Appears in W2SP 2009: The Web 2.0 Security and Privacy 2009 Workshop, Oakland, California, May 2009

Evaluating Attack Amplification in Online Social Networks

Blase E. Ur and Vinod Ganapathy
Rutgers University

Abstract

Many online social networking websites allow arbitrary Web users to easily add popular users, such as famous celebrities and musicians, into their circle of friends. Such popular users, or "hubs," have a large number of connections in the social network. However, most online social networks treat such hubs in much the same way as they do ordinary users in terms of security and privacy.

In this paper, we experimentally demonstrate the dangers of not differentiating between hubs and ordinary users. In particular, we show how malicious social network users can leverage their connections with hubs to amplify misdeeds, such as small-scale DDoS attacks and botnet command and control. While instances of these attacks can readily be detected and prevented using previously proposed techniques, the ease with which a social networking user can abuse connections with hubs is a worrisome attack vector. This work also underscores the need for online social networks to have better access control policies for such hubs, *e.g.,* in how users can interact with such hubs.

We conclude with design requirements for online social networking websites that would protect against amplification attacks and yet preserve the freedom of Web users and the openness of social networks.

1 Introduction

Like many types of networks, online social networks such as MySpace, Facebook and Flickr contain *hubs*—members of the network who are extraordinarily well-connected compared to an average user. Such hubs have a large friend circle, have many visitors to their profiles and are centers of activity on social networking sites. Hubs can include both highly social users and popular celebrities, such as famous artists and musicians.

Friendship on a social networking site is generally a binary relationship: two users either are friends or are not. Therefore, friendship with a hub typically allows the same sort of interaction as friendship with an ordinary user. These interactions include allowing friends to post content on each others' profiles and pages. For instance, a section of each MySpace profile is reserved for the user's friends to post comments. The photo-sharing site Flickr similarly allows its users to comment on each other's work. Both Flickr and MySpace, among other social networking sites, allow users to include multimedia content as part of these comments by using HTML tags. For instance, a user can post an image of a "first place ribbon" below her friend's Flickr photographs. On both Flickr and MySpace, users are better able to interact with their friends and acquaintances because of these comment features.

In this paper, we quantify the extent to which a malicious user can exploit her friendship with one of the network's hubs in order to amplify her actions. Amplification is possible in several ways. For example, social networks amplify *content distribution*. A social network user who can freely post content can leverage the social network to have her content downloaded by a large number of other users. Similarly, social networks amplify *bandwidth*. If a user posts a number of hotlinks to media files (hosted by a third party) in a hub's profile, visitors to this profile will cause a flash crowd at the server hosting these files. The ability of arbitrary users to post rich content coupled with the massive scale of social networks thus provides malicious users an ideal platform to launch stealthy attacks.

In order to quantify the extent to which posting on hubs' pages amplifies an attack, we applied two previously-known attack techniques to social networks. In the first attack, we posted hotlinks to large multimedia files, thereby causing a (small-scale) denial of service attack on the server that hosts these files. In the second attack, we simulated using comments on profiles as a botnet command and control channel. Such attacks can possibly be prevented by a number of techniques, such as load balancing, filtering requests by the HTTP referrer tag, or any number of DDoS and botnet detection strategies. However, the ability of an *arbitrary web user* to amplify any malicious attack to a large extent using only her connections to hubs suggests a need for mitigation strategies applied by social networking websites themselves.

An interesting characteristic of the amplification methods we describe is that a malicious user who wants to deploy large-scale, real-world versions of the attacks that we

discuss needs neither special skills nor resources. An attacker needs only to create a social networking account and to possess rudimentary knowledge of HTML. As a result of this very low barrier to entry, any large amplification effect of these attacks is particularly worrisome and thereby motivates solutions at the level of social networking sites. We thus present proactive design recommendations for social networking sites (Section 4) that characterize and discuss the roles and expectations for hubs in a social network, and also suggest monitoring techniques to detect attempts at amplification.

