Efficient and Scalable Socware Detection in Online Social Networks

Md Sazzadur Rahman, Ting-Kai Huang, Harsha V. Madhyastha, Michalis Faloutsos
Department of Computer Science and Engineering
University of California, Riverside

Abstract—

Online social networks (OSNs) have become the new vector for cybercrime, and hackers are finding new ways to propagate spam and malware on these platforms, which we refer to as socware. As we show here, socware cannot be identified with existing security mechanisms (e.g., URL blacklists), because it exploits different weaknesses and often has different intentions.

In this paper, we present MyPageKeeper, a Facebook application that we have developed to protect Facebook users from socware. Here, we present results from the perspective of over 12K users who have installed MyPageKeeper and their roughly 2.4 million friends. Our work makes three main contributions. First, to enable protection of users at scale, we design an efficient socware detection method which takes advantage of the *social context* of posts. We find that our classifier is both accurate (97% of posts flagged by it are indeed socware and it incorrectly flags only 0.005% of benign posts) and efficient (it requires 46 ms on average to classify a post). Second, we show that socware significantly differs from traditional email spam or web-based malware. For example, website blacklists identify only 3% of the posts flagged by MyPageKeeper, while 26% of flagged posts point to malicious apps and pages hosted on Facebook (which no current antivirus or blacklist is designed to detect). Third, we quantify the prevalence of socware by analyzing roughly 40 million posts over four months; 49% of our users were exposed to at least one socware post in this period. Finally, we identify a new type of parasitic behavior, which we refer to as "Like-as-a-Service", whose goal is to artificially boost the number of "Likes" of a Facebook page.

1 Introduction

As online social networks (OSNs) are becoming the new epicenter of the web, hackers are expanding their territory to these services [8]. Anyone using Facebook or Twitter is likely to be familiar with what we call here **socware**[1]: fake, annoying, possibly damaging posts from friends of the potential victim. The propagation of socware takes the form of postings and communications between friends on OSNs. Users are enticed into visiting suspicious websites or installing apps with the lure of false rewards (e.g., free iPads in memory of Steve Jobs [30]), and they unwittingly send the post to their friends, thus enabling a viral spreading. This is exactly where the power of socware lies: posts come with the implicit endorsement of the sending friend. Beyond this being a nuisance, socware also enables cyber-crime, with several Facebook scams resulting in loss of real money for users [11, 12].

Defenses against email spam are insufficient for identifying socware since reputation-based filtering [29, 28, 51] is insufficient to detect socware received from friends and, as we show later, the keywords used in email spam significantly differ from those used in socware. We also find that URL blacklists designed to detect phishing and malware on the web do not suffice, e.g., because a large fraction of socware (26% in our dataset) points to suspicious applications hosted on Facebook. Finally, though Facebook has its own mechanisms for detecting and removing malware [52], they seem to be less aggressive either due to what they define as malware or due to computational limitations.

In this paper, we present the design and implementation of a Facebook application, MyPageKeeper [24], that we develop specifically for the purpose of protecting Facebook users from socware. For any subscribing user of MyPageKeeper, whenever socware appears in that user's wall or news feed, we seek to detect the socware soon thereafter and alert the user (hopefully before she views the post). Until October 2011, MyPageKeeper had been installed by more than 12K Facebook users (since its launch in June 2011). By monitoring the news feeds of these users, we also observe posts on the walls of the 2.4 million friends of these users. In this paper, we evaluate MyPageKeeper using a dataset of over 40 million posts that it inspected during the four month period from June to October 2011.

The key contributions of our work can be grouped into three main thrusts.

a. Designing an accurate, efficient, and scalable detection method. In order to operate MyPageKeeper at

[1] We find the introduction of the term socware necessary because, as we elaborate later in Section 2, the types of intent associated with socware encompasses more than traditional phishing and malware.

scale, but at low cost, the distinguishing characteristic of our approach is our strident focus on efficiency. Prior solutions for detecting spam and malware on OSNs (which we describe in detail later) rely on information obtained either by crawling the URLs included in posts or by performing DNS resolution on these URLs. In contrast, our socware classifier relies solely on the *social context* associated with each post (e.g., the number of walls and news feeds in which posts with the same embedded URL are observed, and the similarity of text descriptions across these posts). Note that this approach means that we do not even resolve shortened URLs (e.g., using services like bit.ly) into the full URLs that they represent. This approach maximizes the rate at which we can classify posts, thus reducing the cost of resources required to support a given population of users.

