Evaluating the Security Risks of Freedom on Social Networking Websites

Rutgers University Technical Report DCS-TR646, January 2009.

Blase E. Ur
Rutgers University
blaseur@rci.rutgers.edu

Crystal Maung
Rutgers University
kyithar@cs.rutgers.edu

Vinod Ganapathy
Rutgers University
vinodg@cs.rutgers.edu

ABSTRACT

Many Web 2.0-based social networking sites permit their users to post comments containing a variety of HTML tags on other users' profiles. In this paper, we show that allowing arbitrary users to post multimedia HTML content on other users' social network profiles is an attack vector. Specifically, we demonstrate three attacks—the Social-DDoS attack, the Social-C&C attack, and the Browser-choking attack—each of which allows an *arbitrary* Web user to jeopardize the security of other Web users.

Using the Social-DDoS attack, a malicious Web user can launch a distributed denial of service attack against a Web server; the Social-C&C attack allows a botmaster to covertly and efficiently deliver commands to bot-infected machines; and the Browser-choking attack cripples Web browsers by increasing their memory consumption and prevents users from viewing targeted social network profiles. We present an experimental evaluation of these attacks on two popular social networking Web sites, Myspace and Flickr. Our results show that the attacks can be highly effective when launched using popular social network profiles. In the context of our results, we discuss the security risks of allowing social network users to post media files on other users' pages, and we conclude with a discussion of possible approaches to mitigate these risks.

1. INTRODUCTION

Online social networks are characterized by rich content and collaboration between users. A large amount of the content on these sites, ranging from personal data to multimedia files, is posted by arbitrary Web users. As part of the collaborative aspects of these sites, users are encouraged to post comments on other users' profiles and pages. However, a number of social networking sites allow users to include multimedia content, using HTML tags, as part of these comments. For instance, Alice can post an image of a blue ribbon on a page Bob has created containing one of his photographs. Then, every other user who visits that photograph's page will also load the blue ribbon. Similarly, a local band can post a comment on a famous musician's profile, and this comment can contain a flier for the local band's concert. Then, every user who visits the popular musician's profile will see the flier for the local band's concert.

However, a comment can contain more than just one picture. A single comment can contain dozens or even hundreds of images, often hotlinked from other sites. While sites filter the HTML posted in comments to eliminate Javascript code (that can lead to cross-site scripting attacks, such as the Samy worm [20]), the filters often don't eliminate the HTML tags for images. Indeed, it could be argued that allowing users to post images allows those users greater expressiveness and creativity on the site, enhancing the content of the social network. However, when users can post a large number of images to highly trafficked parts of a social networking site, a number of brute force attacks become possible.

In this paper, we show that allowing arbitrary Web users the unfettered ability to post large numbers of multimedia files, such as images, to popular social network profiles is an attack vector. By misusing the freedom to post multimedia files, an *arbitrary Web user* can use this attack vector to launch several attacks; we describe three such attacks in this paper—the *Social-DDoS* attack, the *Social-C&C* attack, and the *Browser-choking* attack. Although brute force attacks that leverage large numbers of multimedia files have been previous explored, our attacks are unique and especially pernicious because they can be launched by *any arbitrary Web user* efficiently and without the aid of any special resources.

The Social-DDoS attack allows an arbitrary Web user to launch a distributed denial of service attack on a victim Web server's multimedia resources. An attacker posts a large number of media files, hot-linked from some victim's server, inside of a comment on a popular social networking page, such as the page of a celebrity. Every time someone visits the page, requests are sent to the victim's Web server for those files. The popularity of the page results in a flash-crowd effect, thus exhausting the victim's resources. We studied the impact of the Social-DDoS attack by hosting several media files on our Web server, and posting hotlinks to these files on several popular profiles in MySpace. Over a 17 day period, we received 7.8 million hits on our server, peaking at 65,000 hits in the busiest hour. When those hits represented requests for large files, dozens of gigabytes of data were being transferred each hour. Overall, our findings lead us to conclude that the Social-DDoS attack may affect the availability of files hosted on shared website hosts or other servers with limited bandwidth.

In contrast to the Social-DDoS attack, where the flash crowd targets a victim server, the Social-C&C attack uses the social network to discreetly control bots installed on infected machines. The botmaster posts commands on a social network profile. These commands are delivered when a user with a bot-infected computer visits that profile. Since popular profiles are already receiving thousands of visits per day, thousands of bots can also visit these profiles in order to discreetly receive instructions without creating anomalous network traffic; they will blend right in. Furthermore, given the large proportion of machines that are members of a particular botnet, a number of the machines that would normally visit a popular social network profile will already be infected and thus able to receive commands from the botmaster in the course of normal Web browsing. Thus, other people's popular social network profiles are an effective command delivery channel for botnets. In our experiments, we found that a single comment posted to a handful of care-

.

fully selected MySpace profiles would normally receive thousands of unique visitors per hour for a number of days without needing to be reposted. Thus, many thousands of bots could fetch that particular profile (and the commands hidden in it) without creating abnormal traffic patterns.

In the *Browser-choking* attack, a malicious user posts a single comment containing hundreds of hotlinked images to someone else's social network profile. When other users attempt to view this page, their web browser will suffer from very high memory usage, potentially crippling their browsers or, at the least, making it very difficult for users to view that particular social networking page. We studied the impact of the Browser-choking attack by posting several large images to a profile that we created on Flickr. In each case, the browser's memory utilization quickly reached several hundred megabytes, and the browser was unresponsive. This attack is particularly effective when a user accesses the profile from a bandwidth-constrained mobile device, such as a smart phone.

While there exist techniques to mitigate the effects of flash crowds or defend against our proposed attacks at a number of different levels, the fact that an *arbitrary* Web user can launch these attacks suggests that the attack vector results from a fundamental flaw in the design of social networking sites. *Our thesis is that such attacks are the result of granting too much freedom to arbitrary Web users in highly trafficked locations.* If the users were more tightly constrained in what they could do in those highly trafficked locations, or if they were allowed the same freedoms only in less popular locations on a social networking site, the attacks would not scale to such dangerous levels.

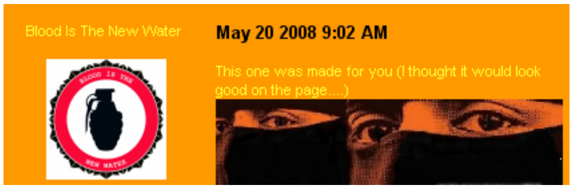
We therefore argue that Web 2.0 social networking sites must be more prudent in proactively reducing unnecessary freedoms and capabilities rather than using reactive techniques, such as load balancing, to thwart these attacks. A sense of scale could be included in HTML filters to disallow these brute force attacks. For example, rather than permitting users to employ HTML in customizing comments on other users' pages, these sites could replace HTML access with GUIs that customize pages and allow users to post only one or two media files, and then only on profiles that are not highly trafficked. More broadly, the attacks that we describe raise the questions of what is reasonable behavior on social networking sites and which freedoms should be permitted in order to minimize security risks without compromising content sharing.