2 Evaluation Methodology

To evaluate amplification in social networks, we adapted two attack techniques to MySpace. The goal of our experiments was to measure amplification rather than launch an actual attack; we therefore took care to prevent disruption to social networking users during our measurements. We chose popular musicians as the hubs on MySpace because many musicians have a large number of friends (over a million, in some cases). These musicians also encourage regular visits to their profiles, which are updated with tour dates and blog posts. Furthermore, MySpace posts a list of "Top Artists," thereby providing a convenient way to identify hubs.

We started by creating MySpace profiles that we could use to befriend hubs in the network. We then used MySpace's "Top Artists" list to identify the top 1,000 major label musicians, top 1,000 independent label musicians, and top 1,000 unsigned artists. Our attacks require that we be able to post HTML tags in comments on these artists' pages; some musicians change MySpace's default settings and disallow HTML tags in comments. We therefore spidered these 3,000 musicians' pages for evidence of HTML in comments; of these users, 1,073 permitted HTML in comments. We sent requests to befriend these users, and 942 accepted our request within three days; we call these 942 users our *target profiles*.

To measure how frequently each target profile is visited, we posted to that profile a comment hotlinking a one pixel image hosted on our web server. Doing so allows us to track the number of unique IP addresses that visit each target profile.

Our first amplification experiment aims to measure how much bandwidth an attacker could consume on another user's server by creating a flash crowd using a hub's profile. In this approach, a malicious user posts hotlinks to large media files hosted by a victim web server. Each visitor to the hub's page will send a request to the victim web server; because a hub receives a large number of hits, requests coerced by hotlinks thereby cause a flash crowd. This attack is somewhat reminiscent of a drive-by-download, except that the victim is not a web surfer, but rather a web server. Although this attack technique has been explored previously on the Web in general [21], our goal was to quantify the extent to which posting hotlinks on a hub's social network would amplify the attack.

To minimize potential disruption to social networking users, we conducted this experiment in two parts. First, we hotlinked a single pixel image (of size 8KB) from our web server in a single comment on 942 musicians' pages. This part of the experiment allowed us to determine the number of unique IP addresses that each of the 942 profiles would refer. For the second part of this experiment, we posted larger comments, which hotlinked 57 images with a total size of 42 MB. In order to minimize any potential disruption to users, we posted this comment on only six musicians' pages. In order to simulate at least moderate traffic on our server, two of these six musicians were among the most popular in our earlier tests. Since some musicians' MySpace pages already load a number of large multimedia files for each visitor, hotlinking 42 MB of files on six pages was designed to provide realistic data yet minimize disruption to users accessing those pages.

Our second amplification experiment aims to measure the dissemination of malicious content posted on a hub's profile. In particular, we studied the effectiveness of using social network profiles as a botnet command and control channel. Such a channel would require bots that scan the contents of a hub's profile when a user visits using an infected computer. Because hubs receive a large number of hits and command delivery happens over the course of normal browsing activity, such a channel provides an effective means to control a large number of bots. Similarly, many bots can be instructed to visit these hubs' profiles to retrieve C&C information without causing anomalous traffic patterns at the social network server since the pages are already very popular. In order to ensure the widest distribution of C&C information with the smallest number of posts, we examined both the popularity of different profiles and the lifetime of a comment before it is pushed off the MySpace page by newer comments.

3 Evaluation

To evaluate the effectiveness of hubs at amplifying attacks, we befriended 942 hubs and posted comments containing hotlinks to an image hosted on our web server, as discussed in Section 2. Overall, during our 12 day test period, we received 2,598,692 hits on our server.

3.1 Experiment 1: Amplifying Bandwidth

Our first experiment, which we conducted in two parts (as described below), measured bandwidth amplification caused by hubs on social networks.

3.1.1 Experiment 1-A: Measuring Unique IPs

In the first part of Experiment 1, in which we tried to maximize the number of unique visitors, we posted comments to 942 hubs' profiles. Because many of these hubs moderated comments, not all of the comments became active on the same day. Indeed, some of the comments never became active. Of the 942 comments we posted, only 719 ever referred a non-trivial amount of traffic to our server. The other comments generated only a few hits, likely from a moderator viewing and then rejecting the comment.

