sections, we introduce the concept of navigation channels and then describe common retrieval algorithms used within the channels.

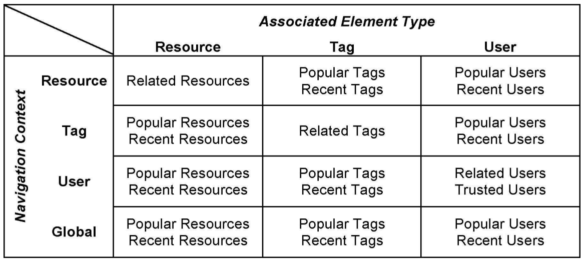
Navigation Channels

The success of collaborative tagging is partially due to facilitating the retrieval and discovery of resources within a single user-centric environment. Many tagging systems publicly display each user’s tags and resources, making retrieval of previous annotations both simple and intuitive. However, the discovery process is much more complex. Users browse the social tagging graph via the many associations between resources, tags, and users. This ability to navigate through the folksonomy is one reason for the popularity of collaborative tagging.

Understanding the avenues for attacking a social tagging system requires analysis of its navigation process. However, there has been little formalization of tagging system outputs, and much research treats tagging systems solely as retrieval engines, ignoring the flexible browsing environment such sites offer. There is a need therefore for a general model of navigation options and system outputs that can help us model the impact that an attacker may have.

It is beneficial to distinguish the roles of interaction between the annotation and navigation processes. In particular, annotation is concerned with a contributor to the tagging system, whereas navigation is concerned with the viewer. There is no requirement that the viewer of a tagging system is also a contributor. Although it is often the case that contributors annotate resources for their own consumption, most tagging systems also allow unregistered visitors to browse. For example, users of del.icio.us typically annotate their bookmarks for personal consumption, but anyone can browse the site.

Each combination of element types R, U, and T represents a specific navigation channel for presenting information in a tagging site. The context of a channel is a reference point for retrieving associated elements. In particular, it is the specific $r \in R$, $t \in T$, or $u \in U$ that serves as a query. Many tagging systems will also include a global context, with no specific query, that facilitates exploration of the site. Given a context, the system will return a set of associated elements of a specified type that are relevant to the context.



		Associated Element Type		
		Resource	Tag	User
Navigation Context	Resource	Related Resources	Popular Tags Recent Tags	Popular Users Recent Users
	Tag	Popular Resources Recent Resources	Related Tags	Popular Users Recent Users
	User	Popular Resources Recent Resources	Popular Tags Recent Tags	Related Users Trusted Users
	Global	Popular Resources Recent Resources	Popular Tags Recent Tags	Popular Users Recent Users

Figure 3: Navigation Channels of a Tagging System

As an illustration, consider the Tag-Resource channel. Conceptually, we consider the Tag-Resource channel from an information retrieval perspective. Viewing the reduced bipartite graph TR as a corpus, we map R and T to documents and terms, respectively. The channel is represented as a single-term query, such that the tag t_q is the user's current tag context. The query returns the most relevant resources $R_t \subset R$ that have been annotated with t_q.

Other navigation channels can be specified in a similar manner, as shown in Figure 3. A tagging system may choose to include only a subset of the possible channels: for example, del.icio.us does not have a "Related resource" function. The information that is displayed, however, will fall into one of the channels described here. This model allows a common analysis of different systems.

Retrieval Algorithms

Within each navigation channel, a retrieval algorithm defines the particular elements considered relevant to the 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 based on popularity or recency, but there is no limitation. Some applications may also allow the viewer to choose the appropriate ranking algorithm. For example, del.icio.us allows a user to view the most popular or most recent resources that are annotated with the specified tag.

While other retrieval models may be used, our work focuses on the vector space model (Salton, Wong, and Yang 1975) adapted from the information retrieval discipline to work with social tagging systems. The following equations assume retrieval is based on the Tag-Resource channel using the reduced TR bipartite graph; however, they may be easily modified to support retrieval in any navigation channel by using an appropriately defined bipartite graph.