In the rest of this paper, we present background material on the usage of HTML on popular social networking Web sites in Section 2 and discuss our attacks in Section 3. In Section 4, we present our evaluation of these attacks on two popular social networking Web sites. We discuss mitigation strategies in Section 5, related work in Section 6 and conclude in Section 7.

2. BACKGROUND

Web 2.0 social networking sites are broadly defined as those Web sites that rely primarily on their users for content and allow users to make visible connections to each other. There are a few common features shared by these sites. Users are often encouraged to create a profile listing their interests and other personal information. Users are also usually permitted to upload content, such as photos and videos, to the site.

Most importantly, these sites encourage interaction by users, including commenting on each other's profiles and uploaded content. MySpace, one of the most popular social networking sites in the United States, centers around the profiles of individual users, who may be unknown teenagers or who may be famous musicians or actors. At the bottom of each profile, there is space for that user's friends to post comments. MySpace profiles display a few dozen of the most recent comments to all visitors. Figure 1 shows an example of a comment on a MySpace profile. Similarly, Flickr, a popular photo-sharing site, also allows users to comment on each other's work. The site centers around uploaded photographs, each of which has its own page. Users thus post comments on the pages for individual photographs. Each photograph's page will display a few dozen of the most recent comments to everyone who views that page.



Blood Is The New Water

May 20 2008 9:02 AM

This one was made for you (I thought it would look good on the page....)

Figure 1: An example comment posted on another user's MySpace page.

Social networking sites differ in the freedom that they offer to users in posting content. For example, a number of social networking sites, including both MySpace and Flickr, allow users to include HTML tags in their comments on other users' profiles. Sites filter posts for potentially malicious HTML tags *e.g.,* the `<script>` tag is forbidden by most sites, whereas tags such as `<img>` (for displaying images) [13] are usually considered benign and are allowed by several sites. Other social networking sites, such as Facebook and Twitter, forbid users from posting comments that contain HTML tags; any tags posted in a message are rendered verbatim, and are not interpreted as HTML.

We conducted a study to better understand the kinds of HTML tags allowed/filtered in user comments on two popular social networking Web sites—MySpace and Flickr. On MySpace, many HTML tags used for formatting, such as `<b>` and `<em>`, are indeed allowed on user comments, as are `<img>` for displaying images, `<bgsound>` for playing sounds, `<a>` to create hotlinks, `<base>` to specify relative URLs and `<head>` to identify the head section of the document. Particularly noteworthy are the `<img>` and the `<bgsound>` tags that enable a user's friends to post media on her profile without her explicit permission (although some users configure their profiles to moderate the comments posted on their profiles).

As with MySpace, Flickr only permits certain HTML tags to be posted inside comments. Most of the tags permitted on Flickr format text, although the `<img>` tag (displaying images) and `<a>` tag (creating links) are also permitted [13]. However, unlike on MySpace, `<img>` tags on Flickr don't allow users to hotlink images from *other* websites; all images must be hosted on Flickr. No restrictions are placed on hotlinking images hosted on Flickr from other Web sites.

3. ATTACKS

In this section, we describe three classes of attacks that allow arbitrary users to create security risks by leveraging social networking sites. Each of these attacks is contingent upon an attacker's ability to post HTML content, such as images, to other people's profiles. These attacks are particularly noteworthy for two reasons. First, they allow an *arbitrary* Web user to launch attacks, such as denial of service, against victims on the Internet; such attacks previously required an attacker to assemble vast arrays of zombie machines. Second, the attacks are enabled by HTML tags, such as

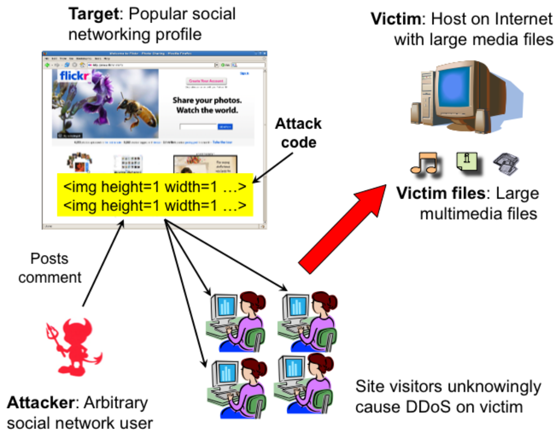
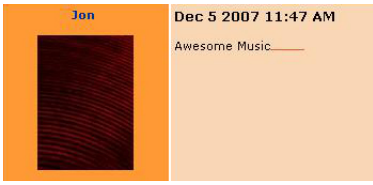

Target: Popular social networking profile
Victim: Host on Internet with large media files
flickr
Share your photos. Watch the world.
Attack code
<img height=1 width=1 ...>
<img height=1 width=1 ...>
Victim files: Large multimedia files
Posts comment
Attacker: Arbitrary social network user
Site visitors unknowingly cause DDoS on victim

Figure 2: In the Social-DDoS attack, a malicious comment posted on a social network profile coerces site visitors to download media files from a victim Web server.



Jon

Dec 5 2007 11:47 AM

Awesome Music

Figure 3: A malicious comment posted to a MySpace profile used in the Social-DDoS attack. The straight line at the end of this message contains a series of hotlinks to high resolution images, scaled down to one pixel each.

`<img>`, that were hitherto considered benign. These tags are currently not filtered out by several social networking sites.

3.1 The Social-DDoS attack

In the Social-DDoS attack, depicted in Figure 2, a malicious Web user exploits the popularity of certain profiles on a social networking site to launch a distributed denial of service attack on some victim Web server. The attacker identifies a handful of media files hosted on the victim server, and then crafts a comment full of hotlinks to those media files. This malicious comment is dubbed the *attack code*. The attacker then searches the social networking site for a popular page, called the *target* in Figure 2. On the *target* profile, the attacker posts the *attack code* as a comment. The *attack code* comment could contain hotlinks to hundreds of images. If each such hotlink points to a high-resolution image, downloading the comment results in a lot of traffic at the server. To visually hide the fact that hundreds of images have been posted, the height and width of each image can be set to a small size, *e.g.,* one pixel. Figure 3 presents an example of such a comment; the straight line at the end of the example message is in fact a series of hotlinks to high resolution images scaled down to one pixel each.

Web users who visit the “infected” *target* profile unknowingly become participants in the attack; each visit to the infected profile results in requests to fetch high-resolution images from the victim’s Web server, thereby resulting in a distributed denial of service attack. Note that this attack is reminiscent of drive-by-download attacks, except that the victim is not the Web surfer, but rather another server on the Internet. This attack is also similar to the “Slashdot” effect, except that it can be initiated by an arbitrary Web user who may hide behind several Sybil identities, thereby making attack attribution difficult.