We employ a Machine Learning classification module using Support Vector Machines on a carefully selected set of such features that are readily available from the observed posts. 97% of posts flagged by our classifier are indeed socware and it incorrectly flags only 0.005% of benign posts. Furthermore, it requires an average of only 46 ms to classify a post.

b. Socware is a new kind of malware. We show that socware is significantly different than traditional email spam or web-based malware. First, URL blacklists cannot detect socware effectively. These blacklists identify only 3% of the malicious posts that MyPageKeeper flags. The inability of website blacklists to identify socware is partly due to the fact that a significant fraction of socware is hosted on popular blogging domains such as `blogspot.com` and on Facebook itself. Specifically, 26% of the flagged posts point to Facebook apps or pages. Moreover, we also observe a low overlap between the keywords associated with email based spam and those we find in socware.

c. Quantifying socware: prevalence and intention. We find that 49% of our users were exposed to at least one socware post in four months. We also identify a new type of parasitic behavior, which we refer to as "Like-as-a-Service". Its goal is to artificially boost the number of "Likes" of a Facebook page. With the lure of games and rewards, several Facebook apps push users to *Like* the Facebook pages of say a store or a product, thus artificially inflating their reputation on Facebook. This further confirms the difference between socware and other forms of malware propagation.

2 Socware on Facebook

In this section, we provide relevant background about Facebook, and we describe typical characteristics of socware found on Facebook.

2.1 The Facebook terminology

Facebook is the largest online social network today with over 900 million registered users, roughly half of whom visit the site daily. Here, we discuss some standard Facebook terminology relevant to our work.

- *Post*: A post represents the basic unit of information shared on Facebook. Typical posts either contain only text (status updates), a URL with an associated text description, or a photo/album shared by a user. In our work, we focus on posts that contain URLs.
- *Wall*: A Facebook user's wall is a page where friends of the user can post messages to the user. Such messages are called wall posts. Other than to the user herself, posts on a user's wall are visible to other users on Facebook determined by the user's privacy settings. Typically a user's wall is made visible to the user's friends, and in some cases to friends of friends.
- *News feed*: A Facebook user's news feed page is a summary of the social activity of the user's friends on Facebook. For example, a user's news feed contains posts that one of the user's friends may have shared with all of her friends. Facebook continually updates the news feed of every user and the content of a user's news feed depends on when it is queried.
- *Like*: Like is a Facebook widget that is associated with an object such as a post, a page, or an app. If a user clicks the Like widget associated with an object, the object will appear in the news feed of the user's friends and thus allow information about the object to spread across Facebook. Moreover, the number of Likes (i.e., the number of users who have clicked the Like widget) received by an object also represents the reputation or popularity of the object.
- *Application*: Facebook allows third-party developers to create their own applications that Facebook users can add. Every time a user visits an application's page on Facebook, Facebook dynamically loads the content of the application from a URL, called the canvas URL, pointing to the application server provided by the application's developer. Since content of an application is dynamically loaded every time a user visits the application's page on Facebook, the application developer enjoys great control over content shown in the application page. The Facebook platform uses OAuth 2.0 [2] for user authentication, application authorization and application authentication. Here, application authorization ensures that the users grant precise data (e.g., email address) and capabilities (e.g., ability to post on the user's wall) to the applications they choose to add, and application authentication ensures that a user grants access to her data to the correct application.

2.2 Socware

We start by defining the meaning of *socware*. We describe typical characteristics of socware and elaborate on how socware distinguishes itself from traditional email spam and web malware.