A resource vector is represented as $\vec{r} = [w_{t1}, w_{t2}, \cdots, w_{tn}]$ such that w_t is the weight of a particular tag $t \in T$. Vector weights may be derived by many methods, including frequency or recency. In this work, we will rely on frequency. The *tag frequency*, *tf*, for a tag, $t \in T$, and a resource, $r \in R$ is the number of times the resource has been annotated with the tag. We define *tf* as:

$$\textit{tf(t,r)} = |\{a = \langle u, r, t\rangle \in A : u \in U\}| \qquad (1)$$

Likewise, the well known *term frequency * inverse document frequency* (Salton and Buckley 1988) can be modified for social tagging systems. The *tf*idf* multiplies the aforementioned frequency by the importance of the tag t. The importance is measured by the log of the total number of resources, *N*, divided by the number of resources to which the tag was applied, n_t. We define *tf*idf* as:

$$\textit{tf*idf(t,r)} = \textit{tf(t,r)} * \log(N/n_t) \qquad (2)$$

With either term weighting, a similarity measure between a query, q, represented as a vector of tags, and a resource, *r*, can be calculated. We use Cosine as similarity measure to retrieve similar resources to a particular resource. In this case, each resouce is represented as a vector of tags.

Attacks Against Tagging Systems

An attack against a social tagging system consists of one or more coordinated attack profiles. Each profile is associated with a fictitious user identity and contains annotations intended to bias the system. Our overall aim is to identify different types of attacks, study their characteristics, and measure their impact on social tagging systems. We first present attack dimensions that are relevant to analysis. Next, we introduce several specific attack types and discuss possible strategies an attacker may choose for implementing them.

Attack Dimensions

In this section, we present seven dimensions of an attack against a social tagging systems. Specifically, we discuss motivation of the attacker, intent of the attacker, genericity of the intended audience, degree of profile obfuscation, size of attack, navigation context, and target element. We believe that studying properties of typical attack strategies can lead to improved attack detection algorithms and to more robust retrieval algorithms.

Motivation of Attacker At a basic level, an attacker may be motivated to either disrupt the tagging system as a whole, or to promote a particular viewpoint within the system. In the first case, an "eBully" may attempt to introduce random noise into the system, simply to promote anarchy or to degrade the reputation of the system. Although certainly a concern, it is difficult to quantify an attack motivated by disruption because of the subjective decision about when an outlier is considered true noise and when it is considered an attack.

Our primary focus is on the attacker interested in promoting a particular viewpoint. Presumably, the attacker wants to bias the system in order to produce some economic or political advantage. Furthermore, the viewpoint may include a short-term or long-term purpose. For example, a political activist or special interest group may have a short-term goal of influencing a particular vote, or a long-term goal of promoting some larger issue. Likewise, a firm may attempt to manipulate a market in the short-term for economic gain or have a long-term goal of promoting a particular product or brand.

Intent of Attacker If motivation describes the "why" of an attack, then an attacker's intent describes the "what". It is the desired outcome of a particular attack campaign. The tagging system may be the direct target of attack, or it may be used indirectly to influence the actual target of attack.

In a direct attack, the intent may be to promote a particular product within the tagging system itself, or to demote a competitor's product. We call these "push" and "nuke" attacks, respectively. In an indirect attack, the intent is to use the tagging system platform in order to bias some other system. For example, an attacker may use a social bookmarking system to create a large number of back-links to some target URL, in an attempt to raise its Google PageRank value.

Intended Audience It may not always benefit an attacker to throw the widest possible net. Instead, an attack is likely to be aimed at those users of the system that are most receptive to the overall intent. For example, in Figure 2, "Jonbucks" coffee shop is attempting to promote its Web site on a social bookmarking system. The company's goal might be to improve its ranking with respect to those users that are interested in coffee, a targeted-marketing strategy.

The genericity of a targeted user segment may be different, depending on the context of the attack. The intended audience may range from universal to focused. An attack on a completely generic user segment is analogous to finding the lowest common denominator within the entire user community – attempting to promote a product to the most common and popular interests.