We received 2,598,692 hits on our server over the 12 day test period, from 1,828,589 unique IP addresses. Each time a Web user viewed a page containing our comment, he or she caused a single hit on our server. The busiest day saw 280,112 hits, and thus the MySpace pages containing our comments were viewed 280,112 times that day. A graph of the traffic summed by hour of the day is seen in Figure 1. Diurnal patters are visible in this traffic pattern; we hypothesize that these diurnal patterns result from the majority of users visiting these hubs being teenagers in the U.S., and thus peak traffic periods occur when these teenagers are awake yet not in school.

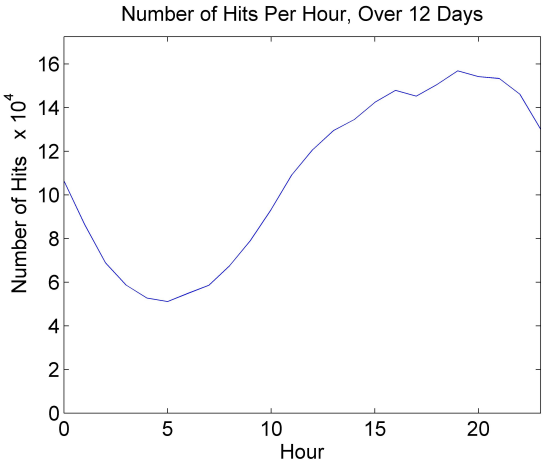

Figure 1: The sum of the hits in each hour of the day over our 12 day test period. Despite evidence of a diurnal pattern corresponding to when American teenagers are awake and not in school, the peak times saw no more than triple the number of hits as the graph's troughs.

The most popular MySpace profile we befriended referred 46,722 hits, whereas a number of profiles referred almost no hits. The cumulative density function of the percentage of hits provided by the top-k profiles is shown in Figure 2. 19% of the profiles referred 50% of the hits. The distribution of popularity among the profiles we tested followed a power law, as predicted in previous work on the structure of networks [4].

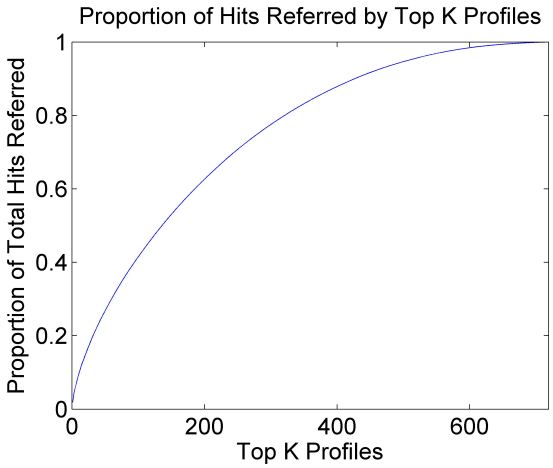

Figure 2: Cumulative density function of the hits referred by the most popular profiles. 19% of profiles generated half of the hits we saw.

3.1.2 Experiment 1-B: Measuring Bandwidth Amplification

To estimate bandwidth amplification, we selected six MySpace hubs that we befriended, and posted a comment containing hotlinks to 57 images of different sizes (classified as small, medium and large), totaling 42MB worth of data, hosted on our web server. We then observed the traffic on our web server for 76 hours. Of these six hubs, two were chosen to be among the most popular referrers, and the other four were chosen at random.

The most popular of these six profiles referred the bulk of the traffic. Our server logged a peak of 20GB of requests per hour from just the single most popular profile. In total, over the 76 hours of our experiment, the most popular referrer drove 606GB of traffic to our server *by itself*.

Since some of the files would be cached, and some users would cancel loading the page before completion, we compared the amount of traffic observed versus the theoretical bandwidth. We consider only the referrer that drove 606GB of traffic to our server.

Based on the number of unique IP addresses contacting our server and the total size of images posted (42MB), a

theoretical upper bound for the amount of traffic driven to our site by that referrer would have been 1,050GB over the experimental run. Since a total of 606GB of traffic was actually observed over those 76 hours, the combined effect of web caches and impatient users leaving pages before downloading everything was observed to contribute less than a 40% loss in efficiency. Thus, we found that bandwidth amplification was operating at around 60% of its theoretical maximum efficiency.

