Detecting and Characterizing Social Spam Campaigns

Hongyu Gao
Northwestern University
Evanston, IL, USA
hygao@u.northwestern.edu

Jun Hu
Huazhong Univ. of Sci. & Tech
and Northwestern University
junehu1210@gmail.com

Christo Wilson
U. C. Santa Barbara
Santa Barbara, CA USA
bowlin@cs.ucsb.edu

Zhichun Li
Northwestern University
Evanston, IL, USA
lizc@cs.northwestern.edu

Yan Chen
Northwestern University
Evanston, IL, USA
ychen@northwestern.edu

Ben Y. Zhao
U. C. Santa Barbara
Santa Barbara, CA USA
ravenben@cs.ucsb.edu

ABSTRACT

Online social networks (OSNs) are popular collaboration and communication tools for millions of users and their friends. Unfortunately, in the wrong hands, they are also effective tools for executing spam campaigns and spreading malware. Intuitively, a user is more likely to respond to a message from a Facebook friend than from a stranger, thus making social spam a more effective distribution mechanism than traditional email. In fact, existing evidence shows malicious entities are already attempting to compromise OSN account credentials to support these "high-return" spam campaigns.

In this paper, we present an initial study to quantify and characterize spam campaigns launched using accounts on online social networks. We study a large anonymized dataset of asynchronous "wall" messages between Facebook users. We analyze all wall messages received by roughly 3.5 million Facebook users (more than 187 million messages in all), and use a set of automated techniques to detect and characterize coordinated spam campaigns. Our system detected roughly 200,000 malicious wall posts with embedded URLs, originating from more than 57,000 user accounts. We find that more than 70% of all malicious wall posts advertise phishing sites. We also study the characteristics of malicious accounts, and see that more than 97% are compromised accounts, rather than "fake" accounts created solely for the purpose of spamming. Finally, we observe that, when adjusted to the local time of the sender, spamming dominates actual wall post activity in the early morning hours, when normal users are asleep.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and behavioral sciences

General Terms

Human Factors, Measurement, Security

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IMC'10, November 1–3, 2010, Melbourne, Australia.
Copyright 2010 ACM 978-1-4503-0057-5/10/11 ...$10.00.

Keywords

Online social networks, Spam, Spam Campaigns

1. INTRODUCTION

Online social networks (OSNs) are popular collaboration and communication tools for millions of Internet users. For example, Facebook alone boasts over 500 million users, and has recently surpassed Google as the most visited site on the Internet [22]. As communities built out of friends, family, and acquaintances, the public perception of OSNs is that they provide a more secure environment for online communication, free from the threats prevalent on the rest of the Internet. In fact, a study of a social auction site demonstrated that a social network could indeed provide a protective environment with significantly lower levels of fraud [38].

Unfortunately, recent evidence shows that these trusted communities can become effective mechanisms for spreading malware and phishing attacks. Popular OSNs are increasingly becoming the target of phishing attacks launched from large botnets [8, 10], and OSN account credentials are already being sold online in underground forums [39]. Using compromised or fake accounts, attackers can turn the trusted OSN environment against its users by masquerading spam messages as communications from friends and family members.

In this paper, we present a first of its kind study to measure and analyze attempts to spread malicious content on OSNs. Our work is based on a large dataset of "wall" messages from Facebook. Wall posts are the primary form of communication on Facebook, where a user can leave messages on the public profile of a friend. Wall messages remain on a user's profile unless explicitly removed by the owner. As such, wall messages are the intuitive place to look for attempts to spread malicious content on Facebook since the messages are persistent and public, *i.e.* likely to be viewed by the target user and potentially the target's friends. Through crawls of several Facebook regional networks conducted in 2009, we obtained a large anonymized dataset of Facebook users, their friendship relationships, and 1.5 year-long histories of wall posts for each user [41]. In total, our dataset contains over 187 million wall posts received by 3.5 million users.

Our study of Facebook wall posts contains two key phases. First, we analyze all wall messages and use a number of complementary techniques to identify attempts to spread malicious content (Section 3). We focus our analysis on messages that contain URLs or web addresses in text form. From these messages, we produce correlated subsets of wall posts. We model each post as a node, and create edges connecting any two nodes referring to the same URL, or any two nodes sharing similar text content as defined by

an approximate textual fingerprint. This process creates a number of connected subgraphs that partition all suspect wall messages into mutually exclusive subsets, where messages in a set are potentially related. Using dual behavioral hints of bursty activity and distributed communication, we can identify subsets of messages that exhibit properties of malicious spam campaigns. We use several complementary mechanisms to validate the effectiveness of our technique, and show that our approach is highly effective at detecting the spread of malicious content (Section 4).

In our second phase, we analyze the characteristics of the malicious wall posts we have identified (Section 5). Our results provide several interesting observations on the spread of malicious content in OSNs, and the behavior of users that spread it. We find that phishing is by far the most popular attack on Facebook. We also find that users who spread malicious content communicate using very different patterns compared to the average user, and that malicious users stand out by both the bursty nature of their wall posts, as well as their diurnal activity patterns. By studying the time-duration of malicious messages and the lifetimes of users that send them, we conclude that the overwhelming majority of spam messages are sent through compromised accounts, rather than fake accounts specifically created for spam delivery. Finally, we study the largest observed spam campaigns, and make observations about their attack goals and sales pitch.

In summary, we present in this project the first attempt to quantify the prevalence of malicious accounts and spread of malicious content on an OSN. We employ multiple techniques to detect correlation between wall messages and to identify the spread of potentially malicious content. Our results are confirmed by a number of validation mechanisms. Our subsequent analysis provides insights into the operation of malicious accounts, and has significant implications on the design of future mechanisms to detect malicious behavior on OSNs.

2. BACKGROUND

In this section, we provide background information about the Facebook OSN, introduce the dataset used in our work, and clarify the scope of this work before we begin analysis of our data.

2.1 Facebook

Facebook is the most popular OSN in the world, boasting a population of over 500 million users. It is the largest photo hosting service on the web [23], and has recently surpassed Google as the most visited site on the Internet [22].

Like most OSNs, Facebook encourages users to create profiles that contain rich information about themselves, including their name, photo, address, occupation, interests. Users create undirected friendship links with their family and acquaintances in order to stay connected. The most popular ways for users to interact on Facebook are posting status updates to their profile and writing on their friends' "walls." In turn, each of these events can be commented on by other users. All status updates, wall posts, and comments are tagged with the sender, the recipient, and a timestamp.

Facebook includes a set of API's for developers to write applications that can plug in to user's profiles. Applications, especially games, are quite popular among Facebook users. Once installed, applications can generate update messages that appear in a user's profile alongside status updates and wall posts from friends.

Originally, users' default privacy settings on Facebook were tied to the concept of "networks." Each network represented a school, company, or geographic region. Joining school and company networks required providing authentication details in the form of a EDU or company email address. Access to regional networks was unauthenticated. By default, users in the same network could view each other's information. In late 2009, after we completed our measurements, Facebook deprecated the use of networks in their system.

2.2 Crawled Facebook Data

Between April and June of 2009 we crawled Facebook for data. Because access to regional networks is unauthenticated, we chose 8 regional networks of various sizes (from over 1.6 million users down to $\sim$14K users) as targets for data collection. These networks are Egypt, Los Angeles, London, Monterey Bay (California), New York City, Russia, Santa Barbara (California), and Sweden. Each crawl was seeded with 50 random users from the given regional network, and includes a breadth-first search of all users in the region with visible profiles (*i.e.* users with the default privacy settings). We acknowledge that there is about one year time gap between the data collection and our analysis. This time gap poses additional difficulty on our analysis process. We leverage a stringent set of heuristic tests (Section 4) to overcome this difficulty.

For each crawled user we recorded their anonymized userID, friend list, and interaction records going back to January 1, 2008. Interaction records include the complete history of wall posts received by each crawled user within the given time frame. Each crawled interaction record is associated with the timestamp when the event occurs, adjusted to the local time of the crawling machine (Pacific Daylight Time). In this study, we focus on the 187M wall posts crawled from roughly 3.5 million users. Table 1 summarizes the dataset characteristics.

For the purposes of this study, we are interested in studying wall posts that could potentially be spam. This means isolating the set of wall posts that are: i) generated by users, not third-party applications, and ii) include embedded URLs to external websites. Note that we do not limit ourselves to URLs in the form of hypertext links. We also handle wall posts with "hidden" URLs, *i.e.* URLs in plain text or even obfuscated form.