The success of this attack is contingent on two key factors. First is the popularity of the social network profile on which the malicious comment is posted. Popular profiles on social network sites receive tens of thousands of views per day, and the attacker must choose such profiles as the *target* for malicious comments. Second, the attack depends on hotlinking high-resolution images (or similarly large multimedia files) hosted on the victim’s server. Provided that the attacker can satisfy the two conditions above, the resulting denial of service attack is a fairly powerful attack that only requires the attacker to have an account on a social networking site.

As a distributed denial of service attack with a number of characteristics, including the fact that all requests list the social networking site in the HTTP referrer, this attack can be detected (and possibly prevented) at either the victim’s server or at the social networking server using one of several previously-proposed techniques. For example, the victim could filter incoming Web requests using the HTTP referrer tag and drop requests originating from an infected profile. Similarly, the social networking server could identify infected profiles and remove the offending comments. Such *reactive* defenses can be effective, but the underlying problem remains. If the victim filters and drops requests originating from one profile, the attacker could simply repeat the attack using another profile on another social networking site. We instead advocate *proactive* defenses, to be implemented by the social networking server, that would *prevent* the attack from happening. We defer discussion of mitigation strategies to Section 5.

3.2 The Social-C&C attack

The basic attack technique used in the Social-DDoS attack can instead be used by a botmaster to deliver commands to bot-infected machines. In this attack, called the Social-C&C attack, the attacker uses comments posted to a social network profile as a command and control (C&C) channel for a botnet. The attacker posts commands as comments on a social network profile. When a user with a bot-infected computer visits this profile, the bot scans the profile for comments containing commands. The Social-C&C attack is similar to a drive-by-download attack; the main difference is that the social network profile is used as a vehicle for command delivery rather than malware delivery.

Three characteristics make social networking site comments a particularly effective C&C channel. First, popular social network profiles, such as those of music bands and celebrities, receive several thousand hits per day. With an estimated 25% of computers on the Internet predicted to be bot-infected, commands posted on profiles can effectively be delivered to thousands of bot-infected machines. In addition, the traffic that accompanies thousands of bots retrieving these profiles would not be anomalous since thousands of users each hour retrieve the profiles.

Second, commands can easily be disguised to avoid being detected by malware scanners or by security administrators. For example, commands can be steganographically hidden within images posted to the profile. Even if the social network prevents displaying images, commands can possibly be posted as plaintext messages that would be interpreted by the bot executable. Third, the ease with which social networks admit new users to the network ensures that the botmaster can avoid being detected. The botmaster could create several Sybil identities and use these to post bot commands. Traceback mechanisms employed by the social networking server can identify the username that posted the comments, but cannot effectively identify the botmaster.

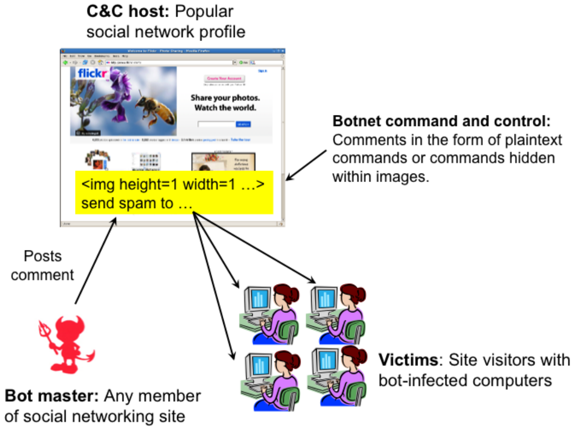

C&C host: Popular social network profile
flickr
Share your photos. Watch the world.
Botnet command and control: Comments in the form of plaintext commands or commands hidden within images.
<img height=1 width=1 …>
send spam to …
Posts comment
Bot master: Any member of social networking site
Victims: Site visitors with bot-infected computers

Figure 4: In the Social-C&C attack, the botmaster uses comments posted to a social network profile as a command and control channel for bot-infected machines.

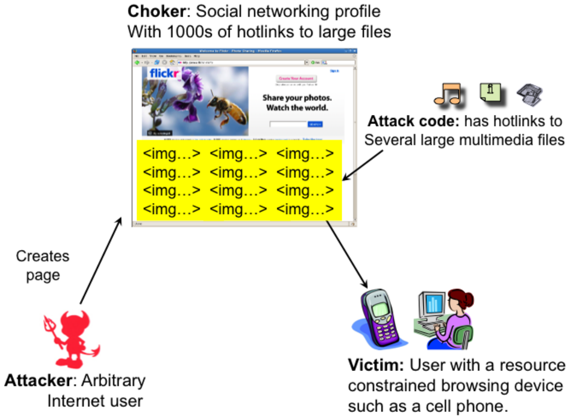

Choker: Social networking profile With 1000s of hotlinks to large files
flickr
Share your photos. Watch the world.
Attack code: has hotlinks to Several large multimedia files
<img…> <img…> <img…>
<img…> <img…> <img…>
<img…> <img…> <img…>
<img…> <img…> <img…>
Creates page
Attacker: Arbitrary Internet user
Victim: User with a resource constrained browsing device such as a cell phone.

Figure 5: In the Browser-choking attack, the attacker creates a page with a large number of high-resolution images. When a victim visits this Web page, downloading these images greatly increases the memory utilization of his Web browser. The attack is particularly effective when the browsing platform is a resource-constrained device, such as a cellular phone.

3.3 The Browser-choking attack

Figure 5 depicts the Browser-choking attack, where the attacker's goal is to increase the memory consumption of the victim's Web browser. As in the Social-DDoS attack, the attacker creates a Web page (or posts a comment) with hotlinks to several hundred high-resolution images. However, unlike the Social-DDoS attack, the goal is not to cause a denial of service on the Web server hosting the images; rather, it is to increase the memory consumption of the victim's Web browser when he visits that Web site. As a result, the images would ideally be hosted by a high-bandwidth or load balanced server, such as a popular image host. As with the Social-DDoS attack, the attacker can hide the images by scaling them down to one pixel each, so as to hide visual cues of the attack from the victim.

In the Browser-choking attack, the attacker forces the victim into downloading hundreds of images, and thus up to hundreds of megabytes of data. Particularly if the victim uses a cellular phone to browse the Web, this attack can cause a denial of resources. Phone users who do not have unlimited data plans will suffer financial ramifications from this attack. Users who do have unlimited data plans will find their phone temporarily slowed by high memory consumption of the browser process. Even on a desktop, high memory usage will cause browser instability or significantly slow browsing activity until that page is closed. This can be particularly problematic if the browser is concurrently being used with stateful Web 2.0 applications, such as Web-based desktops and Web-based spreadsheets. Causing the Web browser to crash may result in the loss of unsaved data.