3.1.3 Total Estimate

Combining the data from Experiment 1-A (the amount of unique traffic driven by many different referrers) with the bandwidth data (60% efficiency) from Experiment 1-B, we can estimate the magnitude of a large scale attack. Had even a small 42 MB worth of images been hotlinked on all 719 hubs on which our 1 pixel comment appeared, 65 Terabytes would have been transferred over 12 days were the server able to handle all requests. In the peak hour of traffic, 525GB would have been trasferred.

In order to reduce the number of posts, the attacker could have posted to only the most popular profiles. About 25% of the total traffic in Experiment 1-A was referred by the 44 most popular hubs. Therefore, 130GB of traffic in peak hours could be directed to a third-party by simply hotlinking 42MB of files on 44 profiles. The top 10 hubs alone contributed 10% of the total amount of traffic, meaning that posting a single comment to just ten profiles would direct 50GB of traffic in a peak hour towards a victim. As 42MB was chosen to be a small amount of hotlinked data that wouldn't disrupt users, an actual attack would be even more pernicious than these estimates.

3.1.4 Factors Causing Loss in Bandwidth

We hypothesized that both caching and impatient users leaving web pages contributed to the observed bandwidth being 40% below its theoretical maximum. We analyzed our data to better understand the effects of each.

Of the hits contributing to the 606GB of traffic referred by the most popular hub, 13% of HTTP requests logged either zero bytes or a (constant) small number of bytes transferred. We hypothesize that these 13% of files were cached, therefore having negligible impact on the attack.

To determine how many users navigated away from a profile page before downloading all of the files, we examined the number of downloads for each individual file, in order. We posted 19 small sized image files (8-28KB each), 19 medium sized image files (30-130KB each), and 19 large sized files (1-4 MB). In Figure 3, we show the number of hits received for each file from the most popular referrer.

About 90% of visitors downloaded all small and medium sized files, but only 50% downloaded all large files. Frustrated visitors would abandon their quest and navigate to a different page if a profile did not load quickly enough. Given that the median speed of broadband in the United States is 1.9MBps [30], the average user seems to be leaving a particular MySpace page open for over 2.5 minutes, providing sufficient time for bandwidth amplification attacks to succeed.

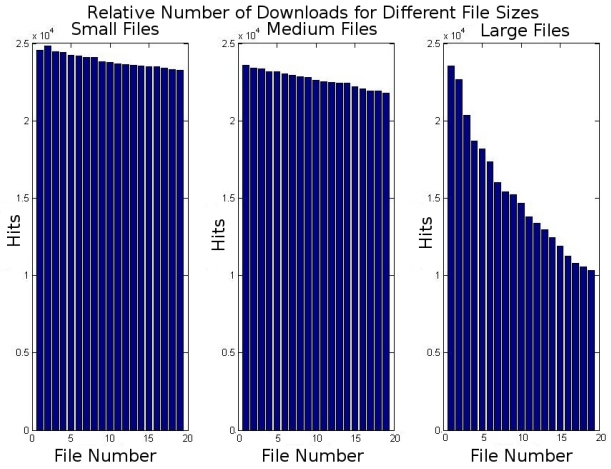

Figure 3: This figure displays the number of hits on each image file. Nearly all users downloaded every small file, yet only 50 percent of users downloaded all large files. This disparity suggests that half of the users navigated away from the page before it finished loading.

3.2 Experiment 2: Amplifying Content Distribution (Botnet C&C)

In addition to amplifying bandwidth usage, posts on social networking hubs' pages can be used for efficient and discreet Botnet Command and Control. Posting Command and Control information on hubs' pages has two major advantages. First, the C&C information can be viewed by many botnet members without creating an anomalous traffic pattern since hubs' pages are already widely viewed by legitimate users. Also, a number of users with infected machines might already visit these pages in normal browsing, obviating the need to distribute C&C information to them.