2.3 Scope of This Work

A wide range of attacks exists in today's OSNs. We do not attempt to address all of them in this work. Our focus is solely on detecting and measuring large-scale *spam campaigns* transmitted via Facebook users' *wall messages*. Although spam traditionally refers to massive, unsolicited campaigns to sell goods over email, we do not restrict ourselves to this behavior alone. Rather, we identify and measure multiple types of attacks that are executed via spam wall posts, including but not restricted to: i) *product advertisements*, ii) *phishing attacks* and iii) *drive-by-download attacks*. Although the purpose of each attack varies, they share one common feature: attackers create or compromise a large number of Facebook accounts and use them to spam wall posts to a even larger set of users. The wall posts left by attackers each contain a potentially obfuscated URL and text to convince the recipient to visit the URL. If a recipient is deceived and visits the URL, she will be led to a malicious website associated with the spam campaign. Throughout this paper, we refer to a wall post as "malicious" if it belongs to a spam campaign. In addition, we refer to an account as "malicious" if it has made at least one malicious wall post.

3. SPAM CAMPAIGN DETECTION

To identify malicious wall posts from our large dataset, we use semantic similarity metrics to identify mutually exclusive groups within the total set of wall posts. We then use behavioral cues to identify distinct wall post groups as benign or potentially malicious. In this section, we describe the multi-step process through

which we organize, cluster, and identify potentially malicious posts. We begin with an overview of the entire process, followed by a detailed description of each step.

3.1 Overview

The design of our system is guided by intuition about techniques used in spam campaigns to maximize reach and avoid detection. Spammers generate profit from these visits by selling products or services, performing phishing attacks, or installing malware onto victim's machines. Therefore, we assume each spam campaign is focused on convincing a maximum number of users to visit some particular URL(s).

To make a spam campaign effective, spammers are likely to a) customize individual messages towards the targeted user, and b) attempt to avoid detection by hiding the destination URL through obfuscation. Thus, it is possible for messages within the same campaign to look significantly different. This "diversity" makes detection of spam campaigns challenging, since neither the textual description nor the destination URL, or even their combination, can be used as an effective signature to detect a single campaign.

Our intuition is to use complementary techniques to overcome these hurdles, with the goal of grouping messages from the same spam campaign together into clusters. Note that we do not aim to completely aggregate spam wall posts from one campaign into one single group, as this level of precision is unnecessary. First, to overcome *user-customization* techniques, we refer to recent work that shows spamming botnets use templates to generate customized email spam messages [29]. We also observe that there are a large number of malicious posts in our dataset that look similar to each other. From this we hypothesize that spam wall posts are also generated using templates, and posts generated from the same template should contain only small differences. Thus, we propose to group wall posts with "similar" textual description together, where similarity is captured by a probabilistic fingerprint. This probabilistic fingerprint must be more efficient to compute than edit distance, yet accurate enough to reflect similarity between text samples despite attempts by attackers to obfuscate or customize the text.

Second, we reason that all attempts to direct OSN users towards a single destination URL must come from the same spam campaign. Thus, we group together all wall posts that include URLs to the same destination, including those that have been hidden through textual obfuscation (*e.g.* www dot hack dot com) and chains of HTTP redirects.

In summary, our detection approach focuses on two techniques that group together wall posts that share either the same (possibly obfuscated) destination URL, or strong textual similarity. We model all wall posts as nodes in a large graph, and build edges when two posts are connected by one of the above techniques. Intuitively, the resulting connected subgraphs could represent messages within the same spam campaign. The pairwise comparison among wall posts results in O(n^2) time complexity where n is the number of wall posts. Although this time complexity can be significant for large values of n, this approach actually performs reasonably fast in practice. Processing our entire dataset with interactions dating back to January 1, 2008 took a total of 2 days of computation time on a commodity server. While the worst case space complexity is also O(n^2), the graph is very sparse in practice, *i.e.* most wall posts are non-malicious, and thus distinct. One of today's modestly configured servers has sufficient memory (4GB) to handle such computations.

After constructing a clustered graph of wall posts, we leverage two additional assumptions about spam campaigns to separate malicious, spam clusters from the large majority of benign wall posts. These assumptions are: a) any single account is limited in the number of wall posts it can post, thus spammers must leverage a significant number of user accounts for large campaigns, and b) spam campaigns must maximize time efficiency of compromised or fake accounts before detection, thus messages in a single campaign are relatively bursty in time. We apply threshold filters based on the number of user accounts sending wall posts and time correlation within each subgraph to distinguish potentially malicious clusters from benign ones. An overview of the system workflow is shown in Figure 1.

3.2 Modeling and Clustering Wall Posts

We model each wall post as a <*description*, *URL*> pair. The *URL* is the destination the spammer wants the target to visit. The URL may be either of legitimate format or obfuscated. The *description* is the text in the wall post surrounding the URL used to convince the target to visit the URL. We build such a model because *description* and *URL* are the *only* information used in the detection phase. Other information, like the sender's and receiver's unique Facebook IDs, are involved in other phases of this study. It is possible for the description to be an empty string, in which case the wall post contains only an URL. Clearly, while most benign wall posts do not contain any URLs, not all wall posts with URLs are malicious.

Since any wall post without a URL cannot achieve a spammer's goals, we begin by first excluding all wall posts without embedded URLs from further analysis. Next, we use our two assumptions (described above) to connect any two wall posts together if they: point to the same destination URL, or share an approximately similar text description. Thus we define two wall posts as *similar* if they share the same URL *or* similar description. Accordingly, the wall post *similarity graph* is an undirected graph $G =< V, E >$, where each node $v \in V$ represents one wall post, and two nodes u and v are connected with an edge e_{uv} if and only if they are similar.

Building the Wall Post Similarity Graph. Before clustering wall posts into a graph, we first identify wall posts containing URLs and extract their URLs. Locating and extracting well formed, properly marked up hyperlinks from wall posts is a simple process. However, it is non-trivial to recover obfuscated URLs that have been hidden inside plaintext. To detect obfuscated URLs, we first use keyword searches, *e.g.* "click here," to detect the starting location of a potential URL. Next, we scan along the text and remove any characters that are not legal in HTTP URLs (*e.g.* whitespace, *etc.*). We also reverse common techniques used by the spammers to obfuscate URLs, *e.g.* replacing " dot " with ".", during the scan. This reconstruction process continues until we either successfully rebuild the URL, or determine that this chunk of text cannot be a legitimate URL. Locating and recovering obfuscated URLs is only done once in the preprocessing stage and is not repeated when building the wall post similarity graph. Following reconstruction, two URLs are considered the same if they match, ignoring HTTP URL parameters.

To find approximate similarity matches between text descriptions, we compute a fingerprint for each block of the description text. We treat the description as a single string, and compute 128-bit MD5 hashes for each 10-byte long substring. We then sort these hash values and choose the 20 numerically smallest values as our approximate fingerprint. Two descriptions are similar if and only if at least 19 of their fingerprints match. This approach matches descriptions even after some substrings have been modified, and has been shown to be successful against adversarial spammers in our prior work [44]. We experimented with our dataset to determine that 19/20 was an appropriate threshold that yielded the best trade-

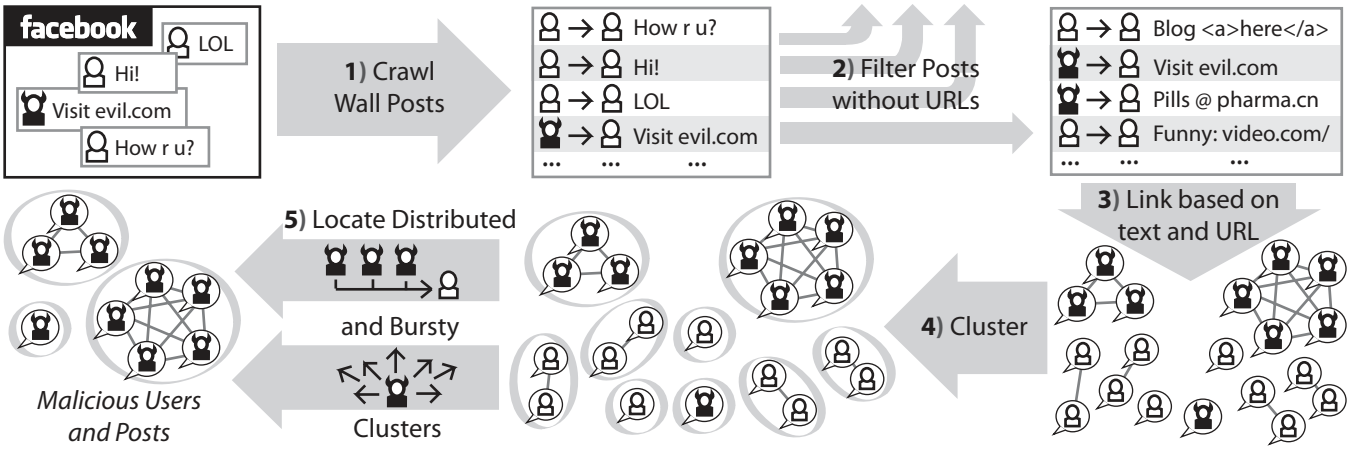

Figure 1: The system design, starting with raw data collection and ending with accurate classification of malicious posts and the users that generating them.

