An Analysis of Socware Cascades in Online Social Networks

Ting-Kai Huang, Md Sazzadur Rahman, Harsha V. Madhyastha, and Michalis Faloutsos
Univ. of California Riverside
{huangt,rahmanm,harsha,michalis}@cs.ucr.edu

ABSTRACT

Online social networks (OSNs) have become a popular new vector for distributing malware and spam, which we refer to as *socware*. Unlike email spam, which is sent by spammers directly to intended victims, socware *cascades* through OSNs as compromised users spread it to their friends. In this paper, we analyze data from the walls of roughly 3 million Facebook users over five months, with the goal of developing a better understanding of socware cascades.

We study socware cascades to understand: (a) their spatio-temporal properties, (b) the underlying motivations and mechanisms, and (c) the social engineering tricks used to con users. First, we identify an evolving trend in which cascades appear to be throttling their rate of growth to evade detection, and thus, lasting longer. Second, our forensic investigation into the infrastructure that supports these cascades shows that, surprisingly, Facebook seems to be inadvertently enabling most cascades; 44% of cascades are disseminated via Facebook applications. At the same time, we observe large groups of synergistic Facebook apps (more than 144 groups of size 5 or more) that collaborate to support multiple cascades. Lastly, we find that hackers rely on two social engineering tricks in equal measure—luring users with free products and appealing to users' social curiosity—to enable socware cascades. Our findings present several promising avenues towards reducing socware on Facebook, but also highlight associated challenges.

Categories and Subject Descriptors

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

Keywords

Online social networks; Facebook; Social spam

1 Introduction

Motivated by the significant amount of time that users spend on online social networks (OSNs) (e.g., users in the US spend over one-fourth of their time on the Internet on OSNs [1]), spammers have shifted their attention from email to OSNs [5, 16, 7]. Unlike email spam, which is mostly sent by spammers directly to intended victims [1], *socware* [2] leverages the viral propagation on OSNs; every user who falls victim to a socware campaign unknowingly exposes her friends to the same. Thus, socware *cascades* through the social network, and since users receive it with the implicit recommendation of a friend, it is more powerful than email spam.

The goal of our work is to understand the ecosystem that enables socware cascades, specifically on Facebook. Socware is distributed through posts, typically consisting of a short text body and a URL, which appear on users' walls and consequently on news feeds (as we explain in more detail later). As the posts that attempt to achieve a single goal (e.g., redirect users to the same malicious website), cascade through the OSN, hackers can modify the text and/or the URL included in these posts. While the definition of a campaign is relatively well defined in the context of email spam [26, 21, 22, 12, 17, 9], the community's understanding of socware cascades is limited in comparison.

So far, there have been few studies that focus on socware, most of them focus on Twitter, and there has been limited investigation of the underlying enabling ecosystem. Most prior work focuses on detection: identifying spam and malicious posts, and pinpointing accounts used by spammers in Twitter [10, 27]. Whereas recent studies [14, 23] have shown that a large fraction of OSN spam is in fact propagated by legitimate users whose accounts are compromised, e.g., because they installed a malicious Facebook app. Other related and earlier work is discussed in section 6.

Our key contribution is a systematic study of socware cascades on Facebook [3] by analyzing a five month long trace of roughly 100K spam posts identified from the walls of over 3 million Facebook users through 15K users that have installed our application (the monitored *newsfeed* of a profile sees the posts on the wall of all its friends). Our work consists of three thrusts: (a) studying spatio-temporal properties of socware cascades, (b) examining the infrastructure and mechanisms that enable socware, and (c) analyzing the social engineering tricks that socware leverages. We discuss our contributions in each of these directions below.

a. Spatio-temporal characteristics. We find that socware cascades are quite prevalent. Over five months, more than 60% of the monitored users were exposed to at least one cascade. Interestingly, over the duration of our dataset, we observe an evolving trend in which the median cascade appears to be less active, probably with the goal of being less conspicuous and thus evading detection. Indeed, hackers appears to be successful in this goal as cascades later in our dataset last longer than those seen in the initial months. Furthermore, we find that users are more likely to receive socware from their most active friends, whereas the number of friends of

[1] In 2011, 81.2% of spam emails were sent by botnets [8].
[2] We use the new term socware to refer to social malware [4].

[3] We do not claim any contribution towards detecting socware, which we consider part of the input for this paper.

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

a user seems to bear little significance to the chances of the user being exposed to socware.

b. Forensic analysis of cascades. We observe the emergence of AppNets, groups of apps that collaborate in enabling cascades. AppNets seem to play the role that botnets play in the email spam campaigns. Specifically, we find that over 44% of cascades are enabled by Facebook applications, i.e., the posts are made by applications that have been installed by users, who may have been tricked into installing those apps. In quantifying the parasitic symbiosis, we find that over 37% of cascades contain URLs which lead users back to Facebook, either to app installation pages or to corporate pages that want to get a "Like" endorsement. Interestingly, these socware-enabling Facebook apps form colluding groups. Collaboration is prevalent as we find at least 144 tightly-connected application groups of five or more apps enabling the same cascades.

c. Social engineering tricks and intents. We investigate how socware entices and fools users, and we identify two dominant methods: (a) exploiting users' social curiosity (e.g., claiming to show which friends deleted a user or viewed the user's profile), and (b) offering fake free or cool products (e.g., free iPads or free Facebook T-shirts). In our dataset, these two methods account for 46% and 32% respectively of socware cascades seen across at least 10 users. In addition, we observe that over 85% of socware cascades are intended to enable phishing, survey scams, or fraudulent inflation of web page reputation. By contrast, email spam uses significantly different "hooks" (e.g., appeals to sexual curiosity) and has different intents (e.g., sell low cost pharmaceuticals) [8].

Implications for anti-socware efforts. Our findings could provide useful guidelines for reducing socware on Facebook.

First, our results highlight the importance of focusing specifically on the identification of malicious Facebook applications, particularly groups of coordinating applications that enable socware cascades. Moreover, Facebook can greatly reduce socware on its platform simply by better policing of the content that it admits to be hosted under its domain, in the form of pages, events, etc.

Second, our results show that key enablers for socware are users falling prey to scams that offer fake rewards (such as free products) or that appeal to the social curiosity of users. Thus, better education of users remains a significant stumbling block towards eliminating the threat of socware.

2 Background

We present an overview of Facebook terminology and describe the socware dataset that we use.

2.1 Facebook

Facebook is the most popular online social network today, with over a billion users, more than half of whom log in to the site daily [3]. On Facebook, information is shared in the form of *posts*. There are four main types of posts on Facebook—*STATUS*, *LINK*, *PHOTO*, and *VIDEO* [2]. In our analysis, we use the following set of features associated with any post:

- *ID*: a unique Facebook-assigned identifier for the post
- *message*: the text message included in the post
- `Posted-URL` or `P-URL`: A URL included in the post
- *create time*: time at which the post was created
- *application*: application that posted on behalf of the user

Figure 1: Example Facebook post, with its components highlighted.