The effectiveness of a social network profile for C&C use depends on both the popularity of a particular profile as well as the length of time a comment remains on the profile's main page. If a profile is not sufficiently popular, the C&C channel will only be able to deliver commands to a few bot-infected machines without creating unexpected

traffic patterns. If a comment remains on the main page only a short time, it will need to be reposted frequently, thereby requiring more work from the botmaster and increasing the chances of traceback.

3.2.1 Relative Popularity

The experimental data supporting bandwidth amplification can also be used to evaluate the popularity of a profile for content distribution. We focused on minimizing the number of profiles to which either plaintext or a steganographically-enhanced image (containing bot commands) is posted. We concurrently looked to maximize the number of unique visitors.

In our experiments, the five most popular profiles alone accounted for over 100,000 unique visitors (IP addresses) during our 12 day experiment. Thus, by targeting only a handful of the most popular profiles, the botmaster in a Social-C&C attack can maximize the audience for the C&C channel while minimizing the number of locations to which commands must be posted.

As seen in Figure 2, 44 hubs referred 25% of the total traffic. About 10% of the total traffic was contributed by the top 10 hubs alone. Therefore, posting one comment each on 44 carefully selected MySpace pages reached 450,000 unique IP addresses in the 12 days of our test. Single comments posted on just 10 carefully selected pages reached 180,000 unique IP addresses. In fact, even randomly choosing any one of the hubs in the experiment would have referred an average of 2,500 unique IP addresses over 12 days.

Therefore, with minimum effort on the part of a botmaster, a large number of zombie computers could receive commands. In fact, many of the zombie machines may already be visiting those pages.

3.2.2 Lifetime of a Comment

Each MySpace page displays only a few dozen of the most recent comments. As comments age and are replaced by more recent comments, they are relegated to secondary pages and receive very few views. About 11% of the comments we posted remained on their pages through the conclusion of our 12 day experiment, continuing to direct traffic.

For the remaining 89% of comments, we defined the lifetime of a comment to begin when we first saw 10% of its maximum observed number of hits, and to end when it first fell below the 10% threshold. This definition allows us to determine when the comment is on the main page of a MySpace profile, yet still account for temporal variation in the traffic. Once the amount of traffic is sharply reduced, the comment can be hypothesized to have been pushed off the main profile page. We found the median lifetime of these comments to be 137 hours, or 5.5 days. (The mean lifetime was 127 hours.)

Therefore, with only a handful of comments, command and control data for a botnet could be posted to just a handful of popular MySpace profiles, reaching thousands of visitors in peak hours. It would only be necessary to repost comments every few days. Large swaths of a botnet would be able to view those MySpace pages and stealthily retrieve the steganographically hidden commands. Since these profiles normally receive thousands of visitors per hour during peak times, the extra traffic from thousands of bots accessing those profile to obtain C&C data will not seem anomalous, making detection of this attack difficult.

4 Mitigation Techniques

Although the amplification effects seen in our evaluation pale in comparison to those possible from a large, coordinated botnet, these effects remain important for two reasons. First, the idea of using the social network to amplify an attack could be used in conjunction with existing techniques to mount a larger attack. Second, many more individuals on the Internet have the technical capabilities of creating a MySpace account and posting HTML on another user's page than the technical know-how required to create and harness a botnet. Therefore, social networks enable an *arbitrary web user* to launch and amplify the effect of attacks.

Many of the effects of amplification can be detected and mitigated using a variety of previously-known techniques [26, 32, 28, 18, 20]. However, rather than mitigating the symptoms of amplification attacks, it is better to prevent them at the source. The best solutions must therefore address the structural considerations of social networks that enable amplification attacks. However, all solutions must consider social networking users' perspectives. To users, interacting with a hub is much the same as interacting with a close friend; this equality of interaction contrasts sharply with the strict hierarchy of the offline world. Excessive restriction would be considered anathema to many social networking users. In the following sections, we examine possible design considerations (and research directions) to mitigate attack amplification using social networks. With all mitigation efforts, an overriding question must be the extent to which a social networking site should balance a responsibility for preventing malicious uses of its site and a desire to give users freedom and the chance to express themselves.