Algorithm 1 **PostSimilarityGraphClustering**($G < V, E >$)

```
traversed ← ∅
clusters ← ∅
Foreach v ∈ V
  If v ∈ traversed
    continue
  EndIf
  one_cluster ← BFS(v)
  traversed ← traversed ∪ one_cluster
  clusters ← clusters ∪ {one_cluster}
EndForeach
return clusters
```

off between robustness against obfuscation while avoiding false positives. Additionally, a relatively high absolute threshold implicitly filters out strings shorter than 19 characters, which is useful since it is common for very short text descriptions to have matching content.

Now the problem of identifying spam campaigns reduces to a problem of identifying connected subgraphs inside the similarity graph. Each connected subgraph is equivalent to one component of a potential spam campaign. Identifying connected subgraphs is easily solved by iteratively choosing arbitrary nodes and identifying their transitive closures. We summarize the implementation in Algorithm 1.

As an optimization, we could first cluster together wall posts that share the same URL, then do a pair-wise comparison between the description of wall posts in different clusters. If two wall posts share similar description, their corresponding clusters are merged and the comparison between the remaining wall posts within these two clusters are skipped. This optimization eliminates the redundant computation to compare the descriptions of two wall posts that are already within one cluster, although the total time complexity remains O(n^2).

3.3 Identifying Spam Clusters

Now that we have organized all wall posts into clusters that could represent coordinated spam campaigns, the next step is to identify which clusters are likely to represent the results of active spam campaigns.

To detect spam clusters, we use two widely acknowledged distinguishing features of spam campaigns: their "distributed" coverage and "bursty" nature. The "distributed" property is quantified using the number of users that send wall posts in the cluster. In email spam campaigns, the "distributed" property is usually quantified using the number of IP addresses or ASes of the senders [42]. The analogous identifier in OSNs is each user's unique ID. The "bursty" property is based on the intuition that most spam campaigns involve coordinated action by many accounts within short periods of time [42]. We characterize each cluster of messages by measuring the absolute time interval between consecutive wall post events (using timestamps associated with each wall post), and extracting the median value of all such intervals. The median interval characterizes how fast attackers are generating the wall posts, and is robust to outlier values.

While we use both assumptions of spam's distributed and bursty nature to detect spam campaigns, it is possible that social cascades in the social network might produce similar message clusters. For example, a URL to buy tickets to a highly anticipated concert might be widely propagated throughout Facebook by friends. However, recent studies of social cascades have shown that cascades take significant time to propagate [19], suggesting that false positives from social cascades would be filtered out by our temporal burstiness filter.

Now that we are using the "distributed" and "bursty" properties to identify malicious clusters, we face the problem of identifying the best cutoff threshold value. We can maximize the number of malicious clusters we identify by relaxing the threshold value, but that generally results in the system producing more false alarms on benign clusters. In contrast, using more restrictive threshold can reduce the number of false alarms, but may fail to identify malicious cluster (false negatives).

To solve this dilemma, we first ask ourselves, how many false positives are we willing to tolerate in order to gain one more true positive. Since our system is designed for postmortem analysis, and not detection in real time, we choose this value to be 2. Our goal is to locate as many spam campaigns as possible, and we do not need to guarantee zero false positives, since they can be filtered out using a follow-up validation phase. Instead, we can tolerate a moderate number of false positives at the benefit of reducing false negatives, *i.e.* , the malicious clusters that are missed by the detection process. For example, we can set our thresholds to initial values of (4, 6hr), where 4 is the lower bound of the "distributed" property, and 6 hours is the upper bound of the "bursty" property. If the spammer is

Dataset	Wall Posts	Distinct Senders	Distinct Receivers
All Posts	187.17M	23.73M	3.46M
With URLs	2.08M	1.08M	0.83M

Table 1: Overall statistics of our Facebook dataset.

using less than 4 accounts to send spam with an interval greater than 6 hours, he is likely to generate negligible impact on the system.

In practice, we test different possible threshold values to determine values that maximize our utility as we defined earlier. We present our detailed experimental results in Section 4.3. Our process found that the best threshold values are (5, 1.5hr), but also that a number of possible threshold combinations can work well.

3.4 Dataset and Detection Results

Before we describe our efforts to validate our methodology in detail, we first briefly summarize the outcome of our clustering and classification approach. Table 1 summarizes statistics of our initial wall posts dataset, and lists the total number of wall posts, distinct users that send posts, and distinct users who received wall posts. The first row corresponds to the entire crawled dataset. The second row is restricted to the subset of wall posts that include embedded URLs, which form the basis of our study.

Applying our clustering approach to our corpus of 2.08 million wall posts produces 1,402,028 clusters. As expected, there is a heavy tail in the size of clusters, *i.e.* a small number of very large clusters and a large number of very small clusters. When we apply our chosen detection threshold, which uses 5 as the minimum number of users involved in each cluster and 5400 seconds (1.5 hours) as the maximum median interval between the timestamp of two consecutive wall posts, we produce 297 clusters that are classified as potentially malicious spam campaigns. There are a total of 212,863 wall posts contained in these 297 clusters.

4. EXPERIMENTAL VALIDATION

We have described our methodology for identifying malicious wall posts and users from the general Facebook population. In this section, we delve more deeply into the results of our detection techniques, in an effort to verify that the posts we identified were indeed malicious. We accomplish this using a combination of techniques and tools from both the research community and the commercial sector. These techniques provide independent, third-party verification of our results.

We apply a stringent set of heuristic tests to each URL that has been embedded in one or more wall posts in a potentially malicious cluster. Whether the URL is malicious determines whether the wall posts containing it are malicious. However, none of these techniques are foolproof, since there are no guaranteed foolproof methods to identify malicious content online. In the absence of such tools, we take a best-effort approach to validating our results.

4.1 Validation Methodology

Our validation methodology includes a series of steps, each of which encapsulates a different heuristic or tool and aims to concretely verify some portion of the suspicious wall posts as definitively malicious. Our goal is to investigate the false positive rate of our proposed methodology by reexamining the 212,863 malicious wall posts identified in Section 3.4. Later in Section 4.4, we will try to determine the false negative rate amongst our entire dataset of 2.08 million wall posts.

Step 1: URL De-obfuscation. A common technique among spammers is to obfuscate malicious URLs by adding white spaces and unicode characters into them. This allows the offending message to bypass filters that look for blacklisted URLs by simple string matching. We do observe that a significant number of wall posts on Facebook included obfuscated links of this nature. Since there is no incentive for benign users to obfuscate links in this manner, we mark any wall posts that include such URLs as malicious. We de-obfuscate URLs by reversing the obfuscation process, including removing whitespace padding and canonicalizing URL encoded characters (*e.g.* , "%65%76%69%6C%2E%63%6F%6D" becomes "evil.com").

Step 2: Redirection Analysis. Another common technique used by spammers to evade detection is to hide malicious sites behind chains of redirects [40]. This serves to obfuscate the true destination of a URL (which may be blacklisted) from users and automated filters. Before proceeding with our validation for a suspicious URL, we use an automated web browser script to detect and follow the chain of redirects, if they exist. More specifically, we use the JSSH extension for Mozilla, which allows us to establish a JavaScript shell connection to a running Mozilla process via TCP/IP [27]. This allowed us to control running Mozilla processes while they traversed through chains of HTTP redirects, eventually retrieving the final destination URL.

Step 3: Third-party tools. We leverage multiple third-party tools from the research community and the private sector to assess the malice of URLs in our dataset. We leverage a number of the most popular URL blacklisting services to determine if URLs are malicious, including: McAfee SiteAdvisor [3], Google's Safe Browsing API [1], SURBL [6], URIBL [7], Spamhaus [4], and SquidGuard [5]. We submit each unique URL from our dataset to each service in order to account for discrepancies between the coverage of different blacklists. If any blacklist returns a positive for a given URL, it is classified as malicious.

In addition to URL blacklists, we leverage the Wepawet [9] tool from UC Santa Barbara. Wepawet is a specialized tool that uses machine learning to identify web pages with characteristics associated with drive-by download attacks [34]. URLs receiving a "malicious" rating from Wepawet are immediately classified as malicious for our purposes as well.

The primary challenge of using automated tools for validation is that our Facebook data was collected during the first half of 2009, roughly one year ago. Given the ephemeral nature of malicious websites, this means that many of the URLs in our dataset point to stale destinations that no longer exist. Blacklists periodically purge old records from their databases, which further complicates matters. Whenever possible, we query the Internet archive service [2] for the content of URLs based on the timestamp of the associated wall post.

Step 4: Wall Post Keyword Search. Certain keywords often appear in spam messages sent over email. The spammed wall posts in our dataset are no exception: many of them attempt to sell the usual assortment of shady merchandise. To capture these wall messages, we built a set of well-known keywords that are indicative of spam, such as "viagra," "enlargement pill," and "legal bud." We then performed full-text searches on the suspicious wall posts from the detection result for these strings, and classify the resulting posts as malicious.