4. EVALUATION

We evaluated the attacks described in Section 3 by implementing them on two popular social networking sites—MySpace and Flickr. We tested the Social-DDoS and Social-C&C attacks on MySpace, and the Browser-choking attack on Flickr.

4.1 The Social-DDoS attack

Methodology. In order to test the Social-DDoS attack, we set up our own Web server that hosted several large images, and attacked this server by posting malicious comments on MySpace.[1] MySpace is ideal for testing the Social-DDoS attack since it has many extremely popular pages, yet allows arbitrary users to post media files on many of those pages. These popular users are often musicians or other celebrities.

In order to identify *target* profiles—the MySpace pages on which an attacker posts *attack code*—we consulted the "Top Artists" list on MySpace. "Top Artists" links to the profiles of the hundred most popular major label musicians, hundred most popular independent label musicians, and the hundred most popular unsigned/local musicians. However, not all of these profiles are potential targets because MySpace allows users to disable HTML in comments posted to their pages. We wrote a spider that crawled the profiles of these artists to identify which profiles contained HTML tags inside of comments. Our spider discovered that 153 artists of the 300 permitted HTML tags inside other users' comments. We then befriended the 153 artists whom we identified as permitting HTML comments. Many of these musicians (94/153) accepted our friendship requests, as expected [7]. We then conducted a two-part experiment, described below.

Part 1 of our experiment characterized the traffic from a set of malicious comments. On each musician's profile, we would post a single comment containing nineteen `<img>` tags, each hotlinking an image from our Web server. We chose images with small file sizes (a few kilobytes) so that we could accurately gauge the number of hits on our server. However, since it took time for the musicians to accept our friendship requests, we conducted *Part 1* in two separate phases. In *Part 1-A* of the experiment, we posted comments to the first 40 MySpace profiles that accepted our friendship requests. We made these posts at Hour 3 of the experiment. In *Part 1-B* of the experiment, we posted comments to an additional 50 profiles that accepted our friendship requests in the interim. We

[1] We implemented this attack only on MySpace because it allows comments to hotlink to images hosted on third-party Web servers. In contrast, Flickr only allows `<img>` tags to reference images hosted on Flickr, precluding accurate measurements of the attack.

made these posts at Hour 155 of the experiment. We collectively refer to *Parts 1-A and 1-B* as *Part 1*.

Many musicians moderate the comments on their profile. Therefore, although we posted 40 comments in Hour 3 as *Part 1-A*, each comment began driving traffic to our server only once the moderator accepted the comment. By Hour 96 of the experiment, our comments on 34 different profiles had been approved by the moderators and were driving traffic to our server. The 50 profiles we posted in Hour 155, as *Part 1-B* of the experiment, added to the total traffic driven to our server. By Hour 173, our comments to 81 different profiles had been approved by the moderators.

Part 2 of our experiment, which begins at Hour 331, investigated the bandwidth an attack would consume. In particular, we wanted to determine the effect that caching or impatient users leaving a page might have on the attack's efficacy. We thus posted three consecutive comments to each of six different MySpace profiles. Two of these profiles were chosen from among the most popular in *Part 1*. The first comment contained the same nineteen images (8-28KB each) from *Part 1*; the second comment contained nineteen medium sized images (30-130KB each); the third comment contained ninetten large images (1-4MB). If a user successfully downloaded all three comments, 57 images totalling 42MB would be displayed.

If the small images are displayed in our server's log but the large images are not, or if the HTTP GET requests for any of the images gives a file size equal to 0, there is evidence of caching (on the server side and the client side, respectively). Furthermore, since the <img> tags were posted in order of ascending file name, we would be able to determine when users became frustrated and navigated away from the page; they would have fetched the first K images, but not the rest.

4.1.1 Overall Traffic Evaluation

Overall, during our seventeen day test period (encompassing both *Parts 1 and 2* of the experiment), we received 7,796,922 hits on our server from 295,645 unique IP addresses. We posted malicious comments to 94 social network profiles.

In the peak hour of traffic from *Part 1* of the experiment (Hours 0-330), we received 64,421 hits *within the hour* from 21:00-21:59 on October 21st (Hour 164). Since each visitor to a profile caused at most 19 hits on our server (for the 19 images), the 64,421 hits represent a minimum of 3,400 page visits.

The amount of traffic on a minute-by-minute basis was fairly consistent, as seen in Figure 6. Within this busiest hour, the maximum number of hits observed in one minute was only double the minimum number of hits observed. That the amount of traffic remains fairly consistent suggests that the Social-DDoS attack creates a fairly constant load on the victim server during the peak hour.

4.1.2 Hourly Patterns

While the amount of traffic each minute in a particular hour was fairly consistent, the amount of traffic from hour to hour over the course of our experiment was not. Figure 7 displays the number of hits in each hour, encompassing both *Parts 1 and 2* of the experiment. In Figure 7, Hour 0 was the first hour of the experiment (00:00-00:59 EDT on October 15th) whereas Hour 407 was the last (23:00-23:59 EDT on October 31st). We posted comments at three distinct times, and our three sets of posts result in the large increases of traffic that appear following Figure 7 at Hour 3 (*Part 1-A* comments are posted), Hour 155 (*Part 1-B* comments are posted), and Hour 331 (*Part 2* comments are posted). Since each profile from *Part 2* contained three times as many images, the number of hits for *Part 2* overrepresents the number of unique visits compared to *Part 1* by a factor of 3.

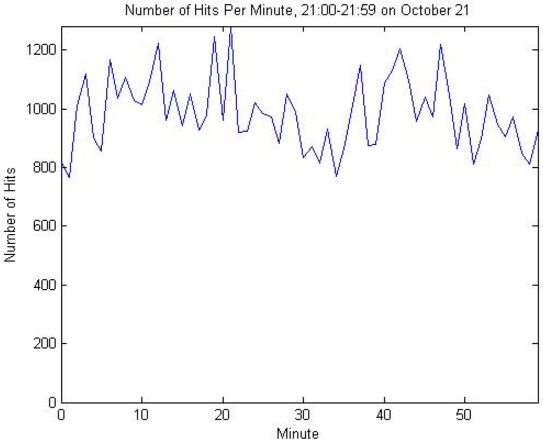


Figure 6: This figure shows the number of hits per minutes during the busiest hour of Part 1 of the experiment. The number of hits per minute was reasonably consistent and non-bursty from minute to minute, in contrast to the traffic pattern from hour to hour. Thus, during peak times, a consistently heavy load will be placed on the victim server.

Figure 7 is periodic, spiking about every 24 hours. These huge bursts of traffic are seen during the evening prime time hours in the continental United States, particularly between 21:00 and 23:00 (EDT). The periodicity of this traffic is displayed more starkly in Figure 8, which shows the total (and median) number of hits in each hour of the day, averaged over our 17 day test period. Spikes in traffic occur between 21:00 and 23:59 (EDT), and the median busiest hour overall was 23:00PM-23:59 PM EDT. The graph's lows occur from 5:00 to 7:59 (EDT) and are approximately an order of magnitude lower than the maxima. This pattern suggests that the vast majority of our hits come from the continental United States. This conclusion makes sense since most of MySpace's Top Artists are American, and MySpace itself is most popular among Americans.