As an illustrative example of the difference between universal and focused attacks, look again at an attack to promote Jonbucks coffee shop. To target all users, Jonbucks would annotate its site with the most popular tags in the entire tagging system, regardless of their relevance: "design" and "blog" are the most popular tags on del.icio.us at the moment. For targeting a coffee-focused user segment, Jonbucks would use tags such as "coffee" and "mocha", which are likely to be of employed by those users. For our purposes, we will consider an attack that uses the most popular tags to be a general attack. An attack using any other set of tags is assumed to be a focused attack directed towards the users who tend to employ those tags.

Degree of Profile Obfuscation Depending on the intent, an attacker may obfuscate the injected user profiles to help mask the attack. In an extreme example, great care may be taken to ensure that an attack profile looks exactly like a real user profile. The attacker tries to mimic an expert in the domain of the targeted user segment, building trust until the attack is carried out.

On the other end of the spectrum, the attacker doesn't care if the profile looks legitimate at all, and focuses only on maximum effect in biasing the system's retrieval algorithms. The degree of profile obfuscation is a tradeoff, as greater obfuscation is more difficult for the system to detect, but is more labor intensive to build and takes longer for the attacker to see returns.

Size of Attack The size of attack measures the number of coordinated attack profiles that are added to the tagging system. The minimum number of profiles required for an attacker to obtain the desired effect is largely influenced by the overall goal of the attack. If the goal is to mimic a domain expert, the attack may be successful by using only one or two carefully constructed user profiles.

However, if the goal is to bias the system's retrieval algorithms, a large number of attack profiles may be necessary in order to bias the aggregate ranking of the attack target, relative to related elements. In this case, the popularity of related elements has a large effect on the point of accelerating returns.

As an illustration, look again at the Jonbucks attack on the tag "coffee". If there are very few bookmarks that are tagged with coffee, then relatively few attack profiles need to be created that annotate Jonbucks with coffee. However, if "Starbucks" has already been tagged with coffee over 100,000 times, then Jonbucks has a much larger hurdle to clear, requiring a very large number of attack profiles to surpass the popularity of Starbucks.

Navigation Context Navigation context refers to a specific resource, tag, or user in the tagging system that provides a mechanism for navigating its associated elements. It is the current location of a viewer who is browsing or querying the system. An attacker may focus on a particular navigation context as the reference point of attack. In the Jonbucks example, the tag "coffee" is the navigation context, and the attacker wants to improve the rank of the Jonbucks Web site within that context.

An attack may include multiple navigation contexts (e.g., Jonbucks might utilize both "coffee" and "mocha" tags). However, for the purposes of this paper we will focus on attack using a single context. This does not mean, however, that an attack aimed at a single context will only impact one aspect of the tagging system. In the Jonbucks example, attacking the "coffee" tag context may have the unintended result of making Jonbucks and Starbucks very similar resources. If the tagging system includes a navigation channel for displaying similar resources, someone viewing the Starbucks resource may then see Jonbucks ranked highly.

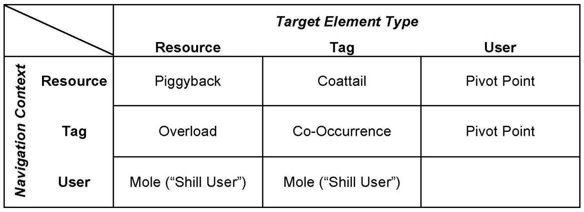
Target Element Target element refers to the specific resource, tag, or user in the tagging system that is the actual target of attack. It is the element that the attacker wishes to promote. In many cases, this is likely to be a resource. In the Jonbucks example, the attacker wants to improve the visibility of the Jonbucks Web site.

However, the target element could also be a tag or user. An attacker may want to push the tag "Jonbucks", simply to raise brand awareness. The tag could be associated to the tag "coffee" such that Jonbucks is advertised as related to coffee, or the tag could be annotated to the resource "Starbucks" as an alternative brand. Similarly, an attacker may want to push a personal user profile as a form of self-promotion.