In addition, a few other fields are included for certain types of posts, e.g., *LINK* posts include a preview of the web page being linked to. Figure 1 presents a breakdown of some of the features associated with an example Facebook post.

When a user clicks on the link included in a post, we refer to the URL of the web page that the user will be led to as the `Landing-URL` of the post. The `Landing-URL` for a post may not always be identical to the `Posted-URL`, e.g., the `Posted-URL` may either be a shortened URL (using a URL shortening service such as *bit.ly*) or there may be a chain of HTTP-based or Javascript-based redirections from the `Posted-URL` to the `Landing-URL`. When a `Posted-URL` is shortened using a URL shortening service, we resolve the first redirect of the `Posted-URL` and refer to the URL obtained as the `Resolved-URL`. We refer to the top two-level domain of the `Resolved-URL` as `Resolved-Domain`.

2.2 Dataset

Detecting socware. We study the characteristics of socware cascades on Facebook by examining a dataset of malicious posts identified by MyPageKeeper [23], our Facebook security application. MyPageKeeper, which has over 15K subscribed users, continually scans the wall and news feed of every subscribed user to detect malicious posts. Since a Facebook user's news feed contains posts made by her friends, the 15K users subscribed to MyPageKeeper enable it to monitor posts seen on the walls of over 3 million Facebook users. The news feed of every MyPageKeeper user contains a subset of the posts made on the walls of the user's friends; the subset of posts from a friend's wall that appear on a user's news feed are selected by Facebook[6].

MyPageKeeper evaluates every URL that it sees on any user's wall or news feed to determine if that URL points to socware. MyPageKeeper classifies a URL as socware if it points to a web page that 1) spreads malware, 2) attempts to "phish" for personal information, 3) requests the user to carry out tasks (e.g., fill out surveys) that profit the owner of the website, 4) promises false rewards, or 5) attempts to entice the user to artificially inflate the reputation of the page (e.g., forcing the user to 'Like' the page to access a false reward). MyPageKeeper evaluates each URL using a machine learning based classifier which leverages the *social context* associated with the URL. For any particular URL, the features used by the classifier are obtained by combining information from all posts (seen across users) containing that URL. Example features used by MyPageKeeper's classifier include the similarity of text message across posts and the number of comments/Likes on those posts. MyPageKeeper has false positive and false negative rates of 0.005% and 3%. For more details about MyPageKeeper's implementation and accuracy,

Dataset	# of posts with URLs	# of malicious posts	# of unique socware URLs
D-Aug	12,436,634	9,718	452
D-Sep	9,295,575	12,777	613
D-Oct	10,103,378	9,070	275
D-Nov	7,974,616	9,851	234
D-Dec	11,373,501	16,045	347
D-Tot	71,861,393	99,935	2,419
D-Land	—	72,702	1,338

Table 1: Summary of datasets.

we refer interested readers to [23]. In this work, we analyze socware identified by MyPageKeeper with the goal of analyzing the characteristics of how socware cascades through Facebook, so as to help improve anti-socware measures such as MyPageKeeper.

Datasets. In this paper, we analyze a dataset of 117 million posts [4] examined by MyPageKeeper over the course of five months (August to December 2011), of which it flagged 99,935 posts as socware. For each post in this dataset, apart from the above mentioned set of features, the dataset also specifies the set of users on whose news feeds this post appeared. We refer to this complete dataset as the *D-Tot* dataset.

Next, in order to analyze malicious activities on a monthly basis—both to identify trends that hold in every month and to identify the evolution of hackers—we partition the *D-Tot* dataset into one dataset for each month. Since the number of users subscribed to MyPageKeeper varies across months, we partition the dataset in such a manner so as to make the datasets across months comparable. To this end, we randomly sample the same number (5K) of users for every month. We then create a dataset for each month as the set of posts collected from the walls and news feeds of the 5K users selected for that month. We thus create five datasets—*D-Aug*, *D-Sep*, *D-Oct*, *D-Nov*, *D-Dec*—corresponding to the five months in our dataset.

During MyPageKeeper's operation, every 12 hours, it crawls the URLs in all posts flagged as socware in the last 12 hours. We limit our crawling frequency due to resource limitations. For every URL crawled, it records the landing URL, its IP address, other *WHOIS* information of the landing domain, and contents of the landing page. It also records the redirection chain of URLs and IPs linking the URL crawled to the landing URL. However, even though MyPageKeeper crawled every URL within 12 hours of it being spotted on Facebook, only 1,338 of the 2,419 unique URLs seen in the *D-Tot* dataset were still active when they were crawled. Our *D-Land* dataset comprises that subset of malicious posts from the *D-Tot* dataset which contain URLs that MyPageKeeper was successfully able to crawl. Table 1 summarizes all of our datasets.

Representativeness. MyPageKeeper monitors posts on the walls of roughly 3M Facebook users —a small fraction of Facebook's billion users. Therefore, it is hard to gauge the representativeness of our results for all of Facebook. Furthermore, the fact that security conscious users are more likely to install MyPageKeeper could introduce potential bias. However, we find that the subset of the social graph comprising MyPageKeeper users and their friends contains 977 connected components, which implies that they do not seem to belong to a closely knit community. Furthermore, since all of our analyses are based on socware observed on the walls of a sufficiently large number of users, we believe that our results have significant takeaways for anti-socware measures (see Section 5). Moreover, where appropriate, we demonstrate the statistical significance of our results by analyzing a subset of our data and showing that the result extrapolates to all of our users.

[4]MyPageKeeper only examines posts that contain URLs (including obfuscated URLs), since posts that do not contain any links typically do not correspond to socware.

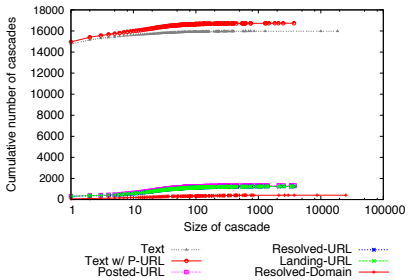


Figure 5: Distribution of cascade size based on different ways of grouping posts into cascades.

3 Spatio-Temporal Properties

In this section, we seek to understand typical characteristics of socware cascades in our dataset: how many cascades are active on any given day? how long do cascades typically last? how are cascades spread across users? While our findings may not be representative for all of Facebook, our analysis here highlights several significant takeaways for anti-socware efforts on Facebook. We present the results from our spatio-temporal analysis of cascades here, and defer a discussion of the takeaways for Section 5.

3.1 Grouping posts into cascades

To analyze socware cascades, we first need to group malicious posts into cascades. Our goal here is to group all posts that lead users to the same landing page. However, as mentioned earlier in the description of our datasets, the landing pages for over 44% of URLs included in malicious posts did not exist 12 hours after they were first monitored on Facebook by MyPageKeeper.

Therefore, in the absence landing page information, we need to group posts based on information readily available in each post, such as the URL and the text message that it includes. To do so, we need to understand how hackers vary or obfuscate the information included in posts of a single campaign as these posts cascades through OSN. For example, Table 2 presents a few example cascades in which the text message included in posts varies across users. On the other hand, some cascades include different shortened URLs, that lead to the same landing page, in different posts.