As a result of this temporal periodicity, our Social-DDoS attack is most effective during evening primetime hours in the continental United States; these primetime hours account for the majority of traffic. In the evening, victim servers in the United States will receive a large amount of traffic. Since legitimate traffic to U.S. Web sites would also be high during the evening, the attacks disrupt an important time of day.

4.1.3 Relative popularity of referers

Over the two weeks of our experiment, 94 different MySpace profiles referred traffic to our server. The most popular MySpace profile we tested referred 1.7 million hits, whereas the least popular referred just 2 hits (likely in rejecting the comment during the moderation stage). Since each comment contained nineteen images, the most popular profile must have directed at least 92,000 unique visits to our server. Out of the 94 MySpace profiles, 15 profiles referred over 100,000 hits, 51 profiles referred over 10,000 hits, and 16 profiles referred fewer than 1,000 hits. Many of these final 16 profiles caused hits only when the comment was being moderated, and subsequently rejected.

The cumulative density function of the percentage of hits provided by the top-*k* profiles is shown in Figure 9. Since the amount of traffic referred is directly proportional to the popularity of artists in the MySpace network, it is not surprising to find that this distribution follows a power law [5]. In fact, more than half of the

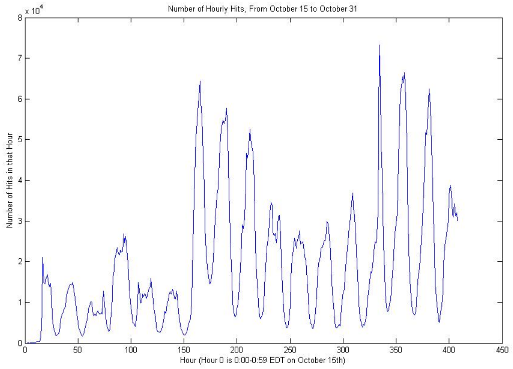


Figure 7: This figure shows the number of hits on our server in each hour of our experiment. Hour 0 corresponds to the first hour of Part 1-A of the experiment, 0:00-0:59 on October 15th. In Hour 3, malicious comments were posted to nearly 40 profiles. Because of moderation, these comments became live between Hour 3 and Hour 112. In Hour 155 (Part 1-B of the experiment), malicious comments were posted to 50 additional profiles; a number of high-traffic comments became live soon after. Part 2 of the experiment began in Hour 331, in which comments containing 57 images of varying sizes were posted to six different profiles, including the most popular profile from Part 1.

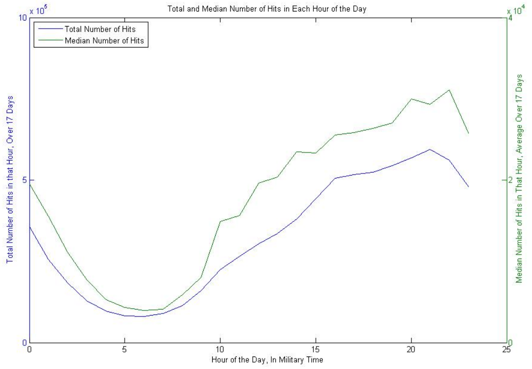


Figure 8: This figure shows the total and median number of hits in each hour of the day, over the 17 days of our experiment. The amount of traffic peaks during prime time evening hours in the continental United States, which means that traffic to our server experienced a large number of bursts in the evening hours.

hits during our experiment were referred by just the five most popular profiles out of the 94. In contrast, the 65 least popular profiles referred only ten percent of the hits.

4.1.4 Bandwidth analysis

In order to evaluate the bandwidth consumption of a Social-DDoS attack, we took a two-pronged approach that combined experimental data and theoretical bounds.

As described earlier, in Part 2 of our experiment, we uploaded new comments to six of our previously targeted MySpace profiles; two of these six profiles were among the most popular referers from the earlier parts of our experiment. To each of these six profiles, we posted 57 images totaling 42MB.

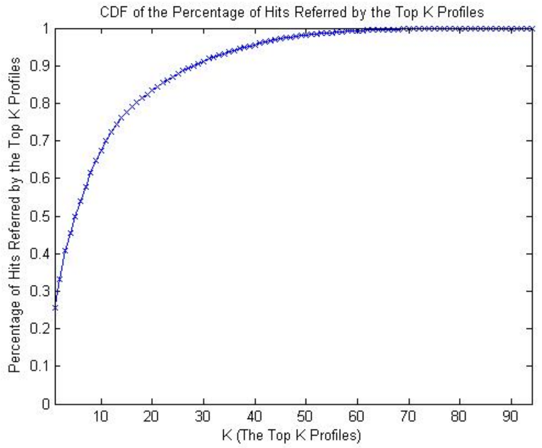


Figure 9: This cumulative density function shows what percentage of hits were referred by the most popular profiles. Over 50 percent of the total hits in our experiment were contributed by the top 5 profiles, suggesting that the handful of most popular MySpace profiles receive the majority of the traffic.

The most popular of the six profiles referred the bulk of the traffic. Our server logged a peak of 20 Gigabytes of requests per hour referred by just the most popular profile. A graph of the number of bytes requested versus the time (in hours) is presented in Figure 10. In total, over the final 76 hours of our experiment, the most popular referrer drove 606 Gigabytes of traffic to our server *by itself*.

Of the hits contributing to those 606 Gigabytes of traffic, 87% of the HTTP GET requests logged filesizes equal to the actual size of the requested file. However, a sizeable minority of requests (13%) logged zero bytes transferred. We hypothesize that these 13% of files were cached, therefore having negligible impact on the attack. There was no evidence that the decision of whether or not to cache a file discriminated based on a file's size; generally, a particular IP address logged either zero bytes for every file, or it logged the correct file sizes.

However, this experimental attack was designed not to overload our server. Therefore, we tested only six profiles in *Part 2*, four of which were unpopular. To estimate a theoretical bound on the perniciousness of this attack, we can consider the number of unique IP addresses driven to the site during the attack and multiply that by the potential bandwidth used by each attacker, taking into account the effect of caching.

We considered just the most popular referrer during Part 2 of the experiment in these calculations, keeping in mind that this referrer alone contributed about 25% of the total number of hits overall during our experiment. First, 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. Files 1-19 were the small sized image files, Files 81-99 were the medium sized image files, and files 101-119 were the large sized files.

We show the number of hits received for each file from the most popular referrer in Figure 11. In each comment, their `<img>` tags were posted sequentially, in ascending order. Nearly all browsers would download the first file, whereas not all would make it to the last file.