Solution 1: Restricting Hubs. Since the amplification attacks that we describe exploit the privileges of friendship with a hub, one solution might be to automatically restrict these privileges once a profile becomes a hub. A hub

can be identified in a number of ways, either based on its connectedness (number of friends) or the frequency with which its pages are visited. Once a page reaches a traffic threshold, or once a user accumulates a particular number of friends, privileges could be restricted to all friends. For instance, friends might no longer be permitted to post comments containing HTML tags to the hub's page, or the contents of the page could be automatically filtered to become less bandwidth-intensive.

Although this solution will likely prevent the amplification of attacks that use hubs, it unfortunately restricts user freedom, and is therefore unlikely to be adopted by popular online social networks. For example, social network administrators are loathe to restrict users' freedom and privileges even when its security is threatened [23].

Solution 2: Focused Monitoring. Rather than actively revoking privileges from users or restricting their privileges, a second solution would be to focus automated monitoring efforts, such as anomaly detection systems, on the hubs in a network. In particular, a monitoring solution might look for users who concentrate a disproportionate number of their posts on hubs' pages, post similar messages to multiple hubs' pages, or regularly hotlink media files from hubs' pages. In essence, these solutions identify behaviors that could permit attack amplification and flag actions with these behaviors for moderation (or forensics). Future work in this area would need to create and test an extensive list of the characteristics of attack amplification. Furthermore, such work would need to determine whether users would consider flagging suspicious actions to be in violation of the open spirit of a social network.

Solution 3: A Hierarchy of Friends. An attacker who performs attack amplification is not likely to be a close, personal friend of the hubs she exploits. Therefore, social networks might consider implementing a finer-grained view of friendship. Close friends of hubs might automatically be extended the full privileges of online friendship, such as posting content on the hub's page. In contrast, users who are better described as random acquaintances might be restricted from posting on the profile of a hub, or need to overcome greater hurdles to do so. Such a scheme can be implemented using hierarchical access control schemes, such as role-based access control [12].

For this approach to be effective, there are two major considerations. First, an average user of the social network must not feel that the openness of the network is being compromised. Indeed, the ability to befriend a celebrity idol or other popular figure is one of the thrilling, democratizing elements of a site such as MySpace.

Second, determining how users are related must be an automatic process. Although the seeds of a tiered view of friendship exist in "Friends Lists" on Facebook and MySpace, these features are not widely used. A possible explanation for users' neglect of such lists is the large amount of effort users need to invest in maintaining them. If a social networking site *automatically* pieced together existing clues to determine whether or not two users were close friends, this approach could be successful.

Solution 4: Reputation Systems. The social networking site could employ reputation-based systems that score user behavior. Users with higher reputation scores are allowed more freedom in posting content. This defense is akin to credit-rating systems because a user builds his reputation over a period of time. Good reputation is rewarded with more freedom, but the user risks his reputation with bad behavior. However, reputation systems do not prevent attack amplification. Rather, they increase the resources needed to launch an attack, *e.g.,* an attacker must wait longer to acquire a good reputation before launching an attack.

5 Related Work

Attacks using Social Networks. Prior works describing attacks using social networking sites focus on a number of different areas. Researchers have designed malicious Facebook applications that assemble a botnet from users who install the application [3], similar to attacks on the standard Web that misuse media files to create a flash crowd [21]. Whereas that work focused on describing and measuring the social engineering aspects of convincing a user to install a malicious application, our work looks at a way for malicious users to leverage the structure of a social network to greatly amplify potential attacks. In particular, amplification attacks require no action on the part of users who unwittingly take part in an attack and are thus more effective. Other work has described how social networks can be used to make both spam [5] and phishing attacks [17] more realistic. Our work again differs in focusing not on users' responses to social engineering techniques, but rather on the structure of the social network.