What is socware? Our intention is to use the term socware to encompass all criminal and parasitic behavior in an OSN, including anything that annoys, hurts, or makes money off of the user. In the context of this paper, we consider a Facebook post as malicious, if it satisfies one of the following conditions: (1) the post spreads malware and compromises the device of the user, (2) the web page pointed to by the post requires the user to give away personal information, (3) the post promises false rewards (e.g., free products), (4) the post is made on a user's behalf without the user's knowledge (typically by having previously lured the user into providing relevant permissions to a rogue Facebook app), (5) the web page pointed to by the post requires the user to carry out tasks (e.g., fill out surveys) that help profit the owner of that website, or (6) the post causes the user to artificially inflate the reputation of the page (e.g., by forcing the user to 'Like' the page). While the first two criteria are typical malware and phishing, the latter four are distinctive of socware.

Disclaimer. As with email spam, there can be some ambiguity in the definition of socware: a post considered as annoying by one user may be considered useful by another user. In practice, our ultimate litmus test is the opinion of MyPageKeeper's users: if most of them report a post as annoying, we will flag it as such.

How does socware work? Socware appears in a Facebook user's wall or news feed typically in the form of a post which contains two parts. First, the post contains a URL [2], usually obfuscated with a URL shortening service, to a webpage that hosts either malicious or spam content. Second, socware posts typically contain a catchy text message (e.g. *"two free Southwest tickets"*) that entice users to click on the URL included in the post. Optionally, socware posts also contain a thumbnail screenshot of the landing page of the URL, also used to entice the user to click on the link. For example, a purported image of Osama's corpse is included in a post that claims to point to a video of his death.

The operation of most socware epidemics can be associated with two distinct mechanisms.

a. Propagation mechanism. Once a user follows the embedded URL to the target website, the post tries to propagate itself through that user. For this, the user is often asked to complete several steps in order to obtain the fake reward (e.g., "Free Facebook T-shirt"). These steps involve "Liking" or "sharing" the post, or posting the socware on the user's wall. Thus, the cycle continues with the friends of that user, who see the post in their news feed. In contrast, users seldom forward email spam to their friends.

[2]We leave the identification of socware posts that do not contain an URL for future work. The propagation of socware is harder in such cases, since the user needs to perform a more laborious operation (e.g., enter an URL into the browser's address bar) than simply clicking on the embedded URL.

App Name	Application Message	Monthly Active Users
Free Phone Calls	I'm making a Free Call with the Free Phone Call Facebook App! ... I'll never pay for a phone call again. Make your free call at URL	435,392
The App	Check if a friend has deleted you URL	35,216
The App	Check if a friend has deleted you URL	25,778

Table 1: Three rogue Facebook applications identified by MyPageKeeper.

Page Name	Message to persuade 'Like'	No. of Likes
Clif Bar	Hey there! Looking for a clif builder's coupon? Just like us by clicking the button above. thanks!	79919
FarmVille Bonus	You can't claim you you haven't clicked on the like button	94907
Courtesy Chevrolet	Like our page to play and have a change to win!	86287
Greggs The Bakers	Like us to claim your voucher	288039
Mobilink Infinity	Like us for big infinite fun	26105

Table 2: Top five pages identified by MyPageKeeper that persuade users to 'Like' them.

b. Exploitation mechanism. The exploitation often starts after the propagation phase. The hacker attempts either to learn the user's private information via a phishing attack [9], to spread malware to user devices, or to make money by "forcing" a particular user action or response, such as completing a survey for which the hacker gets paid [19].

Where is socware hosted? Socware can be broadly classified into two categories based on the infrastructure that hosts them.

a. Socware hosted outside Facebook: In this category, URLs point to a domain hosted outside Facebook. Since the URL points to a landing page outside the OSN, hackers can directly launch the different kinds of attacks mentioned above once a user visits the URL in a socware post. Though several URL blacklists should be able to flag such URLs, the process of updating these blacklists is too slow to keep up with the viral propagation of socware on OSNs [44].

b. Socware hosted on Facebook: A significant fraction of socware is hosted on Facebook itself: the embedded URL points to a Facebook page or application. Naturally, current blacklists and reputation-based schemes fail to flag such URLs. Such URLs typically point to the following types of Facebook objects:

- **Malicious Facebook applications:** Rogue applica-

tions post catchy messages (e.g., *"Check who deleted you from your profile"*) on the walls of users with a link pointing to the installation page of the application. Table 1 lists three such socware-spreading applications in our data. Users are conned into installing the application to their profile and granting several permissions to it. The application then not only gets access to that user's personal information (such as email address, home town, and high school) but also gains the ability to post on the victim's wall. As before, posts on a user's wall typically appear on the news feeds of the user's friends, and the propagation cycle repeats. Creating such applications has become easy with ready to use toolkits starting at $25 [18].