For the small and medium image files, about 90% of visitors downloaded every file. However, for the large image files, only

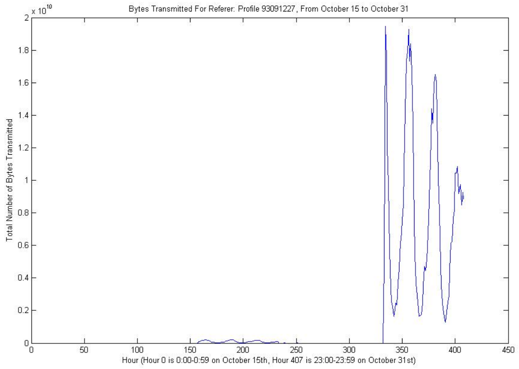


Figure 10: This figure shows the bandwidth (bytes per hour) of files transmitted from just the most popular referrer, MySpace profile 93091227. At its peak, nearly 20 gigabytes per hour were transmitted due to this profile alone. Over the final 76 hours, over 600 gigabytes of data were transmitted overall. No comments were posted to this profile in Part 1-A of the experiment, which is why no bandwidth is observed prior to Hour 150. In Part 1-B of the experiment (Hours 150-250), only 19 images with small file sizes were posted. In Part 2 of the experiment (Hours 330-407) small, medium, and large images totaling 42MB were posted to the profile; therefore, each visitor to the profile requested much more bandwidth in Part 2.

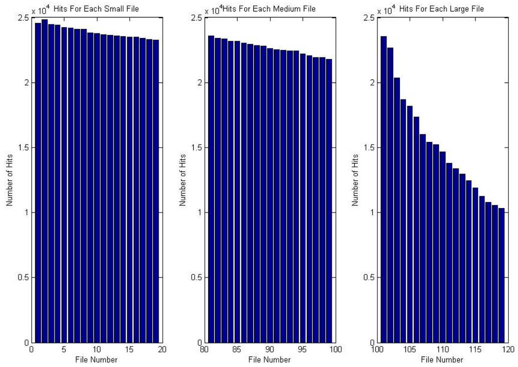


Figure 11: This figure displays the number of hits on each image file in Part 2 of the experiment, considering only hits from the top referrer. Files 1-19 were small (under 30KB), Files 81-99 were medium sized (under 130KB), and Files 101-119 were large (1-4MB). In each comment, the <img> tags were posted sequentially. Thus, a browser that downloaded the images sequentially but navigated away from the page before completing the transfer would likely download only the first few images.

50% of visitors downloaded every file, with an approximately linear drop-off from file to file. This is expected, because the large files alone totalled nearly 40MB in size. Frustrated visitors would abandon their quest and navigate to a different page if a profile did not load quickly enough. Regardless, about half of the users downloaded all of the large files. Given that the median speed of broadband in the United States is 1.9MBps [29], the average user seems to be leaving a particular MySpace page open for over 2.5 minutes, which gives the Social-DDoS attack sufficient time to download imges. Given that MySpace pages for musicians often contain media content, the fact that MySpace users show some patience in waiting for a page to load is not surprising.

In the 76-hour run of Part 3, the most popular referrer directed 20,858 unique IP addresses to our site. The most popular individual file logged approximately 25,000 hits, meaning that there were no more than 25,000 total visits (non-unique). Since the small, medium, and large comments placed on the most popular profile totaled 42MB, a theoretical upper bound for the amount of traffic driven to our site would have been 1,050 Gigabytes over the experimental run. Since a total of 606 Gigabytes 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 percent loss in efficiency. Thus, we found that the Social-DDoS attack was operating at around 60% of its theoretical maximum efficiency.

Since the most popular referrer directed 20 Gigabytes of traffic per hour at its peak, and this referrer contributed only 25% of the total number of hits in our experiment, the attack can be scaled roughly linearly. Since the top five profiles accounted for half of our total hits, 40 Gigabytes of traffic per hour would be possible from posting a single comment to just five profiles in total.

4.2 The Social-C&C attack

Methodology. The effectiveness of a social network profile as a discreet command and control channel for botnets 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. 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.

The experimental data supporting the Social-DDoS attack in the previous section can also be used to evaluate the popularity of a profile for the Social-C&C attack. We focused on minimizing the number of profiles to which either plaintext or a steganographically enhanced image is posted. We concurrently looked to maximize the number of unique visitors. Note that the Social-C&C attack could also have been executed on Flickr. However, as with the Social-DDoS attack, we lacked a reliable method of analyzing traffic patterns from Flickr because `<img>` tags posted to Flickr comments can only hotlink to other images on Flickr itself.

4.2.1 Popularity of each profile

In our tests, the five most popular profiles alone accounted for over 150,000 unique visitors during our 17 day experiment. Thus, by targetting 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.

4.2.2 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. In our experiments, comments posted on even the most popular profiles stayed on the main page for at least a few days. A single comment remained on the main page of even the most popular profile (peaking at 40,000 hits, or 2,000 unique visitors, per hour) for over three days, as shown in Figure 12. Our comment remained on the main page even longer on the less popular profiles; many comments from early in the experiment remained on the main page even at the end of our 17 day experiment.

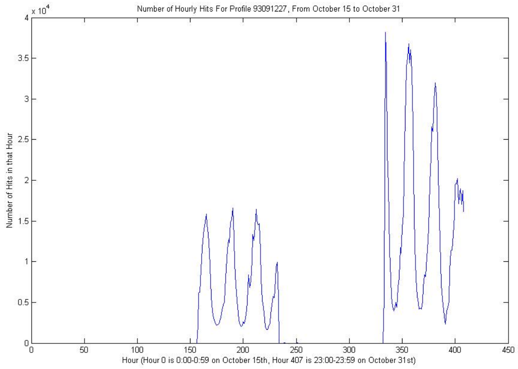


Figure 12: This figure shows the number of hits per hour referred by profile 93091227, the top referrer. The first set of points (Hours 150-250) corresponds to Part 1-B of the experiment, in which each visitor would make up to 19 hits. The second set of points (Hours 330-407) correspond to Part 2 of the experiment, in which 57 images, rather than 19, were posted, causing the number of hourly hits to nearly triple. The precipitous drop in the number of hits around Hour 230 corresponds to when our comment was pushed off of the main page of the profile by more recent comments.

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. 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.

4.3 The Browser-choking attack

Methodology. To test the Browser-choking attack, we needed to measure the memory consumption of web browsers when viewing a social networking page containing a malicious comment full of hundreds of images. We posted a dummy photograph to Flickr [2] and posted a comment containing 400 images to this page. The images in the comment were scaled down to just one pixel each[3]. To do this, we wrote a spider that sampled 400 recent images from the "recent photos" page of Flickr (`http://www.flickr.com/photos`), and chose the highest resolution version available for each image. These images ranged in size from a few dozen kilobytes to a few megabytes each. We chose Flickr over MySpace to implement this attack because MySpace places an size limit on the length of comments posted to a profile, limiting each individual comment to about 30 images. Because Flickr imposes no reasonable length restriction on the size of each comment, we tested our attack on Flickr. Recall that Flickr allows comments with `<img>` tags, but only to images hosted on Flickr itself. However, this restriction did not impede the Browser-choking attack because high resolution images are available aplenty on Flickr.