Step 5: URL Grouping. In this point in our validation, we have verified that some portions of the wall posts in our dataset are malicious. However, shortcomings will likely have prevented us from performing full validation. For example, some URLs are too old and stale to be verified by blacklists. Similarly, we may have

# of URLs	Common features
1,895	URL signature: www.facebook.com.profile.id.* Wall post content: Invitation to visit a fake Facebook profile
407	URL signature: domain followed by single folder with obfuscated video as name, *e.g. www.nemr.net/pub1icsh0w/* Wall post content: Either "wow video" or "cool video"
511	URL signature: */imageshack.us/img[0-9]{2-3}/[0-9]{3-4}/mcr[a-z]{2-2}[0-9].swf Wall post content: Invitation to find out about a secret admirer or read a disparaging remark written by a personal enemy
296	URL signature: *ring*.blogspot.com Wall post content: Facebook is giving out ring tones for the user's cell phone via the provided URL
317	URL signature: *mcy[sz]*[0-9]{3-11}.com or (multilrng)tn[sz]*[0-9]{2-6}.com Wall post content: Invitation to win a free Playstation 3

Table 2: Examples of URL groups and their shared features.

missed variants of spam keywords (*e.g.* "v14gr4" vs. "viagra") in our full-text search. To expand the coverage of our detection techniques beyond positive results using the prior techniques, we will use a grouping strategy to further detect relationships between posts.

We manually construct groups of URLs that exhibit highly uniform features, which is a strong indicator that the whole group is under the control of a single attacker and is typical of many well-organized spam campaigns. Benign URLs are more random in nature, and highly unlikely to exhibit such clustering. The features we leverage include signatures that characterize the URL group [42] and similarity of textual content of the associated wall posts. For each group, if any of its constituent URLs has been identified as malicious in a previous validation step, then all URLs in the group are classified as malicious. We identify 8 such groups in total and list examples with their features in Table 2.

Step 6: Manual Analysis. Even after all five previous validation steps, a small portion of suspicious wall posts still remain unclassified. Since widespread spam campaigns are likely to be reported and discussed by people on the web, we can manually validate these URLs by searching for them on Google. Because this task is highly time-intensive, we only use this approach on URLs that each appear in at least four hundred wall posts.

4.2 Validation Results

We use the multi-step validation process outlined above to confirm the malice of each wall post identified in our detection process. We assume that all wall posts whose malice cannot be confirmed are false positives. Table 3 summarizes the number of URLs and wall posts confirmed as malicious following each validation step, as well as the number of false positives. We see that our heuristic verification mechanisms are surprisingly successful at confirming the large majority of our detected spam messages. Only a very small portion of URLs (roughly 3.9%) remains unconfirmed.

The bulk of true positive URLs are either blacklisted, redirect to a blacklisted site, or grouped together with other blacklisted URLs/wall posts. In contrast, when viewed in terms of total wall posts, Table 3 shows that obfuscated and manually verified URLs account for a significant proportion of wall posts. This demonstrates that some URLs are spammed a disproportionally large number of times, while other spam campaigns spread their traffic across a long tail of unique URLs.

4.3 Burst and Distribution Thresholds

Our detection mechanism relies heavily on the bursty and distributed nature of spam messages. In order to choose a good cutoff threshold between normal and aberrant behavior, we define a desired tradeoff between false positives and true positives: we are willing to tolerate two false positives if we can gain at least one true positive. We then choose the threshold value with the best utility (Section 3.3). In order to find the optimal threshold, we vary our filtering threshold for the "distributed" property of wall posts from 4 to 10, and for the "bursty" property from 0.5 to 6 hours, varied in increments of 0.5 hour. Every combination of these two properties is tested to examine the resulting false positives and true positives, with the results plotted in Figure 2. The resulting points roughly form a convex curve. We choose a straight line with slope of 0.5, which represents the desired tradeoff, to approach the curve from above. The straight line and the curve intersect at the point representing the threshold (5, 1.5hr). Thus we choose (5, 1.5hr) as the threshold for our detection system. Note that, as Figure 2 shows, a number of alternative threshold value combinations would also have yielded good results.

Reason for Classification	# of URLs	# of Wall Posts
Obfuscated URL	1,003 (6.3%)	45,655 (21.4%)
Blacklisted URL	4,485 (28.0%)	55,957 (26.3%)
Redirects to blacklisted URL	4,473 (27.9%)	29,365 (13.8%)
Contains spam keywords	196 (1.2%)	19,018 (8.9%)
Grouped with malicious URL	5,300 (32.5%)	33,407 (15.7%)
Manual confirmation	27 (<0.1%)	16,380 (7.7%)
Malicious, True Positives	15,484 (96.1%)	199,782 (93.9%)
Benign, False Positives	616 (3.9%)	13,081 (6.1%)

Table 3: Validation results. Each row provides the number of confirmed malicious URLs and wall posts in a given validation step. All URLs that have not been validated after all steps are assumed to be benign.

4.4 False Negative Rate Estimation

In addition to estimating the rate of false positives resulting from our detection methodology, it would also be desirable to characterize the amount of false negatives it can produce, *i.e.* the number of malicious wall posts that go undetected. Unfortunately, computing the false negative rate is not feasible in practice for a number of reasons. Given the sheer size of our dataset (2.08 million wall posts with URLs), none of our validation techniques, third-party or manual, would scale. While we cannot compute a real false negative rate, we can offer some numbers on the effectiveness of some of our mechanisms, as rough indicators of how many potential malicious posts could be missed.

Obfuscated URLs. As mentioned in Section 4.1, we believe that all obfuscated URLs are indicative of malicious activity. Hence, any obfuscated URL that is not included in the detection results should be viewed as a false negative. We searched our entire dataset

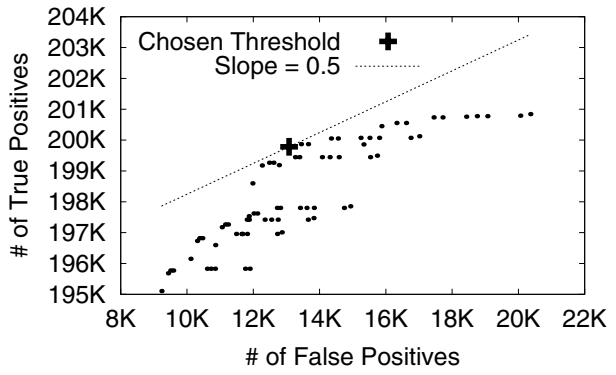

Figure 2: Number of true positives and false positives for each tested combination of threshold values. The dotted line represents our assumed best trade-off between the two (1:2). The highlighted point (cross) represents the threshold we chose for our use.

for obfuscated URLs and identified 1,012 total URLs, of which 1,003 were found in our detection results. These 9 missed URLs are only used in 141 wall posts. In contrast, the 1,003 correctly identified obfuscated URLs appear in 45,655 wall posts. Thus using the pool of obfuscated URLs as ground truth, our detection mechanisms are more than 99% effective.

Blogger Pages. We notice that Blogger pages are frequently used by attackers in our dataset to host malicious content. We took all the *blogspot.com* pages from our full dataset, and applied our validation process to them. This generated a total of 8,162 Blogger pages identified as malicious. Using this set as ground truth, only 406 of these URLs were not detected by our detection mechanism, for a false negative rate of 5%.

5. ANALYSIS OF SPAM ACTIVITY

After going through the detection (Section 3) and the validation (Section 4) steps, we arrive at a set of true positive malicious wall posts. In this section, we analyze the detected malicious users and wall posts to understand their characteristics and their impact on the OSN. Wherever possible, we compare the behavior of malicious accounts using corresponding characteristics of benign users as points for comparison.

Because of the way default privacy settings are configured on Facebook, we have more complete information on users within the crawled regional networks than outside users. Thus we prepare two datasets. The first contains the full set of detected malicious posts and users. We refer to it as the *full set*. The second excludes all malicious users outside the crawled regional network, and all wall posts they generated. We refer to it as the *local set*. The full set contains 199,782 posts from 56,922 users, while the local set includes 37,924 posts from 6,285 users. In the following analyses, we present results from the more appropriate dataset, depending on whether completeness is necessary for the metric.

5.1 URL Characteristics

We categorize malicious URLs by URL format and by domain name, and measure the prevalence of each category. We use the *full set* for this analysis.

Type	# of URLs	# of Wall Posts
Obfuscated	1,003	50,459
Plaintext	583	13,361
Hypertextlink	13,898	135,962

Table 4: Number of malicious URLs of each format type, tabulated over distinct URLs and all wall posts.

5.1.1 Categorized by Format

Throughout this study, we identified three different formats used to embed URLs in wall posts: *hyperlinks*, *plain text* and *obfuscated text*. A hyperlink, *e.g.* http://2url.org/?67592 , is a standard link embedded in the wall posts. It is the easiest format for victims to visit, but is also easily recognized by automated detection techniques. A plain text URL, *e.g.* mynewcrsh.com, is not as easy to use. The victim must copy and paste the URL to the browser's address bar to visit the site. An obfuscated URL, *e.g.* nevasubevu\t. blogs pot\t.\tco\tm (take out spaces), is the most sophisticated. It describes the actual URL to the victim in a human decipherable way. The victim must comprehend and reconstruct the URL by hand before she can visit. It is the most inconvenient to traverse, but is also the most difficult to detect.