- **Malicious Facebook events:** Sometimes hackers create Facebook events that contain a malicious link. One such event is the *'Get a free Facebook T-Shirt (Sponsored by Reebok)'* scam. This event page states that 500,000 users will get a free T-shirt from Facebook. To be one among those 500,000 users, a user must attend the event, invite her friends to join, and enter her shipping address.
- **Malicious Facebook pages:** Another approach taken by hackers to spread socware is to create a Facebook page and post spam links on the page [27]. We also identified a trend in aggressive marketing by companies that force users to click "Like" on their Facebook pages to spread their pages as well as increase the reputation of the page. Table 2 lists the top five such Facebook pages, along with the message on the page and the number of Likes received by these pages.

3 MyPageKeeper Architecture

To identify socware and protect Facebook users from it, we develop MyPageKeeper. MyPageKeeper is a Facebook application that continually checks the walls and news feeds of subscribed users, identifies socware posts, and alerts the users. We present our goals in designing MyPageKeeper, and then describe the system architecture and implementation details.

3.1 Goals

We design MyPageKeeper with the following three primary goals in mind.

1. Accuracy. Our foremost goal is to ensure accurate identification of socware. We are faced with the obvious trade-off between missing malware posts (false negatives), and "crying wolf" too often (false positives). Although one could argue that minimizing false negatives is more important, users would abandon overly sensitive detection methods, as recognized by the developers of Facebook's Immune System [52].

2. Scalability. Our end goal is to have MyPageKeeper provide protection from socware for all users on Facebook, not just for a select few. Therefore, the system must be scalable to easily handle increased load imposed by a growth in the number of subscribed users.

3. Efficiency. Finally, we seek to minimize our costs in operating MyPageKeeper. The period between when a post first becomes visible to a user and the time it is checked by MyPageKeeper represents the window of vulnerability when the user is exposed to potential socware. To minimize the resources necessary to keep this window of vulnerability short, MyPageKeeper's techniques for classification of posts must be efficient.

3.2 MyPageKeeper components

MyPageKeeper consists of six functional modules.

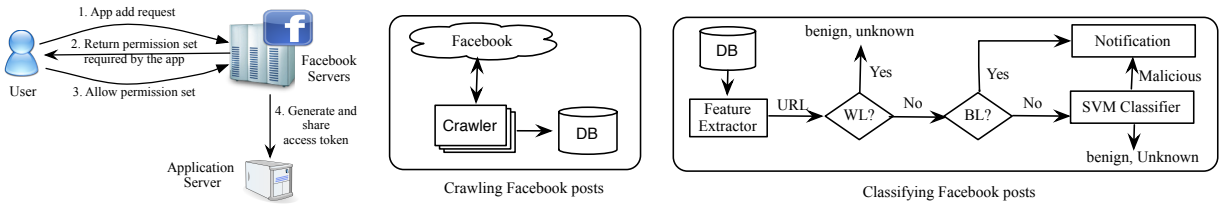
a. User authorization module. We obtain a user's authorization to check her wall and news feed through a Facebook application, which we have developed. Once a user installs the MyPageKeeper application, we obtain the necessary credentials to access the posts of that user. For alerting the user, we also request permission to access the user's email address and to post on the user's wall and news feed. Figure 1(a) shows how a Facebook user authorizes an application.

b. Crawling module. MyPageKeeper periodically collects the posts in every user's wall and news feed. As mentioned previously, we currently focus only on posts that contain a URL. Apart from the URL, each post comprises several other pieces of information, such as a text message associated with the post, the user who made the post, number of comments and Likes on the post, and the time when the post was created.