Attack Types

An attack type is a strategy for building attack profiles. Studying attack types allows us to classify common patterns of attack and identify their aims and tactics. Our categorization of attack types is based on the navigation channels shown in Figure 3. Figure 4 summarizes the types.

An attack type is a generic strategy for building attack profiles. It is a partial model based on abstract navigation context and target element types. A particular implementa-



		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”)	

Figure 4: Summary of Attack Types

tion of an attack type includes specific details, and should be analyzed according to the attack dimensions introduced in previous section. However, studying generic attack types allows us to classify common patterns of attacks at a strategic level. We now propose a number of attack types that correspond to the different navigation channels within a social tagging system. A summary of attack types is shown in Figure 4.

Overload [Context: tag. Target: resource] 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 resource. If the intended audience of the attack is general, a popular tag is chosen. If the intended audience is specific, a focused tag is chosen that is particular to the targeted user segment.

Piggyback [Context: resource. Target: resource] 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, attempting to associate the target resource with some resource context, such that they appear similar. The resource context may be popular or focused, depending if the intended audience is generic or specific.

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.

Coattail [Context: resource. Target: tag] The goal of coattail 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.

For example, an attack can associate the tag “Jonbucks” to the resource “Starbucks”. By creating multiple attack profiles, the Jonbucks tag may be pushed to the top of the list of popular tags for Starbucks, making it highly visible to users looking for tags associated with Starbucks.

Co-Occurrence [Context: tag. Target: tag] 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. The assumption is that the attacker wants the target tag to show up as a ”related tag” to the tag context. Tagging systems that measure the similarity between tags may increase the rank of the target with respect to the tag context. A user that views 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, preferably with the same distribution as the tag context. The *resource overlap* technique is to pick any number of random resources and annotate them with both the tag context and the target tag within each attack profile.

Mole [Context: user. Target: resource or tag] 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 [Context: resource or tag. Target: user] 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. The user segment may be generic or focused, which determines the choice of resources and tags in the attack profile.

A mole attack may utilize a pivot point in order to establish the attack profile as an expert in the particular user segment. However, pivot point may be used in any general scenario where attack profiles are meant to be highly visible, with the hope that the profiles will receive more traffic. 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.

Experiments

Data Description

Our analysis is performed using data collected from the del.icio.us bookmarking service. We collected the data from the Web site using an HTML crawler. We have a complete profile of about 29000 users which contains all of the their tags and bookmarks.

To find the initial users, we started from the del.icio.us popular feed (http://del.icio.us/popular). First we collected all the users who have used the tag “design”, which is the most popular tag in the del.icio.us system. For each of these users, we downloaded their RSS feed containing their most recent postings. We extracted all the tags from all of the postings in the dataset. Then we downloaded the RSS feed for each tag, which contained the most recent postings using that tag. From the dataset at that point, we extracted all users. Then for each user, we downloaded their complete history using an HTML spider. The final dataset consists of complete histories for all of these users (29,918 users). We didn't use any of the previous intermediate datasets in the final dataset; so it consists only of postings by those 29918 users, and there is a complete posting history for each user at the time of our crawling in April 2007. Our final dataset contains 29,918 users; 6,403,441 unique URLs; 1,035,177 tags; 13,222,166 (User, URL) Pairs; and 47,185,789 (User, URL, Tag) Triples.

Data Partitioning

One of our goals is to identify whether tagging distribution of the target object influences attack effectiveness. It has been observed that the probability 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.

The most popular URL has frequency of 14,353, while 99% of the URLs have frequency less than 77 and 50% of the URLs have frequency of less than 3. These numbers show that large portions of the data are in the long tail. Removing the long tail means ignoring a huge part of the data, so we decided to use all of the data in our experiments. However, since different part of the distribution may show different behaviors, we divide the data to different partitions and we experiment on each partition independently of other partitions, comparing the results from each partition.