4.3.1 Memory utilization

We visited our dummy Flickr post and measured the memory utilization of a Web browser. We used a laptop computer (with a 1.66GhZ Intel Core 2 T5500 processor and 1GB RAM running Windows XP SP2) with a 11Mbps wireless link to the Internet. We tested the memory utilization of the Internet Explorer 7, Firefox 2.0, and Firefox 3.0 web browsers as each viewed this post in turn. Figure 13 shows the memory utilization of each Web browser.

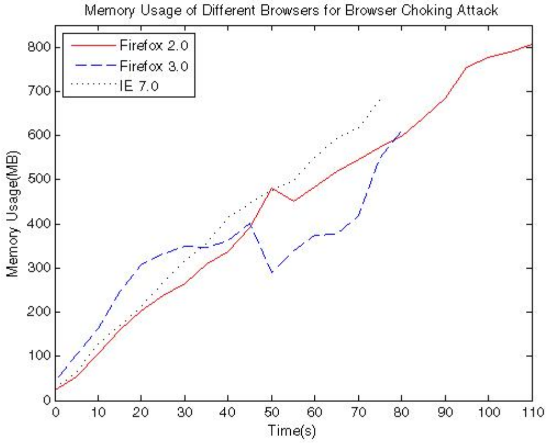


Figure 13: Memory utilization of varios Web browsers during the Browser-choking attack.

As Figure 13 shows, memory utilization rapidly increases when a user visits the infected Flickr profile. Within 10 seconds, memory utilization topped 100mb on all three browsers. Within two minutes, memory utilization had peaked at 600mb for IE 7.0, 700mb for Firefox 3.0, and 800mb for Firefox 2.0. Since this system had only 1 gigabyte of RAM, these browsers were using most of the system's memory at this time.

By the end of the experiment, both versions of the Firefox browser were unresponsive and the OS started swapping the browser process. Similarly, when we attempted to scroll down the page in IE 7.0, that browser became similarly unresponsive. Most users would kill the browser process at this point, which could lead to loss of data, especially if the browser is concurrently being used with stateful Web 2.0 applications.

Although we did not do so in our experiments, if the victim visited the infected Flickr profile using a cellular phone, this attack would also result in several hundred megabytes of data being downloaded to the phone. Because the images posted on the profile are scaled down to a pixel each, there are few visual cues that the machine is in the process of downloading large amounts of data and that the browser's memory consumption is continuously increasing; this attack would successfully work on all but a few technically-savvy users. Users who do not have unlimited data plans will therefore suffer financial losses because of the amount of data downloaded on their machines. Increased memory and CPU utilization may also drain the cellular phone's battery.

5. DISCUSSION

The attacks described in this paper can be detected and prevented using a variety of previously-known techniques [24, 31, 25, 19, 21]. Indeed, the goal of this paper is *not* to demonstrate virulent and indefensible attacks, but rather to show that social networks are a framework via which ordinary, powerless adversaries can feasibly

[2] `http://www.flickr.com/photos/31302508@N04/3000384525`

[3] In an actual Browser-choking attack, the attacker would post such a comment on a popular Flickr profile. However, because our goal was to measure memory consumption of the browser, our experiments did not require us to deface a popular profile; we therefore chose not to do so.

launch attacks that normally require powerful attackers with significant resources at their disposal. We therefore question the security design principles of current and future social networking sites. The attacks presented in this paper are predicated on users' freedom to add media files (using HTML) onto other users' pages. By allowing the use of HTML, albeit filtered for certain tags, MySpace and Flickr allow profiles and comments to have unique appearances. However, it is possible for the social networking servers to implement this customization using several alternative techniques that offer less freedom—and yet the same expressive power—to social network users. The techniques discussed below may *proactively* prevent the kinds of attacks discussed in this paper, obviating the need for *reactive* techniques.

Limiting the number of HTML tags. The social networking site could limit the number and form of HTML tags that can appear in an HTML comment. For example, it can restrict the maximum number of hotlinks using `<img>` tags per comment and can also restrict the number of posts by a user on each profile over a time interval. The server could implement such a scheme as part of its HTML filter as it receives the comments. Alternately, for usability, it could provide a GUI that allows the users to customize the content and formatting of comments. Indeed popular third-party profile editors have already spring up [8], adding an interface to MySpace's HTML capabilities to meet the desires of MySpace users.

Forbidding media files on popular profiles. The Social-DDoS attack and Social-C&C attack both depend on the *attack code* being posted to very popular profiles. Thus, the social networking site could automatically forbid media files from profiles that pass some threshold of popularity. The average user will still be able to post multimedia content in comments to his friend's pages, yet the impact of the Social-DDoS and Social-C&C attacks will be greatly reduced.

Using 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 awarded with more freedom, but the user risks his reputation with bad behavior.

Controls over account creation. The ease of account creation on social networking websites allows malicious parties to create multiple accounts (called Sybil identities). This in turn reduces the effort and risk for attackers to engage in malicious activities. For example, Sybil identities allow a botmaster to launch the Social-C&C attack without the fear of traceback.

This problem can be addressed with stronger authentication mechanisms. For instance, social networking sites could tie user accounts to data such as a public key during registration. Doing so may enable easy attribution of user behavior to actual identities.

Controls over social networking. Social networking sites place few restrictions on how users network with each other. Keen to improve their social profile, users (and celebrities in particular) often accept friend requests from strangers, some of whom may be attackers.

While restricting networking would contradict the purpose of social networking Web sites, these sites can offer tiered levels of friendship rather than the binary relationship (friend of not) that currently pervades sites. For example, the social networking site could allow a user to specify a trust level when he accepts a friend request from a stranger. (Alternately, the social network itself could infer a trust level, based on user behavior.) The trust level can later be used to determine the freedom that a user has in posting content to his friends' pages. Social networking sites such as Facebook already implement tiered privacy controls based on user-defined groups. The same trust levels can easily be extended to restrict the way in which one user interacts with another user.

6. RELATED WORK

We classify related work into four categories, one for each flavor of attack, and one for social networking research in general.

The Social-DDoS attack. Most closely related to the Social-DDoS attack are the work on Puppetnets [22] and Antisocial Networks [4]. In the Puppetnets attack, the owner of a popular website posts Javascript or HTML code that hotlinks a large number of media files from a victim website. All visitors to the popular website are coerced to download images from the victim, thereby causing a denial of service attack on the victim. Our Social-DDoS attack contrasts with Puppetnets in that Puppetnets requires the attacker to be resource-privileged, *e.g.,* the Web master of a very popular Web site. Our Social-DDoS attack does not require the attacker to have any resources other than an account on a social networking site, thus making our attack applicable to a much broader range of possible attackers.