Some of the most high-profile examples of social networking attack techniques are taken directly from attacks observed in the wild. The Koobface Worm [8] sends messages to friends of infected MySpace and Facebook users, using social engineering techniques to coerce the friends into visiting a malicious website to watch a video, and thus infect their own computer. Koobface requires significant effort and technical knowledge on behalf of the worm's authors. In contrast, our work identifies how, with minimal effort and resources, attackers can amplify the effects of an attack. Furthermore, we suggest how social networks could defend against the weaknesses we

identify. Other observed attacks, including worms such as Samy, that utilize XSS (Cross Site Scripting) techniques to spread on social networks [19, 7, 29, 22] differ in similar ways from this paper. Those attacks exploit the social network's structure to spread and thus infect many users. In contrast, our amplification techniques don't spread from users like a worm or a virus. Rather, a large amplification of attacks and actions is possible without using exploits to spread through a network.

Defense Techniques. Although traditional defense techniques against flash crowds [20, 18] and botnet attacks [16, 9, 11, 14, 26, 32] would mitigate the effects of amplification attacks, superior defense techniques would prevent amplification attacks at their source—the social networking site.

Several recent papers investigated how to use the structure of social networking graphs to defend against different classes of attacks. For example, SybilGuard [34] and SybilLimit [33] show how social network data can defend against Sybil attacks. Similarly, other recent projects have used the contents of a social network to improve email security [27, 13]. The SocialTrust project [6] evaluates an algorithm for establishing trust in a network containing malicious users by simulating the algorithm on social graphs scraped from MySpace. Similarly, work [25] in mobile systems proposes using social information to improve the performance of mobile systems, finding for instance that social information used as part of a firewall would improve the firewall's ability to slow the spread of worms without affecting legitimate services. Many other projects [31, 24, 15, 2] describe how knowledge of social connections can be used to make access control decisions.

Each of these methods aims to use the structure of social networks to mitigate classes of attacks. In contrast, our amplification techniques use the structure of social networks to create attacks, rather than to protect against them. Our subsequent design recommendations for social networking sites thus use the structure, rather than the contents, of a social networking graph to stop the attacks that are enabled by that same structure. Our defense approaches need only examine the connectedness (node degree and popularity) of a user to choose a course of action, rather than requiring any knowledge of the contents of a social graph.

Since we consider the structure of a social network to prevent attacks, prior work on the topology of social networks is relevant. Prior work has identified the presence of hubs in both networks in general [4], as well as in social networks specifically [1, 10]. These projects focus on the measurement and understanding of social graphs, rather than on security issues. Our work builds upon these topological observations in order to both initially formulate the attack vector and estimate the number of nodes that will need to be monitored in a defense.

6 Conclusions

Hubs in online social networks play a major role in the daily interactions of users of the network. Currently, social networks do not differentiate between hubs and regular users, and in general, it is easy for an arbitrary user to befriend a hub. This paper shows that hubs can play a major role in allowing arbitrary web users to launch and amplify the effects of previously-known attacks. Although we have yet to observe users performing this sort of attack amplification in the wild, we have observed the idea of posting on hubs' pages being used to amplify advertisements. For example, many hubs' pages are filled with comments advertising other users' concerts, products or promotional items. We also observed instances of users posting 0 pixel comments on hubs' pages that "autoplay" their YouTube videos, inflating their "YouTube views." It is likely only a matter of time before an attacker leverages the hubs in a social network to greatly amplify the effects of attacks such as flash crowds and worm distribution. We therefore conclude that the critical nature of hubs calls for techniques to protect these hubs, either by restricting how users connect to hubs without reducing users' perception of the network's openness, or by closely monitoring hubs for suspicious activity.

7 Acknowledgements

We would like to thank Crystal Maung, Komail Ali, and Ehud Cohen (all of Rutgers University) for their assistance in collecting data for the project.

References

[1] Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *WWW*, 2007.

[2] B. Ali, W. Villegas, and M. Maheswaran. A trust based approach for protecting user data in social networks. In *SESSION: Social computing*, 2007.

[3] E. Athanasopolous, A. Makridakis, S. Antonatos, D. Antoinades, S. Ioannidis, K. G. Anagnostakis, and E. P. Markatos. Antisocial networks: Turning a social network into a botnet. In *Information Security Conference*, September 2008.

[4] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(509), 1999.