c. Feature extraction module. To classify a post, MyPageKeeper evaluates every embedded URL in the post. Our key novelty lies in considering only the social context (e.g., the text message in the post, and the number of Likes on it) for the classification of the URL and the related post. Furthermore, we use the fact that we are observing more than one user, which can help us detect an epidemic spread. We discuss these features in more detail later in Section 3.3.

d. Classification module. The classification module uses a Machine Learning classifier based on Support Vector Machines, but also utilizes several local and external whitelists and blacklists that help speed up the process and increase the overall accuracy. The classification module receives a URL and the related social context features extracted in the previous step. Since the classification is our key contribution, we discuss this in more detail in Section 3.3. If a URL is classified as socware, all posts containing the URL are labeled as such.

e. Notification module. The notification module notifies all users who have socware posts in their wall or news feed. The user can currently specify the notification mechanism, which can be a combination of emailing the



(a) (b)

Figure 1: (a) Application installation process on Facebook, and (b) architecture of MyPageKeeper.

user or posting a comment on the suspect posts. In the future, we will consider allowing our system to remove the malicious post automatically, but this can create liabilities in the case of false positives.

f. User feedback module. Finally, to improve MyPageKeeper's ability to detect socware, we leverage our user community. We allow users to report suspicious posts through a convenient user-interface. In such a report, the user can optionally describe the reason why she considers the post as socware.

3.3 Identification of socware

The key novelty of MyPageKeeper lies in the classification module (summarized in Figure 1(b)). As described earlier, the input to the classification module is a URL and the related social context features extracted from the posts that contain the URL. Our classification algorithm operates in two phases, with the expectation that URLs and related posts that make it through either phase without a match are likely benign and are treated as such.

Using whitelists and blacklists. To improve the efficiency and accuracy of our classifier, we use lists of URLs and domains in the following two steps. First, MyPageKeeper matches every URL against a whitelist of popular reputable domains. We currently use a whitelist comprising the top 70 domains listed by Quantcast, excluding domains that host user-contributed content (e.g., OSNs and blogging sites). Any URL that matches this whitelist is deemed safe, and it is not processed further.

Second, all the URLs that remain are then matched with several URL blacklists that list domains and URLs that have been identified as responsible for spam, phishing, or malware. Again, the need to minimize classification latency forces us to only use blacklists that we can download and match against locally. Such blacklists include those from Google's Safe Browsing API [17], Malware Patrol [23], PhishTank [26], APWG [1], SpamCop [28], joewein [20], and Escrow Fraud [7]. Querying blacklists that are hosted externally, such as SURBL [31], URIBL [33] and WOT [34], will introduce significant latency and increase MyPageKeeper's latency in detecting socware, thus inflating the window of vulnerability. Any URL that matches any of the blacklists that we use is classified as socware.

Using machine learning with social context features. All URLs that do not match the whitelist or any of the blacklists are evaluated using a Support Vector Machines (SVM) based classifier. SVM is widely and successfully used for binary classification in security and other disciplines [49, 46] [32]. We train our system with a batch of manually labeled data, that we gathered over several months prior to the launch of MyPageKeeper. For every input URL and post, the classifier outputs a binary decision to indicate whether it is malicious or not. Our SVM classifier uses the following features.

Spam keyword score. Presence of spam keywords in a post provides a strong indication that the post is spam. Some examples of such spam keywords are 'FREE', 'Hurry', 'Deal', and 'Shocked'. To compile a list of such keywords that are distinctive to socware, our intuition is to identify those keywords that 1) occur frequently in socware posts, and 2) appear with a greater frequency in socware as compared to their frequency in benign posts.

We compile such a list of keywords by comparing a dataset of manually identified socware posts with a dataset of posts that contain URLs that match our whitelist (we discuss how to maintain this list of keywords in Section 7). We transform posts in either dataset to a bag of words with their frequency of occurrence. We then compute the likelihood ratio p_1/p_2 for each keyword where $p_1 = p(word|socwarepost)$ and $p_2 = p(word|benignpost)$. The likelihood ratio of a keyword indicates the bias of the keyword appearing more in socware than in benign posts. In our current implementation of MyPageKeeper, we have found that the use of the 6 keywords with the highest likelihood ratio values among the 100 most frequently occurring keywords in socware is sufficient to accurately detect socware.