Our approach in partitioning is to ensure that each partition exhibits significantly reduced variability in comparison to the variability of the entire data set. We use the coefficient of variation (CV) to determine the partition boundaries. CV is a statistical measure of the dispersion of data points in a data series around the mean. It can be written as $CV = stdev/mean$ and is a useful statistic for comparing the degree of variation from one data series to another, even if the means are drastically different from each other.

```
Input: Frequency distribution of objects in ascending order
Output: $P_i$ , $i$ partitions
FREQ = $\{F_1, ..., F_n\}$, Frequency scores
$CV_{MAX} = \delta$, threshold
$i = 1$
$bin = \emptyset$
foreach $F_j \in FREQ$ do
    $bin = bin \cup C_j$
    Calculate $AVG = avg(bin)$;
    Calculate $STD = std(bin)$;
    Calculate $CV = \frac{STD}{AVG}$;
    if $CV > CV_{MAX}$ then
        $P_i = bin$;
        $i = i + 1$;
        $bin = \emptyset$ ;
    end
end
```

Algorithm 1: Partitioning Algorithm

We followed the procedure described in Algorithm 1 to partition URLs based on their frequency distribution. Using 107 as the threshold results in three partitions, which we have used throughout each experiment. Partition 1 contains low frequency URLs tagged less than 1450 times, Partition 2 contains medium frequency URLs with a tag count between 1450 and 4850, and Partition 3 contains high frequent URLs tagged more than 4850 times.

Evaluation Metrics

In each experiment, we wish to measure the effectiveness of an attack. There are a number of possible evaluation metrics to measure the desired outcome for the attacker. In an attack scenario, the attacker may desire that the target element is more likely to be encountered after the attack than before. Commonly, a user navigates through a tagging site by clicking on a tag and retrieving a ranked list of associated resources to that tag. However, as we have seen there are a variety of navigation channels in a tagging system. A user might click on a tag, a resource, or another user to change the current context. Within each context, related tags, resources, or users are ranked based on some criteria. Since there are several possible kinds of output and the ranking criteria is different for each kind, we suggest that evaluation metrics will be different for each type of output. We describe three such metrics that are a focus in this paper.

Hit Probability Hit probability is an extention to Hit Ratio to estimates the probability of a page being visited by a random user navigating through the Web site. The probability that a user clicks on a specific tag can be estimated as the ratio of the frequency of that tag to the sum of the frequencies of all tags in the system. Therefore, we have a probability for each tag represented by $P(t_i)$ computed as follows:

$$P(t_i) = \frac{tf(t_i)}{\sum\limits_{j=1}^{N} tf(t_j)} \qquad (3)$$

We use Tag Frequency, *tf*, as the retrieval algorithm and find the average likelihood that the target resource is in the top n results of a tag context for a random user navigating through the system. To this end, we use top n hit ratio such that a hit value is 1 if the target resource appears in top n results, and 0 otherwise. For each tag context t_q we multiply $P(t_q)$ by the hit ratio. Thus hit probability for each resource is $P(t_q)$ if the resource appears in the top n list, and 0 otherwise. In our experiments, we use the average hit probability for all target resources in each partition of the data. More formally, the hit probability of resource r_i given top n results $R_t \subset R$ of a tag t_q is:

$$HitProb(r_i, t_q) = \begin{cases} P(t_q) & \text{If } r_i \in R_t \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

This metric is appropriate for evaluating attacks intended to have a general impact, when the behavior of the average user is the target.

Rank Improvement Our "push" attack model is designed in such a way that the search technique should improve the position of the resource, placing it at a higher ranking.

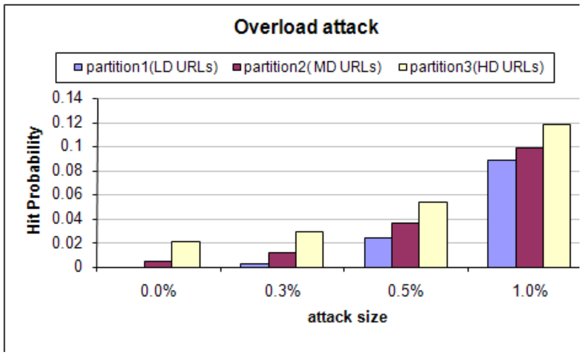


Figure 5: Overload attack hit probability for varying attack sizes, 50 selected popular tags, within top-20 list

The difference in the inverse of the ranks before and after can be used to judge the improvement in rank gained by attack. The rank improvement metric can be written as:

$$Imp = \frac{1}{rank_{after}} - \frac{1}{rank_{before}} \tag{5}$$

The average rank improvement can then be calculated as the sum of the rank improvements for all target elements divided by the total number of target elements. In our experiments, attack types are designed to increase the rank of a target and the average rank improvement will always be positive. This metric is appropriate for a focused attack that seeks to impact a particular channel.