Table 4 shows the number of distinct URLs and the number of malicious posts that contain URLs of each format. Normal hyperlinks are the dominant format when counting either the number of distinct URLs or the number of wall posts. They make up 89.8% of all distinct destinations and 68.1% of all malicious posts. However, the other two formats still account for a non-negligible fraction of the overall results. 25.2% of malicious posts contain plain text URLs, while 6.7% contain obfuscated text URLs.

Interestingly, non-hyperlink format URLs are repeated in more total wall posts. On average, each hyperlink URL appears in 9.8 malicious posts. In contrast, an average plain text URL is used in 22.9 malicious posts, while each obfuscated URL is observed on average in 50.3 wall posts.

These results convey two takeaways. First, attackers are willing to embed URLs in a way that is relatively difficult for the target user to visit, potentially for the sole purpose of evading detection. Second, obfuscated URLs are much more likely to be used repeatedly in many wall posts. One cause might be that human effort is required to construct each obfuscated URL, thus reducing the likelihood of mass-producing those URLs.

5.1.2 Categorized by Domain

We extract the domain names of malicious URLs and categorize them into four general types: *content sharing services*, *URL shortening services*, *blogs* and *others*. The first three types demonstrate attackers misusing legitimate services. For content sharing domains, the attacker puts the malicious content into a file and uploads it to the content sharing service, *e.g.* imageshack. For URL shortening domains, the attacker uses the URL shortening service, *e.g.* tinyurl, to hide the real (malicious) destination website. For blog domains, the attacker registers an account on the blog service, *e.g.* blogspot, and uses it to host malicious content. Finally, the *other* category contains domain names that do not appear to have any systematic or contextual similarities.

Table 5 shows the number of distinct domains in each category, as well as the the number of malicious posts that contain domains in each category. The *blog* category dominates in terms of distinct domains. We conjecture that the ease of registering new accounts is the main reason why blogs are popular. However, since all major blog services are administered centrally, it is also easy to identify

Type	# of Domains	# of Wall Posts
URL-short	110	10,041
Blogs	8,609	31,488
ContentShare	440	9,506
Other	6,325	148,747

Table 5: Number of malicious domain names in each group, tabulated over distinct domains and all wall posts.

Campaign	Summary of wall post	Clusters	Posts
Crush	Someone has a crush on you	21	51,082
Ringtone	Get free ringtones	23	31,329
Pharma	Pharmaceutical products like viagra	20	17,614
Narcotics	Sell illegal drugs	11	16,668
Love-calc	Test love compatibility	5	16,354
Macy-gift	Get free Macy's giftcard	4	14,092
Fake-video	Checkout this cool video	114	11,464
Pic-misuse	Someone is misusing your photo	1	10,683
Iphone	Get a free iPhone	4	6,317
Blog	You're mentioned in a blog	2	3,948
Fake-fbid	View a (fake) Facebook profile	1	3,556
Fake-news	View (fake) breaking news	1	2,707
Is-that-you	Is this webpage about you?	4	2,620
IPod-touch	Get a free iTouch	1	2,125
Denigration	Someone is disparaging you	2	1,440
PS3	Get a free PlayStation 3	2	1,131
Webcam	Invitation to video chat	4	1,127
Luxury	Get cheap luxury item	1	981
Online-job	Work online and earn big money	5	502
Others	No common patterns	64	4,042

Table 6: A summary of spam campaigns encountered in our study. Spammers use wall posts to entice the target to visit an embedded malicious URL. Clusters is the number of clusters involved in the campaign. Posts is the number of malicious posts in the campaign.

and remove malicious users and their pages. This potentially explains the high turnover in blog domains: on average, each individual blog is used in only 3.7 malicious posts, while domains in other categories are observed in 20 malicious posts on average.

5.2 Spam Campaign Analysis

A spam "campaign" refers to a set of malicious wall posts that all strive towards a common goal, *e.g.* selling back-door pharmaceuticals. We use the description of each wall post to distinguish between campaigns, without considering the destination URL.

5.2.1 Campaign Identification

To isolate individual spam campaigns out of the full set of malicious posts, we iteratively classify wall posts by identifying characteristic strings in each campaign. Human input is used to aid this process. For example, any wall post containing words like "viagra" or "enlarge pill" are classified into the "pharmaceutical" campaign. If the wall post contains words like "crush on you" or "someone admires you," we classify it into the "crush" campaign. Overall, we identified 19 unique campaigns. They represent 19 common ways that spammers entice targets to visit malicious URLs. Malicious wall posts that cannot be grouped into any campaign are put into an additional "other" group for statistical purposes. We present all campaigns along with a summary of their characteristics in Table 6.

At a high level, three major types of campaigns appear most popular amongst OSN spammers. First, spammers may promise free gifts, *e.g.*, free iPhones, free ringtones, *etc.* Second, spammers may trigger the target's curiosity by saying that someone likes them, is disparaging them, or has written about them on a blog. Finally, spammers may simply describe a product for sale (usually drugs). These three types of campaigns account for roughly 88.2% of all malicious posts.

We associate each campaign with the clusters produced by the detection mechanism. The "fake-video" campaign appears in an exceptionally large number of clusters. The reason is that the description in these wall posts is very short, thus the detection mechanism does not merge the clusters.

For most campaigns, the corresponding clusters form mutually exclusive groups. There are only two instances where one cluster is shared by multiple campaigns. The "crush" campaign shares one cluster with the "love-calc" campaign and the "PS3" campaign. This overlap occurs because both the "crush" and "love-calc" campaigns use a single common URL in the same period of time. The "crush" campaign also shares 400 URLs with the "PS3" campaign. The wall posts containing these shared URLs are naturally grouped into one cluster. This suggests that there is likely a single entity controlling all three of these campaigns.

5.2.2 Phishing and Malware

We turn our focus to spam campaigns that attempt to lure victims to phishing and drive-by download sites. To determine which malicious URLs in our dataset link to sites of these types, we rely on McAfee SiteAdvisor's [3] user review feature, which allows users to collaboratively mark websites as malicious and categorize their misbehavior. We discard reported sites for which the number of benign reports exceeds the number of reports of malicious behavior. If multiple malicious behaviors are reported, *i.e.* phishing and malware propagation, we count all of them.

Our results demonstrate that phishing is an extremely common practice among OSN spammers. Approximately 70.3% of malicious wall posts direct the victim to a phishing site. We encountered two different types of phishing attacks during our investigation. In the first case, spammers target confidential information. Instead of bank account credentials, the OSN account credentials are the primary targets. For instance, the spammer posts wall messages saying Facebook is giving out free ringtones to its users. When the victim visits the provided URL, he is led to a page identical to the Facebook login page and prompted to "log in" again. In the second case, the spammer is after monetary gain. For example, the spammer leaves wall posts asking victims to take a love compatibility test. Clicking the malicious link directs the victim to a site that asks them to provide their cellphone number and agree to a "terms of service" before they can see the results of the love compatibility test. If the victim proceeds, she is automatically signed up for a service and charged a monthly fee.

Malware propagation is the second most common attack associated with malicious URLs in our dataset. About 35.1% of malicious wall posts direct victims to sites laced with malware.

The high percentage of phishing and malware attacks likely stems from the social context of OSNs. It has been shown in prior work [25] that adding a bit of personal information to spam greatly increases the effectiveness of phishing attacks. Intuitively, OSN spam messages appear to come directly from trustworthy friends, and are thus more likely to successfully trick their victims. The same argument holds for malware propagating messages, which effectively turn legitimate users into "bots" controlled by attackers. They are used to send out more phishing and malware propagating wall posts.

5.2.3 Temporal Behavior

We study the temporal characteristics of the identified spam campaigns, and plot the result in Figure 3. The x-axis represents the

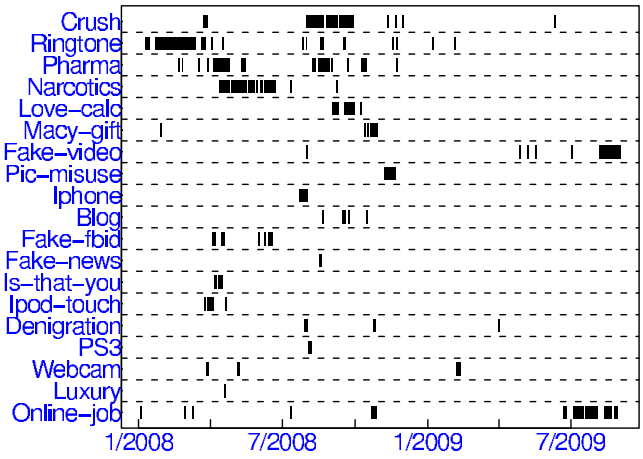

Figure 3: The timeline of each campaign.