Thereafter, to classify a URL, MyPageKeeper searches all posts that contain the URL for the presence of these spam keywords and computes a spam keyword score as the ratio of the number of occurrences of spam keywords across these posts to the number of posts.

Message similarity. If a post is part of a spam campaign, it usually contains a text message that is similar to the text in other posts containing the same URL (e.g., because users propagate the post by simply sharing it). On the other hand, when different users share the same

popular URL, they are likely to include different text descriptions in their posts. Therefore, greater similarity in the text messages across all posts containing a URL portends a higher probability that the URL leads to spam. To capture this intuition, for each URL, we compute a message similarity score that captures the variance in the text messages across all posts that contain the URL. For each post, MyPageKeeper sums the ASCII values of the characters in the text message in the post, and then computes the standard deviation of this sum across all the posts that contain the URL. If the text descriptions in all posts are similar, the standard deviation will be low.

News feed post and wall post count. The more successful a spam campaign, the greater the number of walls and news feeds in which posts corresponding to the campaign will be seen. Therefore, for each URL, MyPageKeeper computes counts of the number of wall posts and the number of news feed posts which contained the URL.

Like and comment count. Facebook users can 'Like' any post to indicate their interest or approval. Users can also post comments to follow up on the post, again indicating their interest. Users are unlikely to 'Like' posts pointing to socware or comment on such posts, since they add little value. Therefore, for every URL, MyPageKeeper computes counts of the number of Likes and number of comments seen across all posts that contain the URL.

URL obfuscation. Hackers often try to spread malicious links in an obfuscated form, e.g., by shortening it with a URL shortening service such as *bit.ly* or *goo.gl*. We store a binary feature with every URL that indicates whether the URL has been shortened or not; we maintain a list of URL shorteners.

Note that none of the above features by themselves are conclusive evidence of socware, and other features could potentially further enhance the classifier (e.g., we can account for spam keywords such as 'free' included in URLs such as http://nfljerseyfree.com). However, as we show later in our evaluation, the features that we currently consider yield high classification accuracy in combination.

3.4 Implementing MyPageKeeper

We provide some details on MyPageKeeper's implementation.

Facebook application. First, we implement the MyPageKeeper Facebook application using FBML [14]. We implement our application server using Apache (web server), Django (web framework), and Postgres (database). Once a user installs the MyPageKeeper app in her profile, Facebook generates a secret access token and forwards the token to our application server, which we then save in a database. This token is used by the crawler to crawl the walls and news feeds of subscribed users using the Facebook open-graph API. If any user deactivates MyPageKeeper from their profile, Facebook disables this token and notifies our application server, whereupon we stop crawling that user's wall and news feed.

Data	Total	# distinct URLs
MyPageKeeper users	12,456	-
Friends of MyPageKeeper users	2,370,272	-
News feed posts	38,764,575	29,522,732
Wall posts	1,760,737	1,532,055
User reports	679	333

Table 3: Summary of MyPageKeeper data.

Crawler instances and frequency. We run a set of crawlers in Amazon EC2 instances to periodically crawl the walls and news feeds of MyPageKeeper's users. The set of users are partitioned across the crawlers. In our current instantiation, we run one crawler process for every 1,000 users. Thus, as more users subscribe to MyPageKeeper, we can easily scale the task of crawling their walls and news feeds by instantiating more EC2 instances for the task. Our Python-based crawlers use the open-graph API, incorporating users' secret access tokens, to crawl posts from Facebook. Once the data is received in JSON format, the crawlers parse the data and save it in a local Postgres database.

Currently, as a tradeoff between timeliness of detection and resource costs on EC2, we instantiate MyPageKeeper to crawl every user's wall and news feed once every two hours. Every couple of hours, all of our crawlers start up and each crawler fetches new posts that were previously not seen for the users assigned to it. Once all crawlers complete execution, the data from their local databases is migrated to a central database.