Here, we systematically examine the effect of different ways of grouping malicious posts into cascades. Our goal is to determine the most effective way that approximates grouping posts with the same `Landing-URL`. We use the features associated with Facebook posts to define six ways of grouping posts into cascades, based on equality in 1) `Text+P-URL`, 2) `Text`, 3) `Posted-URL`, 4) `Resolved-URL`, 5) `Resolved-Domain`, or 6) `Landing-URL`. [5] As an illustration of the significance of choosing the right definition, Figure 5 shows the distribution of cascade sizes (number of posts in each cascade) for different groupings. We see that, depending on how cascades are defined, the average cascade size can vary by as much as 11x.

[5]We defer for future work the use of more sophisticated text comparison techniques for grouping posts into cascades.

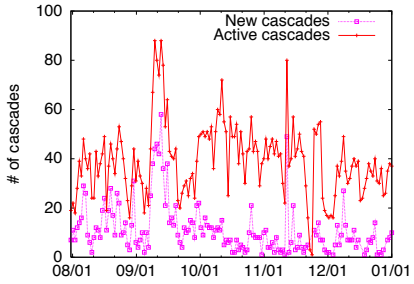


Figure 2: Number of active/new cascades over time.

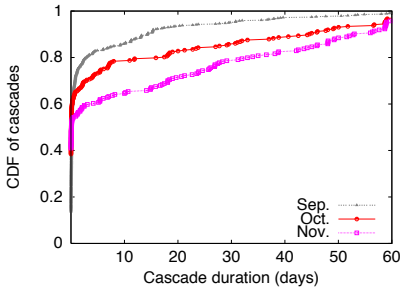


Figure 3: Cascade duration for cascades observed in Sep., Oct., and Nov. 2011.

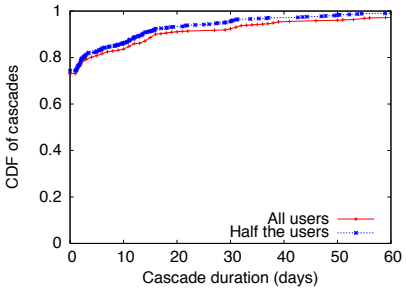


Figure 4: Cascade durations on *D-Sep* with full dataset and with half as many users.

Cascade1	***angela*** *guess what? i just got a new dell xps 1530 laptop to test out and keep without paying anything! i doubt this will last long, dell is only sending out a limited supply of promo laptops in each area so i would hurry. let me know when you get one to.*
Cascade2	***WOW I can't*** *believe that* ***u*** *can check who* ***ls viewing*** *ur profile! I just checked my TOP* ***profile*** *visitors and I am trully SHOCKED at who* ***is*** *seeing my* ***profile****! You can also see* ***WHO*** *VIEWED YOUR PROFILE here.*
	W0W I cant *believe that* ***you*** *can check who* ***is viewing*** *ur profile! I just checked my TOP* ***proffile*** *lookers and I am trully SHOCKED at who* ***ls*** *viewing my* ***proffile*** *You can also see* ***VVHO*** *VIEWED YOUR PROFILE here.*
Cascade3	*wow, got a super free sexy t-shirt from facebook team...grab ur's from here...fb team is giving free t-shirts :d :-*
	*facebook gifting free tee's to all users, just verify ur account & grab the super cool t-shirt! link : d <3 ps : got mine ^_^*

Table 2: Example cascades in which hackers vary text messages across posts.

Definition	Landing-URL
Text	[0.01, 0.29]
Text+P-URL	[0.26, 0.96]
Posted-URL	[0.98, 1.0]
Resolved-URL	[0.95, 1.0]
Resolved-Domain	[0.04, 0.49]

Table 3: Minimum and Maximum ARI (similarity) values between different cascade definitions per month over five months. We compare Landing-URL with other definitions based on posts which contain landing URLs in monthly datasets.

Dataset	cascades with (Active Days = Duration)		cascades with (Active Days < Duration)	
	#	%	#	%
Sep	151	64.26%	84	35.74%
Oct	26	25.00%	78	75.00%
Nov	14	13.46%	90	86.54%

Table 4: Comparison of active days and duration for cascades that last longer than a day.

To analyze which cascade definitions are equivalent to Landing-URL, we consider every cascade definition and compare the set of cascades obtained with that obtained based on Landing-URL. Note that the total number of posts is the same irrespective of how cascades are defined. Hence, the set of cascades obtained with any particular cascade definition can be thought of as a different clustering of the same posts.

To compare the clustering produced by a pair of cascade definitions, we use the *Adjusted Rand Index (ARI)* metric [20]. In our use case, the ARI metric compares two cascade definitions by 1) counting the number of pairs of posts that are in the same cascade in one case but not in the other, and then 2) computing the fraction that these represent out of the total number of pairs. ARI values vary from 0 to 1, and the greater its value the greater the similarity of clusterings being compared. For each cascade definition, we compute the ARI value when compared with Landing-URL cascades on each month's data separately. Table 3 lists the minimum and maximum ARI values across the five months in our data.

First, we find that Posted-URL and Resolved-URL cascades are highly similar to those based on Landing-URL; the ARI value is greater than 0.95 in all months. Note that this high degree of similarity is seen over a dataset which includes over 58 posts on average for every Landing-URL cascade. This indicates that, within each cascade, hackers typically use the same Posted-URL to ultimately redirect users to a particular target landing page.

On the other hand, Text, Text+P-URL, and Resolved-Domain cascades are consistently different from Landing-URL cascades. Cascades based on Resolved-Domain differ because, as we show later in Section 4, a significant fraction of socware is hosted on Facebook itself and many cascades reuse the same hosting providers to host redirectors. On the other hand, we find that Text and Text+P-URL cascades differ from Landing-URL cascades because hackers often vary the text message included in different posts that advertise the same URL. They seem to do so for two reasons: (a) to appeal to users by including users' names in the posts, or (b) to obfuscate the text and evade detection; see examples in Table 2.

Based on these results, for the remainder of this paper, we consider socware cascades obtained by grouping posts with identical posted URLs.

3.2 Temporal properties of cascades

We begin our spatio-temporal study of cascades by analyzing the distribution of the number of active cascades over time. Figure 2 shows that, on any given day, there are typically around 40 socware cascades active in our dataset. On some days, the number of active cascades goes up to as high as 80. Further, we see that typically less than 10 new cascades begin on any given day. Thus, this seems to indicate that, at any point in time, most active cascades are ones that have been active for a while.

To confirm the presence of long-lasting cascades, we next examine the durations of cascades. To avoid edge effects, we perform this analysis as follows. We consider our *D-Sep*, *D-Oct*, and *D-Nov* datasets, and in each case, we consider the cascades active at some point during the respective month. We then compute the duration for a cascade found in a particular month's dataset by examining the posts for that cascade across the previous month, the month considered, and the subsequent month. As a result, a cascade's duration can be at most 3 months in this analysis.