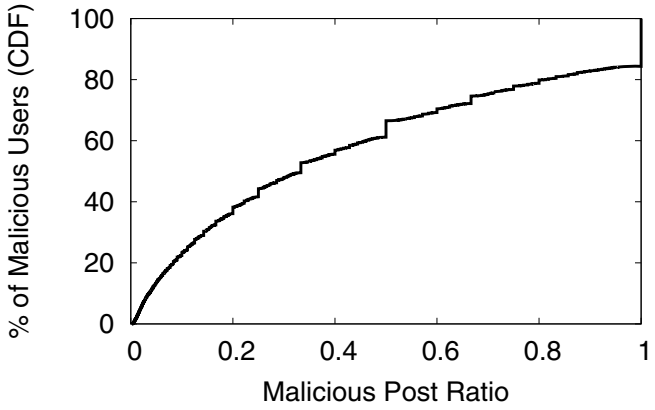

Figure 4: The malicious wall post ratio.

time period between January 1, 2008 and September 1, 2009, which corresponds to the timeframe of wall posts from our dataset. Each spam campaign is represented by a different horizontal strip. A short, thin vertical line within the strip corresponds to one malicious post within the campaign. A block in the strip reflects a burst of messages in the campaign.

Figure 3 shows the bursty nature of all the campaigns. The majority of malicious posts within each campaign are densely packed into a small number of short time bursts, while the entire campaign may span a much longer time period. As stated earlier, URL overlap seems to indicate that the "crush", "love-calc" and "PS3" campaigns are correlated. This is consistent with the observation that active time periods of the "crush" campaign overlaps with the active times of the "love-calc" and "PS3" campaigns.

5.3 Malicious Account Analysis

We now examine characteristics of accounts where malicious wall posts originated. We use our data to study the possible origins of these accounts, their impact on their social circles, temporal characteristics of their wall post activity, and whether malicious activity dominates the accounts they originate from.

5.3.1 Are Malicious Accounts Compromised?

Spammers can obtain their malicious accounts in one of two ways: *compromising* existing accounts and creating *fake* or Sybil accounts [20]. In the first case, the spammer takes control of a legitimate account following a successful phishing or password-cracking attack. This method is attractive, because legitimate accounts already have a significant number of friends (or potential targets). Plus, since these accounts are trusted or at least known to their friends, spam messages sent from these accounts are be more likely to succeed. However, the downside of this approach is that it is a non-trivial task for the spammer to compromise such accounts. Additionally, they may lose control over these accounts at any time, *i.e.* the original owners may detect the account compromise and change the password. Alternatively, spammers may create brand new Sybil accounts. These Sybil accounts are "fake" in the sense that they do not represent a real person. Despite the use of mechanisms like CAPTCHAs, account registration is still relatively easy to automate, and attackers can potentially create a large number of accounts [15]. However, they have to establish friendship with potential victims before they can post spam messages to the victims' walls.

To distinguish between the two types of malicious accounts, we analyze the overall behavior of malicious accounts to determine if they have ever exhibited characteristics of a benign user. In particular, we study each account's *application usage* and number of *received benign wall posts*. Application usage includes uploading photos and video, as well as using third-party social applications. All of these activities are indicative of "normal" account behavior, since it is highly unlikely that attackers would expend time and effort to perform these activities on fake Sybil accounts. We find that 33.9% of malicious accounts exhibit application usage, while 84.5% of accounts received benign wall posts from their friends. Only 11% of malicious accounts do not use applications or receive benign wall posts.

Simply receiving benign wall posts may not be enough to prove that a malicious account is a compromised legitimate account. First, the received posts may be complaints from other accounts. Second, a sophisticated attacker may create Sybil accounts to post messages on each others' walls, making them look like normal user accounts. Unfortunately, it is very difficult to use automateds tool to identify these cases. Instead, we chose uniformly at random a sample of 200 malicious accounts, approximately 5% of the local malicious accounts, and manually inspect the wall posts they received. The key property that we look for is the diversity of the topic and the context of offline events. For example, if the friends are talking about the party last week, the college they go to, *etc.* , it strongly suggests that this conversation involved two real human users. For 5 accounts in the sampled set, users conversed in a foreign language we were unable to reliably translate. For the remaining 195 accounts, we only suspect one account to be a fake Sybil account created by attackers. This user only received complaints and wall posts asking who he is. In addition, the corresponding user account has been suspended. The other 194 (97%) accounts exhibited normal communication patterns with their friends, despite their posting of spam-like content. This sampled result, while not representative, strongly suggests that compromised accounts are the prevailing source of malicious accounts for OSN spammers.

5.3.2 Malicious Post Ratio

We now study the ratio of malicious wall posts to the total number of wall posts made by the detected malicious accounts. This ratio indicates how dominant the malicious posting activity is. A compromised account is expected to exhibit mixed behaviors, *i.e.* they post benign wall messages before the account compromise and (potentially) after the compromise is detected and the account re-secured.

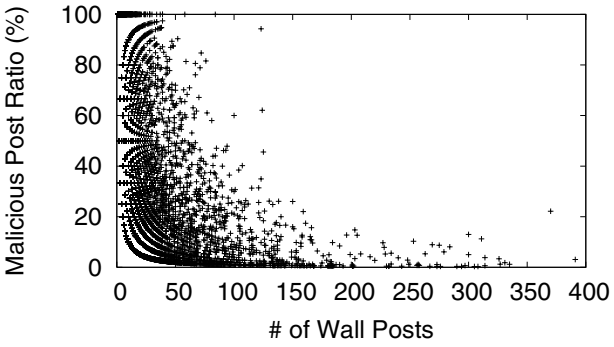

Figure 5: The malicious wall post ratio as a function of number of wall posts made by the user.

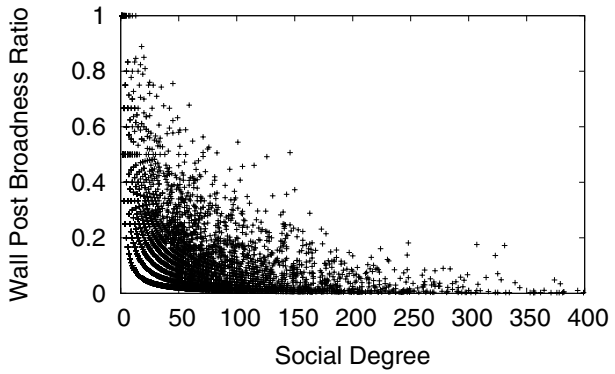

Figure 6: The wall post broadness ratio for malicious users, as a function of their social degree.

Figure 4 plots the CDF of this ratio for all malicious accounts. Less than 20% of the malicious accounts only post malicious wall messages. The remaining 80% of accounts post a mixture of malicious and benign wall posts. Among these accounts, the ratio of malicious wall posts distributes quite evenly. If we make the assumption that fake Sybil accounts would not post non-malicious messages to friends' walls, then this result means that at most 20% of all malicious accounts were created by attackers as Sybil accounts.

Figure 5 considers the malicious post ratio relative to the total number of wall posts. Most malicious users have less than 100 total wall posts. Within this range, the malicious post ratio is distributed relatively evenly. For users with larger number of wall posts, the malicious post ratio decreases. It suggests that attackers might be avoiding excessive malicious wall posts, possibly to avoid detection.

5.3.3 The Impact of Malicious Accounts

We now measure the extent of influence of malicious accounts over their friends, *i.e.* how many of their friends receive spam messages. We quantify this using the "broadness ratio," defined as *the portion of friends that receive wall posts from a user*. Since this measurement requires us to know a user's social degree (total number friends), we use the local data set and extract the social degree from the measured social graph.

Figure 6 plots the broadness ratio of malicious accounts as a function of the user's social degree. Surprisingly, malicious accounts are not posting malicious messages all of their friends' walls. Instead, broadness ratio values are most concentrated around the

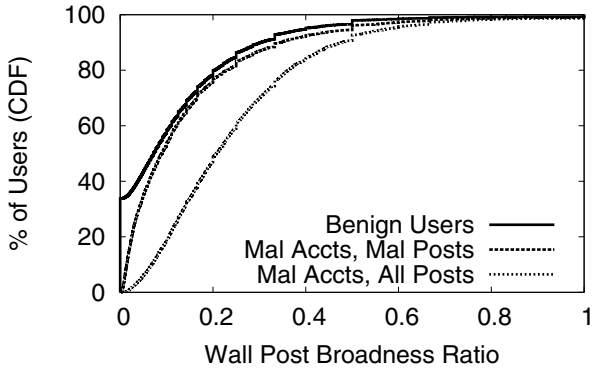

Figure 7: The post broadness ratio of both malicious and benign accounts.

20% range. As social degree increases, the trend is that the broadness ratio decreases. The result is consistent with Figure 5. Note that there are striped, round patterns towards the lower end of the x-axis in both Figures 6 and 5. This is due to the fact that we are dividing two integers to compute the percentage value. For example, the bottom left curve is formed when the numerator is one while the denominator increases. This is a numerical artifact, and represents no underlying trends in the dataset.