[5] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders. Social networks and context-aware spam. In *CSCW*, 2008.

[6] J. Caverlee, L. Liu, and S. Webb. Socialtrust: Tamper-resilient trust establishment in online communities. In *JCDL*, 2008.

[7] E. Chien. Malicious yahooligans. *Virus Bulletin*, 2006.

[8] M. Ciubotariu. W32.koobface.a. *Symantec*, August 2008.

[9] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding detecting and disrupting botnets. In *Workshops on Steps to Reducing Unwanted Traffic on the Internet*, June 2005.

[10] G. Csanyi and B. Szendro. Structure of a large social network. In *PHYSICAL REVIEW E 69, 036131*, 2004.

[11] D. Dagon, G. Gu, C. Lee, and W. Lee. A taxonomy of botnet structures. In *Annual Computer Security Applications Conference*, December 2007.

[12] D. Ferraiolo and D.R. Kuhn. Role-based access controls. In *15th National Computer Security Conference*, 1992.

[13] S. Garriss, M. Kaminsky, M. Freedman, B. Karp, D. Mazieres, and H. Yu. Re: Reliable email. In *NSDI*, April 2006.

[14] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol and structure independent botnet detection. In *USENIX Security Symposium*, August 2008.

[15] M. Hart, R. Johnson, and A. Stent. More content - less control: Access control in the web 2.0. In *W2SP*, 2007.

[16] Thorsten Holz, Moritz Steiner, Frederic Dahl, Ernst Biersack, and Felix Freiling. Measurements and mitigation of peer-to-peer botnets: A case study on the Storm worm. In *Workshop on Large Scale Exploits and Emergent Threats*, April 2008.

[17] T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer. Social phishing. In *Vol. 50, No. 10 Communications of the ACM*, October 2007.

[18] J. Jung, B. Krishnamurty, and M. Rabinovich. Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites. In *International Conference on the World Wide Web*, May 2002.

[19] S. Kamkar. Technical explanation of the myspace worm, 2005.

[20] S. Kandula, D. Katabi, M. Jacob, and A. Berger. Botz.4.sale: Surviving organized ddos attacks that mimic flash crowds. In *Symposium on Networked System Design and Implementation*, April 2004.

[21] V. Lam, S. Antonatos, P. Akritidis, and K. Anagnostakis. Puppetnets: Misusing web browsers as a distributed attack infrastructure. In *ACM Conference on Computer and Communications Security*, October 2006.

[22] E. Levy and I. Arce. New threats and attacks on the world wide web. *IEEE Security & Privacy*, 2006.

[23] I. Mackenzie. Facebook boss rejects app controls. *BBC News*, March 2009.

[24] M. Mannan and P. van Oorschot. Privacy-enhanced sharing of personal content on the web. In *WWW*, 2008.

[25] A. Miklas, K. Gollu, K. Chan, S. Saroiu, K. Gummadi, and E. de Lara. Exploiting social interactions in mobile systems. In *UbiComp*, 2007.

[26] J. Mirkovic and P. Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communications Review*, 3(2), April 2004.

[27] V. Pathak, D. Yao, and L. Iftode. Improving email trustworthiness through social-group key authentication. In *Proceedings of the 5th Conference on Email and Anti-Spam*, August 2008.

[28] T. Peng, C. Leckie, and K. Ramamohanarao. Survey of network-based defense mechanisms countering the dos and ddos problems. *ACM Computing Surveys*, 39(1), April 2007.

[29] P. Ritchie. The security risks of ajax/web 2.0 applications. *Network Security*, 2007.

[30] Speed-Matters. A report on internet speeds in all 50 states. `http://www.speedmatters.org/document-library/sourcematerials/sm_report.pdf`.

[31] A. Tootoonchian, K. Gollu, S. Saroiu Y. Ganjali, and A. Wolman. Lockr: Social access control for web 2.0. In *WOSN*, 2008.

[32] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker. Ddos defense by offense. In *ACM SIGCOMM*, August 2006.

[33] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *SOSP*, 2008.

[34] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *SIGCOMM*, 2006.