The work on Antisocial Networks is based upon the same principles as the Puppetnets attack, but uses a different attack vector. An attacker creates a malicious Facebook application and uses social engineering techniques or deception to encourage users to install that application. In the paper, for instance, the authors built a bot (called FaceBot) under the guise of an application that displays a picture of the day. When installed by a Facebook user, this bot covertly directed traffic towards a victim host, thereby causing a denial of service. This work differs from the Social-DDoS attack in two ways. First, Antisocial Networks require social engineering techniques to coerce users into downloading and executing a malicious application. In contrast, our attack works much like a drive-by-download attack—*simply visiting the infected Web page results in traffic being directed to the victim server*. Second, for Antisocial Networks to be effective, a large number of Facebook users must actively choose to install the malicious Facebook application. As with Puppetnets, creating an effective Antisocial Networks attack requires significantly outlays in resources by an attacker who must be skilled at creating Facebook applications and social engineering.

In contrast, the Social-DDoS attack can be launched by an arbitrary social network user. A Social-DDoS attacker relies on the popularity of *other users'* profiles to launch the attack, whereas a Puppetnets or Antisocial Networks attacker relies on the popularity of his own page or application.

The Social-C&C attack. Botnets are arguably the biggest threat to the Internet infrastructure, and much research has been devoted to studying the characteristics and propagation of botnets [12, 14, 11, 16, 15]. Botnets require communication between the botmaster and bot-infected machines via a C&C channel. However using centralized C&C channels, such as IRC, results in easy detection and quarantine of infected machines [6, 10]. This has lead to the use of more covert channels for C&C, including newsgroups, P2P communication [1, 2, 18], and smaller botnets [30].

Recent work demonstrated the use of email as a C&C channel [28]. In this work, commands to a bot are hidden using steganographic techniques within email messages. The authors demonstrate the use of both spam and non-spam messages to deliver commands to bots. Email as a C&C channel is particularly effective because of its decentralized nature. In particular, the attacker can send commands from one email address and then abandon that email address. Our Social-C&C attack is comparable in power to email as a C&C channel both because it operates stealthily and also because the attacker can use several Sybil identities to avoid being detected by the social networking site.

The Browser-choking attack. Our Browser-choking attack attempts to exhaust resources, *e.g.,* memory and bandwidth, especially on resource-constrained devices. Prior work has also developed techniques to exhaust resources on such devices. For instance work by Racic *et al.* [26] showed that MMS vulnerabilities can be used to stealthily exhaust a mobile phone's battery. Similar studies have shown the potential for denial of service attacks, identity theft and wiretapping by launching attacks either using compromised phones or using channels such as SMS [17, 3]. Our Browser-choking attack is unique, however, in that it both denies service to the infected profile and exhausts resources on the device an individual is using to view the profile.

Other research on Web 2.0 security. Much prior research on Web 2.0 security has focused heavily on detecting and preventing cross-site scripting and similar attacks. To combat these attacks, it is recommended that user input be properly sanitized at the Web server level before being displayed [23, 27]. However user-input sanitization is challenging to implement correctly, in part because of lax standards in how browsers parse HTML, which lead to worms such as the Samy worm [20] and the Yamanner worm [9]. Both of these worms exploited bugs in sites' input sanitization methods to launch cross-site scripting attacks on MySpace and Yahoo Mail, respectively.

While such attacks have forced several social networking sites to blacklist certain HTML tags, such as `<script>`, our work shows that HTML tags hitherto considered benign (such as `<img>`) can also be used to launch attacks. These attacks are made possible in part by the scale of social networking sites and the fact that arbitrary Internet users can post multimedia content on other users' highly trafficked pages.

7. CONCLUSIONS

Social networking Web sites currently offer too much freedom to their users. The bar for entry into a social network is relatively low, as is the effort needed to form social relationships. Many social networks also allow their users to post multimedia content on other users' profiles by using HTML tags.

This paper shows that HTML tags that were hitherto considered benign can be used maliciously to launch distributed denial of service attacks, as a channel to deliver command and control to bot-infected computers, and to cause denial of service and extremely high memory usage on resource constrained browsing devices. While these attacks themselves can be detected and prevented reactively using previously-developed techniques, we argue that the attacks result from a fundamental flaw in the design of social networking Web sites—too much freedom is given to Web users on the profiles and pages of other, much more popular, users. We therefore conclude that social networking Web sites must employ techniques to restrict content posted by arbitrary Web users on the profiles of other users in order to mitigate the possibility of the attacks discussed in this paper.

8. REFERENCES

[1] Phatbot trojan analysis. `http://www.lurhq.com/phatbot.html`.

[2] Sinit p2p trojan analysis. `http://www.lurhq/sinit.html`.

[3] N. Agarwal, L. Chandran-Wadia, and V. Apte. Capacity analysis of the gsm short message service. In *National Conference on Communications*, 2004.

[4] 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.

[5] A-L. Barbasi. *Linked: The New Science of Networks*. Persius publishing, 2002.

[6] Paul Barford and Vinod Yegneswaran. An inside look at botnets. 2006.

[7] D. Boyd. Friends, friendsters, and myspace top 8: Writing community into being on social network sites. *First Monday 11(12).*

[8] P. Cashmore. Myspace layouts top 10. *Mashable*, July 28, 2006.

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

[10] 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.

[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. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *Networked and Distributed System Security Symposium*, February 2006.

[13] Flickr frequently asked questions. http://www.flickr.com/help/faq/.

[14] S. Gianvecchio, M. Xie, Z. Wu, and H. Wang. Measurement and classification of humans and bots in internet chat. In *USENIX Security Symposium*, August 2008.

[15] 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.

[16] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: Detecting mwlare infection through ids-driven dialog correlation. In *USENIX Security Symposium*, August 2007.

[17] C. Guo, H. J. Wang, and W. Zhu. Smart phone attacks and defenses. In *Workshop on Hot topics in Networking*, Nov 2004.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

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

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

[25] 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.

[26] Radmilo Racic, Denys Ma, and Hao Chen. Exploiting MMS vulnerabilities to stealthily exhaust mobile phone's battery. In *International Conference on Security and Privacy in Communication Networks*, August 2006.

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

[28] Kapil Singh, Abhinav Srivastava, Jonathon Giffin, and Wenke Lee. Evaluating email's feasibility for botnet command and control. In *International Conference on Dependable Systems and Networks*, June 2008.

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

[30] Ryan Vogt, John Aycock, and Michael J. Jacobson Jr. Army of botnets. In *Networked and Distributed System Security Symposium*, February 2008.

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