Figure 7 shows the CDF of the broadness ratio of both malicious and benign accounts. For malicious accounts, we plot two broadness ratio curves. One curve represents the broadness or coverage of malicious wall posts sent by the account, and the other represents the broadness of all wall posts (malicious and benign) sent by the account. The broadness of benign accounts reflects how broadly each user actually interacts with its friends on Facebook, and the results match prior work from [41] that most users interact with a small portion of their friends. About 60% of benign accounts have broadness value less than 0.1, meaning they interact with less than 10% of their friends. In addition, more than 30% of users have not posted any wall messages.

Overall, malicious accounts tend to interact with a broader portion of their friends than benign users. However, if we only consider malicious wall posts, the results are similar: spammers tend to spam the same distribution of friends as normal users post to their friends. If we consider both malicious and benign posts for malicious accounts, however, the broadness value becomes considerably larger, indicating that the friends that the accounts normally interact with are disparate from the friends receiving the malicious wall posts. Note that the our data only reveals the lower bound of the broadness of malicious accounts, since recipients of spam wall posts will often delete them from their wall, making it impossible for us to find those posts through measurement.

5.4 Temporal Properties of Malicious Activity

We now turn our attention to analyzing the temporal activity patterns of malicious accounts. Specifically, we examine the length of time that accounts are actively under the control of malicious attackers, and the diurnal activity patterns of malicious accounts.

5.4.1 Malicious Account Active Times

We define a malicious account's "active time" to be the time span between the posting time of its first and last *malicious* wall posts. We ignore benign wall posts made by the malicious account when computing its active time. Since most malicious accounts are actually compromised accounts, the active time is also a conservative estimate of the length of time it took for the account owner to detect

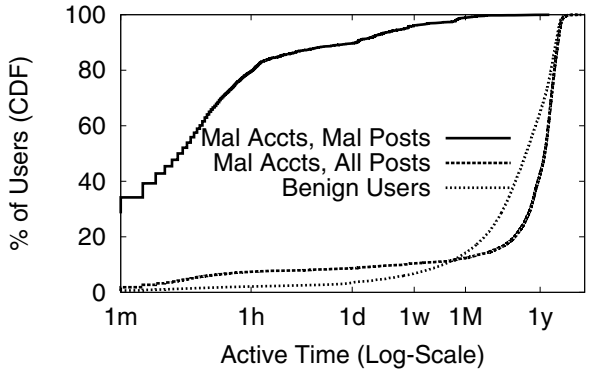

Figure 8: The active time of malicious and benign accounts.

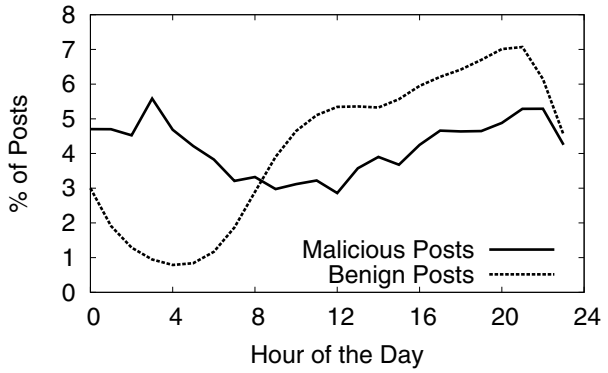

Figure 9: The hourly distribution of malicious and benign wall posts. The time has been adjusted to the local time corresponding to the regional network.

the account compromise and fix the problem. Similarly, we define a benign account's "active time" to be the time span between its first and last wall posts.

Figure 8 plots the CDF of the active time of both malicious accounts and benign accounts. Note that we have again plotted two curves representing malicious accounts. One shows active time as defined above, the other captures the time between the first and last wall posts (benign and malicious) of a malicious account. In all cases, only the accounts that have made at least 2 wall posts are included. To show greater detail on the curve, we plot the x-axis using log-scale, and mark ticks for 1 minute, 1 hour, 1 day, 1 week, 1 month and 1 year.

Clearly, most malicious accounts are actively under control of the attacker for only a short period of time. Roughly 30% of all malicious accounts are active for only a single moment in time, when it posts a single burst of spam to its friends, and never behaves maliciously again. Roughly 80% of the malicious accounts are active for less than 1 hour, and only about 10% of them are active for longer than 1 day. The reason for this phenomenon may be that when an account starts to post malicious wall messages, its friends will immediately complain, and the account owner will quickly realize that their account has been compromised. We observe a considerable number of such complaints in our wall post dataset. Once account owners recognize the attack, they can quickly reclaim control by contacting Facebook and changing the password. Nevertheless, a small portion of malicious accounts continues to post malicious wall messages for a longer period of time (1 month or more), perhaps exploiting accounts of users who rarely log in to Facebook.

In contrast, the benign accounts exhibit a drastically different pattern. More than 80% of benign accounts are active for more than 1 month, and about 35% of them remain active for more than 1 year. However, if we count all wall posts made by the malicious accounts, *i.e.* including benign messages, the curve for malicious accounts shifts drastically and becomes quite similar to the curve for benign users. This is further support for the belief that most malicious accounts are actually compromised legitimate accounts, rather than those created by the attackers.

5.4.2 Diurnal Patterns in Malicious Activity

Finally, we study diurnal activity patterns in the malicious wall posts. We extract the Unix timestamps attached to both malicious and benign wall posts. Since timestamps correspond to the local time where our crawling machines reside, (see Section 2.2), we adjust them to the local time zone based on the location of the regional network. For example, to compute local timestamps of events in the New York City regional network, we add an offset of 3 hours to timestamps to account for the 3-hour time difference between New York City and U. C. Santa Barbara (California). Next, we group the wall posts by the hour of day, *e.g.*, all wall posts that are made between 8am and 9am are grouped together. After that, we aggregate events based on local time, and plot the results in Figure 9. For readability, we normalize the y-value based on the total number of wall posts.

The curve for the benign wall posts clearly reflects the typical diurnal pattern. Few posts are made between 1am and 7am. After the morning, people become active, and more wall posts are made. Wall activity increases steadily over the day, peaking at 9PM and dropping quickly afterwards. However, the hourly distribution of malicious wall posts shows a much different pattern. Malicious wall posts peak at 3am, when most users are asleep and away from their accounts. This is one possible reason for the attackers to pick this time to post messages, so that they avoid immediate detection by the account owners. This suggests that mechanisms for detecting spam activity can prove the most useful in the early morning of each time zone.

6. RELATED WORK

We discuss prior related work by organizing them into three general areas: measurement studies of online social networks, studies of spam on social sites, and design of security and privacy mechanisms for online social networks.

OSN Measurements. A rich corpus of research work lies in measuring and understanding online social networks. Schneider *et al.* use clickstreams to study how users actually use OSNs [36]. Benevenuto *et al.* studied how users interact with friends, like how frequently users visit their friends' pages [18]. They discovered that browsing is actually the dominant behavior on OSNs. Jiang *et al.* studied user browsing behavior in the context of a 42 million user measurement of the Renren social network, through detailed OSN logs of user profile visits [26]. Wilson *et al.* also studied user interactions on OSNs and found that users only interact with small portion of their online friends [41]. There are other works focusing on studying the topological characteristics of OSN graphs [11, 30, 33]. They confirm that OSNs obey approximately power-law scaling characteristics [16], and are small-world networks [12].

Spam Studies. There is a large body of prior work on email spam measurements [13, 28, 29, 42]. However, to our knowledge, little work focuses on understanding the behavior of spam on OSNs. Benevenuto *et al.* propose a supervised learning ap-

proach [17] for detecting spammers in the user feedbacks of Youtube videos. However, due to the difficulty of creating a training set for Facebook, we do not adopt this detection approach. Markines *et al.* proposed to detect spam on social bookmarking sites [32]. Webb *et al.* place honeypot accounts on MySpace and study the captured social spammers [40]. At the beginning of 2010, Facebook launched a new feature to combat unknown friend requests, giving users the ability to reject friend requests as "don't know." Facebook collects this information in order to identify and remove spamming users leveraging fake accounts [21]. It is not yet reported how well this feature works in practice. Finally, Yardi *et al.* proposed to detect spam in the Twitter network [43] using techniques from the traditional email spam detection domain.

OSN Security. Given the overwhelming popularity of OSNs, a significant number of studies have focused on security mechanisms to protect OSN users. A. Felt *et al.* propose a privacy-by-proxy design for third-party OSN applications [24]. They suggest that OSNs hide actual user information from social applications and replace it with tags that represent the information. Other systems, such as xBook [37], require third-party applications to specify how they will use personal information and leverage information flow control to enforce the specification. Persona [14] lets users define fine-grained data access policies; FaceCloak [31] hides personal information from the OSN itself; and StarClique [35] provides strong user privacy guarantees in social applications by modifying the underlying social graph.