Similarity Depending on navigation channel, the output of a system can be based on the similarity between resources, tags or users. For example, del.icio.us shows related tags to a particular tag. In the same way, it is possible to have related or similar resources to a particular resource. This output can be based on a similarity measure that identifies similar tags or resources to a particular tag or resource. In our experiments we use cosine similarity to find similar resources. Such a metric can measure the effectiveness of a piggyback or similar attack.

Experimental Results

In this section we present preliminary results showing the impact of two types of discussed attacks. In particular, we model the Overload and Piggyback attacks and use the evaluation metrics described in the preceding section to test attack effectiveness. 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 .

Overload Attack In this experiment, we implement the Overload attack by adding fake profiles to the system that associate the target resource with popular tags. We use a set of 50 most frequently used (popular) tags from our database and we test the attack effectiveness in the three different distributions of resources described in data partitioning section. We randomly select 10 resources from each partition and average the results over the resources.

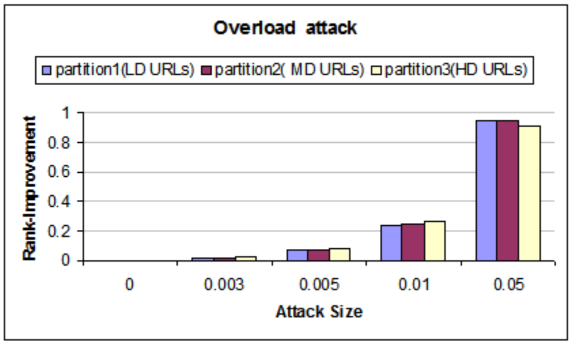


Figure 6: Overload attack rank improvement for varying attack sizes, 50 selected popular tags

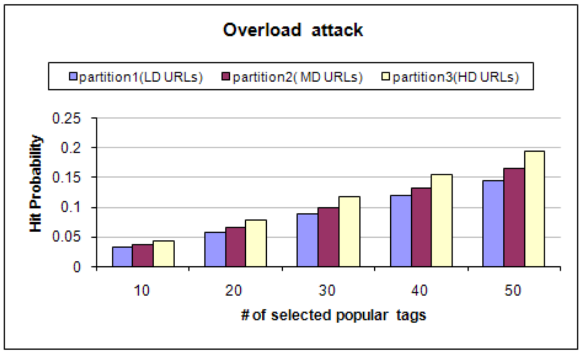


Figure 7: Overload attack hit probability for varying # of popular tags, at 1% attack size, within top-20 list

We use the hit probability and rank improvement measures to evaluate the attack impact when using *tf* as the retrieval algorithm. We do not take into acount recency in our retrieval algorithm , so our retrieval is only based on popularity of resources for each tag. We look at the impact of attack by changing two variables: size of attack and number of popular tags that are associated to the target resource. We measure “size of attack” as a percentage of the actual users in the system. There are approximately 29,000 users in the database, so an attack size of 1% corresponds to 290 attack profiles added to the system.