Most cascades are short-lived. For each of September, October, and November 2011, Figure 3 shows the distribution of duration for cascades active during that month. In all three months, we find that most cascades are short-lived; 60% of

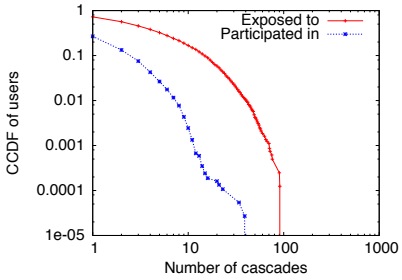


Figure 7: Distribution of no. of cascades that users are exposed to or participated in.

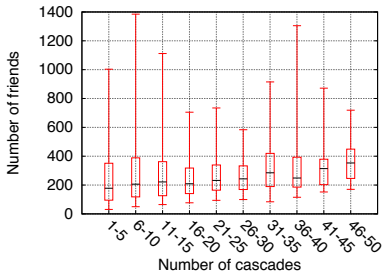


Figure 8: No. of friends versus the no. of cascades that a user was exposed to.

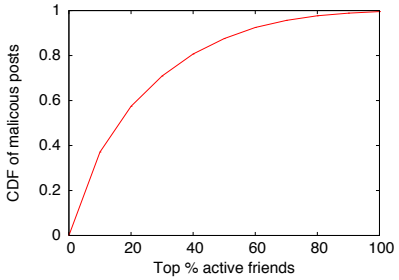


Figure 9: Fraction of malicious posts by top x% of friends, ranked by no. of posts.

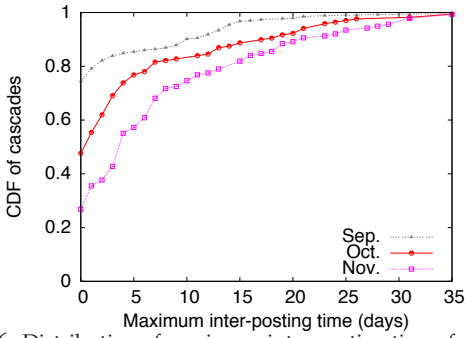


Figure 6: Distribution of maximum inter posting times for cascades observed in Sep., Oct., and Nov. 2011.

cascades last shorter than 1 week. However, there is a long tail of long-lived cascades in all three months; 5%–22% of cascades last longer than a month. It is due to these few long-lived cascades that we previously observed the number of new cascades on any given day to be fewer than the total number of cascades active on that day.

Since MyPageKeeper monitors posts on the walls of a subset of Facebook users, we are potentially under-estimating cascade durations. To evaluate the robustness of our inferred durations, we examine the cascade durations that we would have inferred if MyPageKeeper had half the users that it does have. From the users selected in our *D-Sep* dataset, we pick 10 different randomly chosen subsets with half the number of users. For every cascade in the *D-Sep* dataset, we compute its duration across the posts in each randomly chosen subset and compute the average duration across subsets. Figure 4 compares the distribution of cascade durations on our original dataset with that obtained using half the users as described above. We see that the distributions would largely be the same even if MyPageKeeper had half as many users.

Cascades are becoming less active and lasting longer. Figure 3 also shows that newer cascades appear to be lasting longer than older ones. The long tail of long-lived cascades increases significantly from September to November.

To understand the cause for the increasing cascade lifetimes, we examine the activity of cascades in these three months. For each cascade, we count the number of days during its lifetime when malicious posts belonging to this cascade appear in our dataset. We refer to this as the number of active days of a cascade. We consider cascades that last longer than a day and partition them into two sets—1) those which are active every day during their lifetime, and 2) the remaining that have number of active days less than the cascade's duration. Table 4 presents the breakdown of cascades into these two partitions for September, October, and November 2011. We see that the fraction of cascades that are active throughout their lifetime significantly reduces from September to October to November. This seems to indicate that, even within the short time frame of a few months, cascades are evolving to become less active. We speculate that less active cascades are more likely to evade detection by Facebook, and this has resulted in the reduction over time of cascade intensity.

We further confirm the reduction in cascade activity over time by studying the distribution across cascades of the maximum gap in time between successive posts of a cascade. Figure 6 shows that the maximum inter-posting time too increases from September to November.

3.3 Relationship between users and cascades

Next, we analyze cascades from the perspective of two types of users—1) those who participate in cascades, and 2) those who are exposed to cascades.

Compromised accounts dominate cascades. Users who participate in cascades are those, on whose behalf, malicious posts are made. Figure 7 shows the distribution across such users of the number of cascades in which they participate. We see that 73% of such users were involved in only a single cascade, and out of those for whom we have at least 10 posts in our dataset, more than 99.5% of their posts are benign for over 99% of users. This seems to indicate that a vast majority of users who enable cascades to propagate on Facebook are those whose accounts are compromised, rather than corresponding to accounts controlled by hackers. This is in line with previous findings by Gao et al. [14].

However, we still find 987 users (around 2.65% of all users who participate in cascades) who were involved in more than 5 cascades. Over 95% of these are personal users, 1.3% of them are Facebook pages, and the remaining were deleted when we tried to obtain their information. In addition, we analyze 91 user accounts that participated in over 10 cascades, and find that, for 29% of these accounts, all malicious posts made on behalf of these users were posted by Facebook applications. Furthermore, across these users, on behalf of whom applications are making malicious posts, we found that the average time between when the first and last malicious posts were made by the account is 21 days. We suspect that fraudulent applications installed by these users are making posts with different posted URLs to advertise new cascades over time.

Interestingly, we find that 2.3% of the 15K users subscribed to MyPageKeeper, i.e., roughly 350 users, have participated in cascades. Since users subscribed to MyPageKeeper are conscious of their security on Facebook, the fact that even such users fall prey to cascades highlights the challenges in ensuring that users are not lured by hackers.

The most active friends expose users to cascades. Next,

Category	Distribution methods
Applications	User installs a Facebook application and the application posts on user's behalf on his wall or on his friends' walls
Plugins	User posts on his wall or on his friends' walls by clicking the Like/Share button from a website
	User clicks on an image or video on a malicious website, where the hacker "hijacks" the user's click to share the link to the website as a status update on behalf of the user
	User gives his Facebook personal upload email address to hackers, who send email to this address to update user's status
Manual posting	Spam posted on the wall of a Facebook page is received in the news feed of a user who follows that page
	User copies specialized javascript snippets designed by hackers into the browser's address bar, thus enabling the hackers to post on the user's wall by utilizing the current session information in the browser.

Table 6: Distribution techniques for each cascade source.

cascade source	Fraction of cascades
Applications	44.7%
Plugins	32.4%
Manual posting	12.9%
Total	*90.0%*

Table 5: Percentage of cascades for which more than 95% of posts originated from a single type of source.

we analyze users who are exposed to cascades, i.e., users who see malicious posts either on their wall or in their news feed. Figure 7 shows that 60% of users were exposed to at least one cascade over the course of the five months in our dataset. Among these victims, over 16% were exposed to more than 10 cascades during this period.