7. CONCLUSION

In this paper, we describe our work on detecting and characterizing spam campaigns performed using asynchronous wall messages on the Facebook social network. We analyze a large dataset composed of over 187 million wall posts written to the profile walls of 3.5 million Facebook users. From this dataset, we use automated techniques to group together wall posts that show strong similarities in advertised URL destination or text description. Using clustering techniques based on text similarity and URL destination matching, we build large subgraphs to represent potential social spam campaigns.

Using threshold-based techniques for spam detection, we identify over 200,000 malicious wall posts attributable to 57,000 malicious accounts. Over 70% of these attacks are phishing attacks, and the overwhelming majority of malicious accounts are compromised accounts, not "fake" accounts created for spamming. Our results show that attackers are actively leveraging the "ready-made" friend links of compromised accounts for spam.

To the best of our knowledge, our study is the first to quantify the extent of malicious content and compromised accounts in a large online social network. While we cannot determine how effective these posts are at soliciting user visits and spreading malware, our results clearly show that online social networks are now a major delivery platform targeted for spam and malware delivery. In addition, our work demonstrates that automated detection techniques and heuristics can be successfully used to detect online social spam.

Acknowledgments

We express our sincere thanks to Christian Kreibich and the anonymous reviewers for their valuable feedback. We would also like to give special thanks to Marco Cova, Christopher Kruegel and the Wepawet team at UCSB, without whose help this work would not be possible. The UCSB team is supported by the National Science Foundation under IIS-0916307, CNS-0546216, and IIS-0847925. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] Google safe browsing API. `http://code.google.com/apis/safebrowsing/`.

[2] Internet archive: Wayback machine. `http://www.archive.org/web/web.php`.

[3] McAfee SiteAdvisor. `http://www.siteadvisor.com/`.

[4] The SPAMHAUS domain block list. `http://www.spamhaus.org/dbl/`.

[5] SquidGuard. `http://www.squidguard.org/`.

[6] SURBL. `http://www.surbl.org/`.

[7] URIBL. `http://www.uribl.com/gold.shtml`.

[8] Users of social networking websites face malware and phishing attacks. Symantec.com Blog.

[9] Wepawet (alpha). `http://wepawet.iseclab.org/`.

[10] Zeus botnet targets facebook. `http://blog.appriver.com/2009/10/zeus-botnet-targets-facebook.html`.

[11] AHN, Y.-Y., HAN, S., KWAK, H., MOON, S., AND JEONG, H. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th World Wide Web Conference* (2007).

[12] AMARAL, L., SCALA, A., AND BARTHELEMY, M. Classes of small-world networks. *Proc. of National Academy of Sciences 97*, 21 (2000), 11149–11152.

[13] ANDERSON, D. S., ET AL. Spamscatter: Characterizing internet scam hosting infrastructure. In *Proc. of the USENIX Security Symposium* (August 2007).

[14] BADEN, R., BENDER, A., SPRING, N., BHATTACHARJEE, B., AND STARIN, D. Persona: an online social network with user-defined privacy. In *Proc. of SIGCOMM* (2009).

[15] BAJAJ, V. Spammers pay others to answer security tests. NY Times, April 2010. `http://www.nytimes.com/2010/04/26/technology/26captcha.html`.

[16] BARABASI, A.-L., AND ALBERT, R. Emergence of scaling in random networks. *Science 286*, 5439 (1999), 509–512.

[17] BENEVENUTO, F., RODRIGUES, T., AND ALMEIDA, V. Detecting spammers and content promoters in online video social networks. In *Proc. of SIGIR* (Boston, Massachusetts, USA, July 2009).

[18] BENEVENUTO, F., RODRIGUES, T., CHA, M., AND ALMEIDA, V. Characterizing user behavior in online social networks. In *Proc. of IMC* (2009).

[19] CHA, M., MISLOVE, A., ADAMS, B., AND GUMMADI, K. P. Characterizing social cascades in flickr. In *Proc. of SIGCOMM Workshop on Online Social Networks* (Seattle, WA, August 2008).

[20] DOUCEUR, J. R. The Sybil attack. In *Proc. of IPTPS* (Cambridge, MA, March 2002).

[21] EHRLICH, B. New facebook feature combats dodgy friend requests. http://mashable.com/2010/01/11/facebook-fights-dodgy-friend-requests-with-mark-you-dont-know-feature/.

[22] Facebook traffic tops google for the week. CNN Money.com, March 2010.

[23] FACEBOOK PRESS ROOM. Statistics, 2009. `http://www.facebook.com/press/info.php?statistics`.

[24] FELT, A., AND D., E. Privacy protection for social networking platforms. In *Workshop on Web 2.0 Security and Privacy, Oakland, CA* (May 2008).

[25] JAGATIC, T. N., JOHNSON, N. A., JAKOBSSON, M., AND MENCZER, F. Social phishing. *Communications of the ACM 50*, 10 (2007), 94–100.

[26] JIANG, J., WILSON, C., WANG, X., HUANG, P., SHA, W., DAI, Y., AND ZHAO, B. Y. Understanding latent interactions in online social networks. In *Proc. of the ACM SIGCOMM Internet Measurement Conference* (Melbourne, Australia, November 2010).

[27] JSSh - a TCP/IP javascript shell server for mozilla. `http://www.croczilla.com/bits_and_pieces/jssh/`.

[28] KANICH, C., KREIBICH, C., LEVCHENKO, K., ENRIGHT, B., VOELKER, G. M., PAXSON, V., AND SAVAGE, S. Spamalytics: An empirical analysis of spam marketing conversion. In *Proc. of the ACM Conference on Computer and Communications Security* (October 2008).

[29] KREIBICH, C., KANICH, C., LEVCHENKO, K., ENRIGHT, B., VOELKER, G., PAXSON, V., AND SAVAGE, S. Spamcraft: An inside look at spam campaign orchestration. In *Proc. of LEET* (2009).

[30] KUMAR, R., NOVAK, J., AND TOMKINS, A. Structure and evolution of online social networks. In *Proceedings of KDD* (2006).

[31] LUO, W., XIE, Q., AND HENGARTNER, U. Facecloak: An architecture for user privacy on social networking sites. In *Proc. of IEEE Conference on Privacy, Security, Risk and Trust* (2009).

[32] MARKINES, B., CATTUTO, C., AND MENCZER, F. Social spam detection. In *Proc. of AIRWeb* (2009).

[33] MISLOVE, A., MARCON, M., GUMMADI, K. P., DRUSCHEL, P., AND BHATTACHARJEE, B. Measurement and analysis of online social networks. In *Proc. of ACM SIGCOMM Internet Measurement Conference* (2007).

[34] PROVOS, N., MAVROMMATIS, P., RAJAB, M. A., AND MONROSE, F. All your iframes point to us. In *Proc. of Usenix Security* (San Jose, CA, July 2008).

[35] PUTTASWAMY, K. P. N., SALA, A., AND ZHAO, B. Y. Starclique: Guaranteeing user privacy in social networks against intersection attacks. In *Proc. of ACM CoNEXT* (Rome, Italy, December 2009).

[36] SCHNEIDER, F., FELDMANN, A., KRISHNAMURTHY, B., AND WILLINGER, W. Understanding online social network usage from a network perspective. In *Proc. of the ACM SIGCOMM Internet Measurement Conference* (2009).

[37] SINGH, K., BHOLA, S., AND LEE, W. xbook: Redesigning privacy control in social networking platforms. In *Proc. of USENIX Security* (August 2009).

[38] SWAMYNATHAN, G., WILSON, C., BOE, B., ALMEROTH, K. C., AND ZHAO, B. Y. Do social networks improve e-commerce: a study on social marketplaces. In *Proc. of SIGCOMM Workshop on Online Social Networks* (August 2008).

[39] Verisign: 1.5m facebook accounts for sale in web forum. PC Magazine, April 2010.

[40] WEBB, S., CAVERLEE, J., AND PU, C. Social honeypots: Making friends with a spammer near you. In *Proc. of CEAS* (2008).

[41] WILSON, C., BOE, B., SALA, A., PUTTASWAMY, K. P., AND ZHAO, B. Y. User interactions in social networks and their implications. In *Proceedings of the ACM European conference on Computer systems* (2009).

[42] XIE, Y., YU, F., ACHAN, K., PANIGRAHY, R., HULTEN, G., AND OSIPKOV, I. Spamming botnets: signatures and characteristics. In *Proc. of SIGCOMM* (2008).

[43] YARDI, S., ROMERO, D., SCHOENEBECK, G., AND BOYD, D. Detecting spam in a twitter network. *First Monday 15*, 1 (2010).

[44] ZHOU, F., ZHUANG, L., ZHAO, B. Y., HUANG, L., JOSEPH, A. D., AND KUBIATOWICZ, J. D. Approximate object location and spam filtering on peer-to-peer systems. In *Proc. of Middleware* (Rio de Janeiro, Brazil, June 2003).