Figure 5 depicts the effect of varying attack sizes (percentage of bad users in the system) when 50 popular tags are associated to the target resource. Note that each attack profile contains the target resource associated with all the 50 selected popular tags. The result indicates that hit probability values before an attack are very low for Partition 2 and Partition 3, and zero for Partition 1. This behavior is expected, as the chance that low frequency URLs show up in search results for popular tags is very small before an attack. However, after attack the hit probability increases steadily as the number of malicious users increases for all three par-

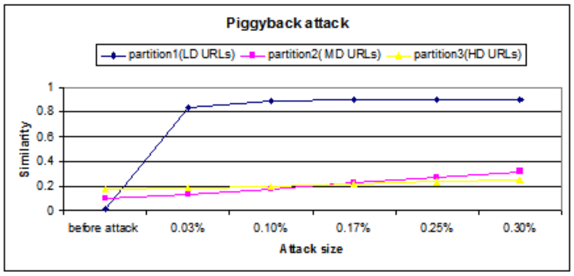


Figure 8: Piggyback attack similarity between popular URLs and target URL for varying attack sizes, 6 top tags duplicated

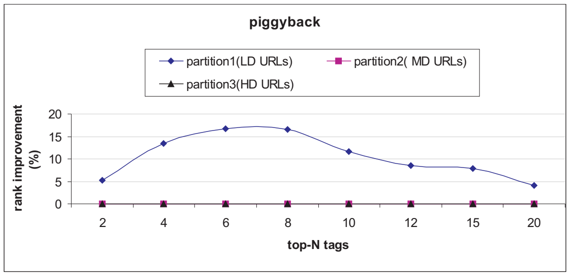


Figure 9: Piggyback attack rank improvement between popular URLs and target URL for varying duplicated tag numbers, 50 users added

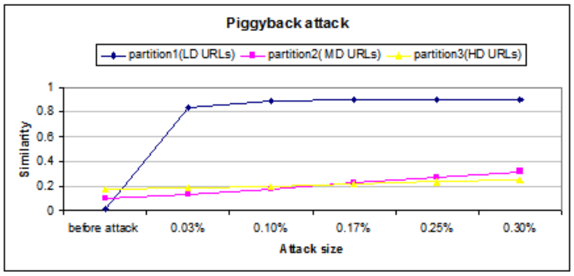


Figure 10: Piggyback attack rank improvement between popular URL and target URL for varying attack size, with top 6 tags duplicated

titions. Low frequency URLs (Partition 1) are more vulnerable to attack than other URLs. Figure 6 shows the rank improvement of the target resource for varying attack sizes.

Figure 7 illustrates the effect of varying the number of selected popular tags with 1% attack size. The result indicates that, as we assign the target resource with more popular tags, the chance of being in top-20 list becomes higher. Note that we select random resources from each partition, so the results may not be identical using the same parameters in different experiments.

Piggyback Attack

We implemented the Piggyback attack using the tag duplication strategy. The goal of this attack is to promote the target resource as a similar resource to one or more selected resources. The selected resources are randomly selected from popular resources. We pick a set of tags which are the most frequently used tags for the selected popular resource, and we associate those tags to the target resource.

To see the impact of the attack we measure the cosine similarity between selected popular resources and the target resource. We have selected 5 popular resources and 10 target resources from each category. Our results include average similarity over selected popular resources and all target resources in each partition. In our experiments, we consider each resource as a vector of tags, which stores the frequency of users who have associated each tag to that resource.

We look at the impact of attack by changing two variables: attack size and number of selected tags from popular resources that are associated to the target resource. Attack size is the number of users added to the system. We look at impact of attack for each different partition of the distribution.

Figure 8 displays similarity when changing the number of duplicate tags. As the results show, the similarity between popular resources and low frequency URLs changes dramatically after the attack. The reason is that the number of tags associated with low frequency URLs are very small before attack. Adding 50 profiles that duplicate tags from selected popular resources will have a large effect in the similarity between the target resource and selected resource. However, similarity score doesn't change much for other partitions before and after an attack, as these resources are already associated with many common tags prior to attack.

Figure 9 shows the average rank improvement between the target resource and popular resources. This result indicates that for low frequent URLs (Partition 1), as similarity increases the rank also increases. After adding the top 6 tags from popular resources, the rank of the target resource is at position 5 and results in a 16% rank improvement. For other partitions, similarity score doesn't change and there is no improvement in the ranking.

Figure 10 shows the results for similarity with varying attack size. It indicates that even small number of attack users (10) added to the system with 6 top tags selected from popular resources, exposes vulnerability in low frequency URLs. The similarity score changes from .014 before attack to .85 after attack. As expected, the similarity score for the other partitions didn't change much before and after attack. Similar results were observed in rank improvement (not shown here).