To understand if users with more friends are likely to be exposed to more cascades, we group users exposed to similar number of cascades using a bin size of 5. In Figure 8, we then plot the 5^{th}, 25^{th}, 50^{th}, 75^{th}, and 95^{th} percentile value within each group for the number of friends. Unlike what one might expect, we see no obvious relationship between the number of friends that a user has and the number of cascades she was exposed to.

However, we do find that users are more likely to be exposed to cascades because of posts made by their most active friends. To show this, we rank every user's friends by the number of posts (both malicious and benign) seen in the user's news feed, and normalize every friend's rank by the total number of friends that the user has. Thus, a friend who makes more posts has rank closer to 0, whereas a friend with fewer posts has a rank closer to 1. As seen in Figure 9, over 57% of malicious posts seen in users' news feeds were posted by the top 20% most active friends. Thus, friends who post more actively on Facebook are more likely to expose a user to socware cascades.

4 Forensics Analysis

Next, we investigate the mechanisms and infrastructure that enable or are used by socware cascades, and the intentions underlying socware.

4.1 Provenance of malicious posts

First, we classify malicious posts according to their sources of origin in order to understand what enables these cascades to reach so many users. Posts can be made by any one of the following: (a) a Facebook app, (b) a plugin function connected to a button on a website that the user is visiting, or (c) manually by a user. We refer to these methods as *cascade sources*.

For every post, we examine the *application* field of the post, which contains the *application ID* of the originating application [2]. If the *application* field exists and is empty, it typically means that the post was made by a plugin [6]. If the *application* field is missing altogether, then it indicates that the post was made manually by a user. Note that, while this enables us to determine the source of every post, our dataset does not enable us to track the propagation of cascades across users; this is because, when a socware post is made by a friend of a user subscribed to MyPageKeeper, we do not have access to posts that that friend would see on her news feed.

Most cascades (90%) are enabled by one type of cascade source. Since posts in each cascade might be from multiple sources, we first study, for each cascade source, the percentage of cascades in which that source accounted for over 95% of posts. Table 5 shows that, for 90% of cascades, more than 95% of the posts were the result of a single cascade source. A potential explanation is that hackers decide on a single distribution method for each cascade. This makes the following discussion easier, since we can associate each cascade with its dominant source.

Roughly 44% of cascades are enabled by Facebook applications. Further, we also observe from Table 5 that applications are the largest contributors, with apps covering 44.7% of cascades. Plugins also account for a significantly high fraction (32.4%) of cascades, illustrating the significant role that "Shares" and "Likes" play in enabling socware to spread. Interestingly, manual posting too accounts for a sizeable fraction (12.9%) of cascades. We discuss each category in more detail below.

A deeper investigation of cascade sources. To dig deeper into the different cascade propagation mechanisms, for each cascade source, we manually studied posts from that source, and read blogs and security articles that discuss the source. In addition, we inspected the landing web pages of several cascades. Below, we discuss each category separately and show real examples for each, while Table 6 provides an overview of our findings.

a. Applications. Applications enable 44.7% of cascades. According to Facebook's policy, applications require user authorization to post on the user's behalf. Once an application has been granted permission, it can post whatever and whenever it wants on the user's wall, until the user changes his password or uninstalls the application. A representative example of such a cascade tries to lure people into installing an application by offering to: "Check if a friend has deleted you". We elaborate on the role of applications in enabling socware cascades in Section 4.2.

b. Plugins. Plugins are the main source for 32.4% of cascades. In this case, users are lured into visiting a website, where hackers get them to share the link of the website on their wall using the *Share, Like, Recommend* functionalities, which are described as "social plugins" by Facebook [7]. There

[6]This behavior changed sometime in April 2012, but it was true during the period of our dataset, as we manually verified on a sample of posts. Detecting plugin-based posts will require new techniques in the future.

[7]To clarify, there are functions with the same names (Share and Like) which operate on Facebook profiles, but we are

are three predominant methods that hackers use here: (a) they convince users to knowingly share the website's link on their wall, (b) they trick users to click in such a manner that enables sharing without them realizing it, and (c) they use the Facebook Personal Upload Email, which we discuss below. Note that, since posts made with any of these three methods are indistinguishable in our dataset, we cannot provide a breakdown among the three cases.

In the first case, a website explicitly asks users to click on the *Share* button, often as a condition for the user to get some benefit, such as allowing the user to see an interesting video, e.g., in a cascade which was spreading a photographer's video posts, users are asked to "Click Jaa twice to confirm" that they are older than 18 years; "Jaa" means *Share* in Finnish.

In the second case, websites (which typically contain videos or pictures) use *click hijacking* to post messages on users' profiles without their knowledge or consent. A click-hijacking example seen in our dataset involves a cascade which was advertising the website `http://birthpostions.blogspot.com/` (note the missing "i" in "postions"). When a user clicks the play button on the video at the site, it automatically posts the link on the user's wall. This cascade managed to distribute $95,556$ posts, as per Facebook's statistics.

In the third case, hackers trick users into giving them their *Facebook Personal Upload Emails*. The use of these email addresses is to let users update their status or upload pictures to their profiles by simply sending an email to this address. Hackers, of course, found this as a way to post on users' walls [8]. Once a hacker has this email address for a user, he can send emails that include a malicious URL, which correspond to a status update. This status update appears as a post in the news feed of all the user's friends.

How are the hackers getting these emails? One way, of course, is via malicious cascades, which indicates the self-perpetuating nature of socware. In our dataset, we observed cascades that lured people into completing several steps in order to get a fake reward. For example, in the "free Facebook T-shirt" cascade we mentioned above, the fourth step asks users to share their Facebook Personal Upload Email addresses. In fact, hackers have created a YouTube video to show people how to find their email addresses.

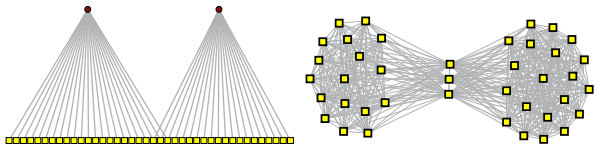
c. Manual posting. 12.9% of cascades are enabled by manual posting as their main source. The posts in these cascades have no application field, and according to the Facebook convention, these posts were posted by users without any help from applications or Facebook's social plugins installed on websites. We identified two large categories of such posts.

First, administrators and owners of Facebook pages (profiles of organizations) can post malicious links on the wall of that profile, possibly unknowingly. As a result, these posts spread to the users who "follow" that profile. For example, the administrator of a page called "I Love My Kids" posted a link to `www.FindJobsForMoms.com`, a fake work-at-home web page. These messages appeared on the news feeds of the users in our dataset that were following that profile.

A second category of manual posts consists of users who blindly follow the instructions on web pages in order to get free stuff. For example, a web page promised "free Facebook T-shirts" and asked users to complete 5 steps. The first step is "Copy And Post Below Message 10 Times on DIFFERENT GROUPS (My Groups) In Facebook."

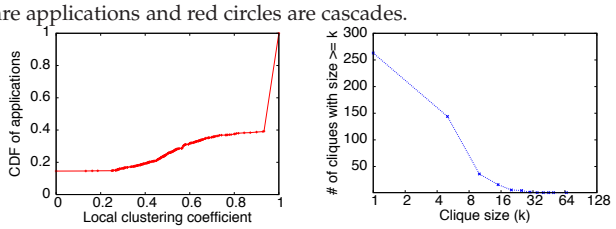
not discussing those here, as they do not lead to posts in this category.

[8]Facebook has removed this email from their mobile website, but the email addresses still work at the time this paper is written.



(a) App-Cascade graph (b) App graph

Figure 10: Sample of the real App-Cascade graph showing two large groups of applications enabling two cascades and the corresponding pair of maximal cliques in the App graph. Yellow squares are applications and red circles are cascades.



(a) Distribution of local clustering coefficient. (b) Lower bound on no. of cliques with size $>= k$.

Figure 11: Properties of App graph.

4.2 Ecosystem of colluding applications

Intrigued by the observation that Facebook apps significantly contribute to enabling socware cascades, we next investigate the relationships between such malicious applications.

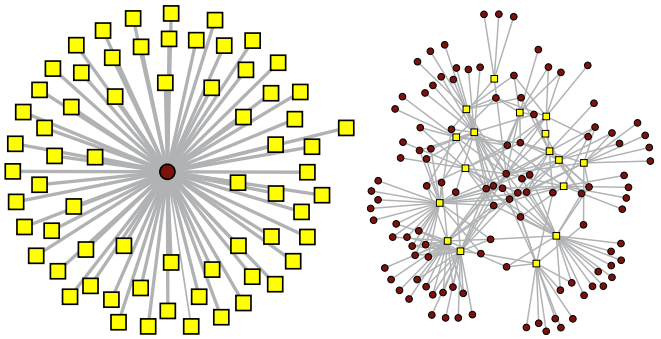
The first natural question is: Do these applications collaborate? We consider two graphs. First, we consider a bipartite App-Cascade graph, whose nodes are applications (as defined by applications IDs) and cascades. An edge between an application and a cascade exists if the application has enabled the cascade, i.e., some posts in the cascade were made by the application. Based on the App-Cascade graph, we create the App graph, whose nodes are applications and an edge between nodes exists iff the applications have enabled one or more common cascades; an edge exists between applications A and B if there exists a cascade in which some posts were made by A and some other posts by B. The App graph from our dataset has $1,670$ nodes and $11,612$ edges, with 220 connected components. Figure 10 shows a real sampled instance of the transformation between the two types of graphs.

We quantify the collaborative nature and the structure of application interactions as follows.

a. Malicious applications are collaborating: Roughly 60% of applications have a fully connected neighborhood. Figure 11a shows the distribution of local clustering coefficient across nodes in the App graph. We can see that over 60% of nodes have their local clustering coefficient equal to 1. This means that 60% of applications are such that their one-hop neighbors form cliques, which occurs only if every pair of them shared at least one cascade. Note that 70% of the nodes with local clustering coefficient equal to 1 have degree greater than 5.

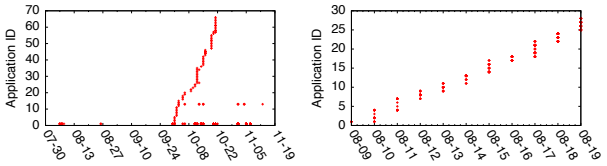
b. There are at least 144 cliques of collaborating applications with more than 5 members. We further use a heuristic method to find a lower bound on the number of cliques that are larger than a threshold k [9] In Figure 11b, we show the

[9]Finding cliques larger than a certain threshold is NP-hard.



(a) 66 apps support one cascade. (b) 17 apps support several cascades.

Figure 12: Sample of collaboration between applications supporting cascades in the App-Cascade graph.



(a) (b)

Figure 13: How apps support the same cascade over time. Y-axis represents unique apps and x-axis is time in days. A point shows that the app propagated posts for the cascade on that day.

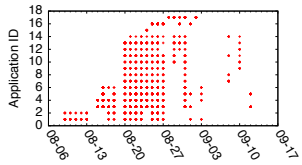


Figure 14: The active days of a clique of applications.

number of cliques at least as large as the x-axis value. We see that there are 144 cliques with size larger than 5, with the largest clique having 66 nodes. Figure 12 shows two real samples of colluding applications (the yellow squares) that form a clique in the App graph.

c. Applications collaborate in a structured manner. To further understand the behavior of applications with respect to cascades, in Figure 13, we show applications that are supporting a single cascade over time, with a different single cascade per plot. The y-axis represents unique applications in the order when they first appeared supporting the cascade, and the x-axis is time in days. In both cases, there seems to exist a structure that implies an engineered process in the number of active applications and the observed duration during which an app supports the cascade. This structure is more pronounced in Figure 13b, where each application seems to be active for two consecutive days for the cascade in question.

d. The behavior of an application group over time exhibits synchronized activity. Finally, we study how a group of applications is active over time across all the cascades that the group supports. In Figure 14, we have identified a collaborating group of 17 applications, and we plot for each application (y-axis) whether it is active on a particular day (x-axis), independently of cascades. We see that the majority of the applications in the group are active during the week 8/20 to 8/27. Note that we show all the activity of this group in our trace; there were no posts from these applications outsides the dates shown.

Our approach hence shows the lower bound of the number of cliques larger than a certain threshold. The specifics of our heuristic are omitted due to space limitations, but the primary takeaway is the existence of many large cliques.

Intention	#/% of cascades	
Phishing and Survey	191	27.64%
Phishing (as apps) and Survey	123	17.8%
Phishing	92	13.32%
Survey	22	3.19%
Advertising Websites	163	23.59%
Unknown	100	14.47%

Table 7: Intentions of socware websites

4.3 Intentions of socware websites

Next, we examine the intentions underlying each cascade by examining the landing web pages that posts in the cascade lead to. We identify 1,247 such unique landing pages in our dataset.

Given the large number of URLs, we cluster them into groups before we investigate them manually. To cluster URLs, we calculate the similarity between every pair of HTML files (without HTML tags) based on the text shingling algorithm [11]. We get 815 clusters, with 40 of them containing more than 2 URLs. We manually check the intention of the cascades in these 40 clusters, which correspond to 691 URLs. We identify five different intentions: *Survey, Phishing, Phishing (as Facebook apps), Advertising Websites,* and *Unknown.* Note that some of these intentions can co-exist; in other words, a cascade may have two or more of these intentions. We show the distribution among intentions and combination of intentions in Table 7.

Most socware cascades do phishing and also benefit from asking users to do a survey. In Table 7, we see that 45.44% of cascades want to collect users' personal information (phishing) and benefit from getting completed surveys at the same time. Hackers either create web pages and ask users to complete a form in order to receive fake rewards, or ask users to install malicious applications, which obtain access to information from their Facebook profile. The lure for users to finish surveys are fake rewards, like free iPads or gift cards. After the survey, users are asked for their contact/shipping information in order to ship fake rewards to users. An example we observed was titled "Check if a friend has deleted you". It shows up as a Facebook application which asks users to authorize them to access users' basic information from their profile. After that, users need to finish a survey in order to see the results.