Conclusions and Future Work

In this paper, we discussed the problem of security and robustness in social tagging systems. We introduced a framework to model the navigation channels in social tagging systems and we identified different types of potential attacks against the system through different navigation channels. We modeled two attack types, Overload and Piggyback, and experimented using a real dataset. Our results from the piggy back attack show that the low density resources are mostly vulnerable while the high density resources are insensitive to to this kind of attack. However, the resutls from the overload attack show that mostly all parts of the distribution are vulnerable to this kind of attack and the rank improvemnet and hit probability results show that an attacker can make considerable changes in the system by inserting 1% attack profiles to the system. In future work, we will model other attack types and compare their impact on the system. We plan to investigate additional metrics for measuring the impact of an attack, including global measures that more accurately measure the relative prominence of nodes in the tagging network. We are also interested to discover algorithms for detection and prevention of attacks.

References

Burke, R.; Mobasher, B.; Zabicki, R.; and Bhaumik, R. 2005. Identifying attack models for secure recommendation. In *Beyond Personalization: A Workshop on the Next Generation of Recommender Systems*.

Burke, R.; Mobasher, B.; and Bhaumik, R. 2005. Limited knowledge shilling attacks in collaborative filtering systems. In *Proceedings of the 3rd IJCAI Workshop in Intelligent Techniques for Personalization*.

Chi, E., and Mytkowicz, T. 2007. Understanding navigability of social tagging systems. In *Proceedings of CHI*, volume 7.

Golder, S., and Huberman, B. A. 2006. The structure of collaborative tagging systems. *Journal of Information Science* 32(2):198–208.

Heymann, P.; Koutrika, G.; and Garcia-Molina, H. 2007. Fighting spam on social web sites: A survey of approaches and future challenges. In *IEEE Internet Computing, vol. 11, no. 6, pp. 36-45, Nov/Dec 2007*.

Hotho, A.; Jaschke, R.; Schmitz, C.; and Stumme, G. 2006. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Reasearch and Applications*, LNAI Volume 4011. Springer. 411–426.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5):604–632.

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

Lam, S., and Reidl, J. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th International WWW Conference*, 393–402.

Macgregor, G., and McCulloch, E. 2006. Collaborative tagging as a knowledge organisation and resource discovery tool. *Library Review* 55(5):291–300.

Mika, P. 2007. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(1):5–15.

Mobasher, B.; Burke, R.; Bhaumik, R.; and Williams, C. 2005. Effective attack models for shilling item-based collaborative filtering systems. In *Proceedings of the 2005 WebKDD Workshop, held in conjuction with ACM SIGKDD'2005*.

Mobasher, B.; Burke, R.; Bhaumik, R.; and Sandvig, J. J. 2007. Attacks and remedies in collaborative recommendation. *IEEE Intelligent Systems* 22(3):56–63.

Niwa, S.; Doi, T.; and Honiden, S. 2006. Web page recommender system based on folksonomy mining for itng'06 submissions. In *Proceedings of the 3rd International Conference on Information Technology: New Generations (ITNG'06)*, 388–393. IEEE Computer Society Washington, DC, USA.

O'Mahony, M.; Hurley, N.; Kushmerick, N.; and Silvestre, G. 2004. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology* 4(4):344–377.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal* 24(5):513–523.

Salton, G.; Wong, A.; and Yang, C. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18(11):613–620.

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

Sen, S.; Lam, S.; Rashid, A. M.; Cosley, D.; Fankowski, D.; Osterhouse, J.; Harper, F. M.; and Riedl, J. 2006. Tagging, communities, vocabulary, evolution. In *Proceedings of 2006 Conference on Computer Supported Cooperative Work (CSCW)*, 181–190.

Xu, Z.; Fu, Y.; Mao, J.; and Su, D. 2006. Towards the semantic web: Collaborative tag suggestions. In *Collaborative Web Tagging Workshop in conjunction with the 15th WWW Conference*.