The second largest category is *Advertising Websites*. These websites are usually scam websites or websites that simply want to increase their traffic and popularity. For example, people keep sharing a well-known fake work-at-home website because they believe that the website is real. Some cascades lure people to websites by promising shocking or funny videos or pictures. However, users need to either "Like" or "Share" this web page with their friends first in order to watch the videos. As mentioned previously, some websites also use click-hijacking to advertise their websites. Facebook has however recently put in measures to prevent click-hijacking; it explicitly requests the user's permission before posting on the user's wall.

The "Unknown" category contains websites which we were

not able to access and crawl. The most common issues were: (a) the site had become inaccessible, or (b) the URL was shortened and the shortening service provider had blacklisted it by the time we attempted to resolve the URL.

4.4 Social engineering tricks

Since many of the underlying intentions are fraudulent, we examine how hackers lure users into clicking on the URLs included in socware posts. After analyzing the posts in our socware dataset, we define six different categories of social engineering techniques based on the text message included in the posts.

- *social-curiosity*: Message tries to arouse users' curiosity about their social relationship, e.g., posts claim a Facebook function that can show who is viewing the user's profile.
- *psycho-curiosity*: Message tries to arouse user's curiosity in general, e.g., a video is advertised with the message: "I dare you can watch this video for 30 seconds".
- *free-stuff*: Message advertises free/cool products such as free iPads or free Facebook T-shirts.
- *combo-winning*: Message tries to attract users by providing free stuff as prize of game and abuse the "willing-to-win" attitude of users at the same time.
- *non-sense*: Message is not in English, is too short, or is too generic (e.g., "hi").
- *no-message*: There is no message; an image is used to attract the user instead.

Based upon manual inspection, we attribute every cascade to one of these six categories; though the text message varies across posts within some cascades, for over 90% of cascades, 95% of posts map to the same category. The detailed breakdown is shown in Tables 8 and 9.

When focusing on cascades seen across at least 10 users, we see that over 75% of the cascades either try to abuse users' curiosity about their friends or try to lure the user by offering free or cool products. These social engineering "hooks" significantly differ from those used in email spam, where hackers either use sexual messages to appeal to users or directly market fraudulent products such as pharmaceuticals, software, and watches [8].

Note that, though *non-sense* is the largest category when we analyze all cascades, its contribution is much lesser on sizeable cascades. This is likely an artifact of most MyPageKeeper users being English speaking. As a result, socware propagated in other languages is not seen in significant numbers in our dataset.

Interestingly, we see that a non-trivial fraction of cascades belong to the *no-message* category. This is because, when a Facebook user shares a link to a website, Facebook will extract the website's description and one image from the site directly. Facebook will then automatically include this additional information with the post. As a result, even if hackers do not explicitly include a text message in a socware post, those who see the post can still see what the post is about. Hackers utilize this feature to presumably evade detection based on the messages included in their posts.

4.5 Cascade hosting infrastructure

Lastly, we analyze the infrastructure that enables cascades. We examined the locations of the servers on which the resolved URLs and landing URLs in our dataset were hosted.

Soc-Eng	#/% of cascades	
non-sense	776	32%
social-curiosity	654	27%
free-stuff	535	23%
no-message	332	14%
psycho-curiosity	57	2%
combo-winning	44	2%

Table 8: Social engineering in cascades with size ≥ 1

Incentive	#/% of cascades	
social-curiosity	478	46%
free-stuff	339	32%
no-message	115	11%
non-sense	65	6%
combo-winning	38	4%
psycho-curiosity	14	1%

Table 9: Social engineering in cascades with size ≥ 10

AS Name	%
Facebook	37.07 %
SoftLayer	10.46 %
Google	8.79 %
Hostway Services	4.52 %
XO Communications	2.85 %

Table 10: Top 5 Landing URL ASes

AS Name	%
SoftLayer	25.99 %
ThePlanet	10.81 %
Google	8.58 %
Hostway Services	4.54 %
Facebook	4.29 %

Table 11: Top 5 Resolved URL ASes

We show the names of the top 5 Autonomous Systems that host these spamming websites in Tables 10 and 11.

37% of unique Landing URLs are in Facebook's domain. When we look at the landing URLs, 37.07% of cascades are hosted in domains that belong to Facebook. These domains include `facebook.com`, `apps.facebook.com`, and `fb.com`. However, if we look at the resolved URLs, only 4.29% of such URLs are hosted in domains belong to Facebook. These two observations show that hackers use redirection websites between the posted URL and the intended destination in order to protect their websites and applications from blacklists, or to have more flexible infrastructure, or both. The flexibility comes from the ability to change the redirection destination without having to change the posted URL.

Other observations of interest include: (a) most resolved URLs are hosted by SoftLayer, a large hosting provider, and (b) the majority of landing URLs (85.6%) reside within the United States. Due to space limitations, we cannot expand this discussion further.

5 Implications

In this section, we summarize the implications that our analysis presents for identifying cascades and for reducing socware on Facebook. Since our results are based on a subset of all of Facebook's users, these implications are speculative to some extent. However, this is a first attempt to put together a set of take-home messages for those combating socware on Facebook. Since hackers continually evolve, analysis along the lines of what we present in this paper will need to repeated to revisit and revise the takeaways.

a. Defining and identifying cascades. We showed that it is significantly more common for hackers to vary the text message they use across different posts in a cascade, than it is for them to vary the `Posted-URL` included in the posts. Therefore, grouping posts based on the `Posted-URL` is an effective way of identifying cascades in the absence of `Landing-URL` information. While this may change in the future—hackers may begin to more commonly use multiple `Posted-URL`s in a cascade to improve resilience to URL blacklisting—our analysis presents the systematic approach that one can repeat to determine how cascades can be identified.

b. Avenues for improving anti-socware efforts. Our results highlight several promising avenues for anti-socware efforts. Our observation that close to half of the cascades in our dataset were due to malicious Facebook applications suggests that socware on Facebook can be significantly reduced by focusing on the detection of malicious applications. Further, we showed that 37% of socware cascades have their landing URL hosted on Facebook—in the form of pages, events, etc. This suggests that Facebook can reduce socware significantly by better policing of the content on its own platform.

c. Speeding up detection approaches via prioritization. For systems such as MyPageKeeper that identify socware, it may prove to be cost prohibitive to check via crawling every URL posted on Facebook. To maximize the chances of detecting new cascades with minimal resources, such systems can prioritize the checking of certain types of posts over others. Studies like this one can help in this direction. Before our study, a potential strategy for such prioritization could be to preferentially focus on posts seen on the news feeds of users with a large number of friends: the more friends a user has, one may expect the chances of that user receiving socware to be greater. However, our analysis shows that this strategy for prioritizing posts will likely not be effective since the chances of a user being exposed to socware appear to be largely uncorrelated with the number of friends that user has. Instead, we see that a user's posting activity is a better indicator of that user aiding a cascade to spread. Thus, an effective strategy for prioritization is to focus on posts made by highly active users.

On the other hand, it appears that focusing on posts made by users who have previously posted spam would also be largely ineffective. In our study, 73% of users (who participated in any cascade) participated in a single cascade through the entirety of our five month dataset. This is because most socware on Facebook is posted by compromised user accounts, rather than by accounts owned by hackers, as further corroborated by previous studies [14, 23].

d. Challenges in combating socware. Finally, our results also highlight several challenges in detecting socware on Facebook. First, we showed that hackers appear to be throttling the rate at which cascades grow. This makes it harder to detect the onset of new cascades simply based on the sudden increase in the sharing of a URL, and thus makes it especially important to rely on other features unrelated to the rate of spreading to detect socware.

Second, we saw that reducing socware by detecting and blacklisting malicious applications is made challenging by the fact that hackers use several applications to spread a single cascade. Unless all applications in a cascade are blacklisted, hackers can simply create new applications and continue the spread of the cascade. This calls for the synergistic identification of all applications associated with a cascade.

Lastly, we found that over half of socware cascades were being propagated by users clicking Like/Share on websites outside of Facebook or by users manually posting spam. Users typically do this in the lure of fake rewards such as free products. Therefore, irrespective of measures adopted by Facebook and others to detect socware, user education is a must to eliminate socware from Facebook.

6 Related Work

We discuss related efforts in brief.

Email spam campaigns. Different campaign definitions have been used in analyzing email spam campaigns. For example, Xie et al.[26] and Moore et al. [21] use the similarity of URLs embedded in emails, while others [22, 12] use email content similarity for grouping spam emails into campaigns. Similar to our definition of campaigns, Konte et al. [17] use landing URLs and Anderson et al. [9] use snapshots of landing webpages to identify email spam campaigns. In contrast to all of these efforts, our focus is on studying socware cascades, which display different characteristics from email spam [23, 13].

Analyzing spamming OSN accounts. Thomas et al. [25] characterized spamming accounts on Twitter and clustered spam campaigns based on the tweets sent from such accounts. Similarly, Benevenuto et al. [10] and Yang et al. [27] developed techniques to identify accounts of spammers on Twitter. Others have proposed a honey-pot based approach [24, 19] to detect spam accounts on OSNs. Yardi et al. [29] analyzed behavioral patterns among spam accounts in Twitter. Recently, Yang et al. [28] show that criminal accounts on Twitter tend to be socially connected. Instead of focusing on accounts created by spammers, we analyze cascades that are predominantly propagated on Facebook by compromised users.

Detecting spam on OSNs. Gao et al. [14, 13] collect posts by crawling the walls of 3.5 million Facebook users and identify spam campaigns based on a combination of the text and the URLs appearing in posts. In a similar study on Twitter, Lee et al. [18] identified spam by grouping tweets into different topic groups based on the similarity of tweet contents. In our work, detection of spam is not of interest. We simply use a dataset of spam previously identified by MyPageKeeper, a Facebook security application, and use this dataset to systematically identify cascades and to analyze the characteristics of cascades.

Analyzing spam campaigns on OSNs. Gao et al. [14] analyzed spam campaigns on Facebook to identify the intention underlying these campaigns. However, they manually select keywords for each intention before grouping posts into campaigns. To understand spam campaigns on Twitter, Grier et al. [15] and Stringhini et al. [24] use landing URLs pointed to by spam tweets to identify campaigns. Unlike prior work, to the best of our knowledge, we present the first systematic analysis of how to identify cascades in the absence of landing URLs pointed to by posts on an OSN. We further analyze the provenance of spam posts and uncover the ecosystem of websites and applications underlying contemporary socware activities on today's most popular OSN—Facebook.

7 Conclusions

In this paper, we presented the first comprehensive study on systematically characterizing socware cascades on Facebook. While most cascades are short, we discovered that hackers seem to be evolving into reducing the activity of cascades in order to evade detection and make cascades last longer. Most interestingly, we showed that a large fraction of socware cascades are supported by Facebook applications that are strategically collaborating with each other in large groups. Lastly, we showed that socware significantly differs from email spam both in terms of the underlying intentions and the tricks used to con users. We believe that these findings will help develop the next generation of anti-socware tools for Facebook.

8 References

[1] Americans spend 23% of Internet time on social networks. http://mashable.com/2011/09/12/23-percent-online/.

[2] Documentation of facebook. http://developers.facebook.com/docs/reference/api/post/.

[3] Facebook: Key facts. http://newsroom.fb.com/content/default.aspx?NewsAreaId=22.

[4] How to fight socware - malware on facebook and other social networks. http://bit.ly/TJvMLd.

[5] Users of social networking websites face malware and phishing attacks. http://www.symantec.com/connect/blogs/users_social_networking_websites_face_malware_and_phishing_attacks.

[6] Your average Facebook post only reaches 12% of your friends. http://techcrunch.com/2012/02/29/facebook-post-reach-16-friends/.

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

[8] Internet security threat report. http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_2011_21239364.en-us.pdf, 2012.

[9] D. Anderson, C. Fleizach, S. Savage, and G. Voelker. Spamscatter: Characterizing internet scam hosting infrastructure. In *USENIX Security*, 2007.

[10] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on Twitter. In *CEAS*, 2010.

[11] A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 1997.

[12] P. Calais, D. Pires, D. Guedes, W. Meira Jr, C. Hoepers, and K. Jessen. A campaign-based characterization of spamming strategies. In *CEAS*, 2008.

[13] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary. Towards online spam filtering in social networks. In *NDSS*, 2012.

[14] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao. Detecting and characterizing social spam campaigns. In *IMC*, 2010.

[15] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The underground on 140 characters or less. In *CCS*, 2010.

[16] G. Keizer. Worm spreads on facebook, hijacks users' clicks. http://www.computerworld.com/s/article/9122724/Worm_spreads_on_Facebook_hijacks_users_clicks.

[17] M. Konte, N. Feamster, and J. Jung. Dynamics of online scam hosting infrastructure. In *PAM*, 2009.

[18] K. Lee, J. Caverlee, Z. Cheng, and D. Sui. Content-driven detection of campaigns in social media. In *CIKM*, 2011.

[19] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots + machine learning. In *SIGIR*, 2010.

[20] G. Milligan and M. Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 1986.

[21] T. Moore, R. Clayton, and H. Stern. Temporal correlations between spam and phishing websites. In *LEET*, 2009.

[22] F. Qian, A. Pathak, Y. Hu, Z. Mao, and Y. Xie. A case for unsupervised-learning-based spam filtering. In *ACM SIGMETRICS Performance Evaluation Review*, 2010.

[23] M. S. Rahman, T.-K. Huang, H. V. Madhyastha, and M. Faloutsos. Efficient and scalable socware detection in online social networks. In *USENIX Security*, 2012.

[24] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *ACSAC*, 2010.

[25] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *IMC*, 2011.

[26] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: signatures and characteristics. In *CCR*, 2008.

[27] C. Yang, R. Harkreader, and G. Gu. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In *RAID*, 2011.

[28] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *WWW*, 2012.

[29] S. Yardi, D. Romero, G. Schoenebeck, et al. Detecting spam in a twitter network. *First Monday*, 2009.