Checker instances. Checker modules are used to classify every post as socware or benign. Every two hours, the central scheduler forks an appropriate number of checker modules determined by the number of new URLs crawled since the last round of checking. Thus, the identification of socware is also scalable since each checker module runs on a subset of the pool of URLs. Each checker evaluates the URLs it receives as input—using a combination of whitelists, blacklists, and a classifier—and saves the results in a database. We use the libsvm [41] library for SVM based classification. Once all checker modules complete execution, notifiers are invoked to notify all users who have posts either on their wall or in their news feed that contain URLs that have been flagged as socware.

4 Evaluation

Next, we evaluate MyPageKeeper from three perspectives. First, we evaluate the accuracy with which it classifies socware. Second, we determine the contribution of MyPageKeeper's social context based classifier in identifying socware compared to the URL blacklists that it uses. Lastly, we compare MyPageKeeper's efficiency

Feature	F-Score
URL obfuscated?	0.300378
Spam keyword score	0.262220
# of news feed posts	0.173836
Message similarity score	0.131733
# of Likes	0.039895
# of wall posts	0.019857
# of comments	0.006367

Table 4: Feature scores used by MyPageKeeper's classifier.

Alternative source	# of posts
Flagged by blacklist	18,923
Flagged by on.fb.me	2,102
Content deleted by Facebook	3,918
Blacklisted app	1,290
Blacklisted IP	5,827
Domain is deleted	247
Points to app install	4,658
Spamming app	6,547
Manually verified	14,876
True positives	58,388 (97%)
Unknown	1,803 (3%)
Total	60,191

Table 5: Validation of socware flagged by MyPageKeeper classifier.

with alternative approaches that would either crawl every URL or at least resolve short URLs in order to identify socware.

Table 3 summarizes the dataset of Facebook posts on which we conduct our evaluation. This data is obtained during MyPageKeeper's operation over a four month period from 20^{th} June to 19^{th} October, 2011. MyPageKeeper had over 12K users during this period, who had around 2.37M friends in union. Our data comprises 38.7 million and 1.7 million posts that contain URLs from the news feeds and walls of these 12K users. We consider only those posts that contain URLs since MyPageKeeper currently checks only such posts. Overall, these 40 million posts contained around 30 million unique URLs. In addition, we received 679 reports of socware from 533 distinct MyPageKeeper users during the four month period, with 333 distinct URLs across these reports. Though it is hard to make any general claims with regard to representativeness of our data, we find that several user metrics (e.g., the male-to-female ratio and the distribution of users across age groups) closely match that of the Facebook user base at-large.

4.1 Accuracy

As previously mentioned, MyPageKeeper first matches every URL to be checked against a whitelist. If no match is found, it checks the URL with a set of locally queriable URL blacklists. Finally, MyPageKeeper applies its social context based classifier learned using the SVM model. In this process, we assume URL information provided by whitelists and blacklists to be ground truth, i.e., classification provided by them need not be independently validated. Therefore, we focus here on validating the socware flagged by MyPageKeeper's classifier based on social context features.

App name	Description	# of posts
Sendible	Social Media Management	6,687
iRazoo	Search & win!	1,853
4Loot	4Loot lets you win all sorts of Loot while searching the web	1,891

Table 6: Top three spamming applications in our dataset.

We trained MyPageKeeper's classifier using a manually verified dataset of URLs that contain 2,500 positive samples and 5,000 negative samples of socware posts; we gathered these samples over several months while developing MyPageKeeper. Table 4 shows the importance of the various features in the SVM classifier learned. During the course of MyPageKeeper's operation over four months, we applied the classifier to check 753,516 unique URLs; these are URLs that do not match the whitelist or any of the blacklists. Of these URLs, the classifier identified 4,972 URLs, seen across 60,191 posts, as instances of socware.

It is important to note that when MyPageKeeper sees a URL in multiple posts over time, the values of the features associated with the URL may change every time it appears, e.g., the message similarity score associated with the URL can change. However, once MyPageKeeper classifies a URL as socware during any of its occurrences, it flags all previously seen posts that contain the URL and notifies the corresponding users. Therefore, in evaluating MyPageKeeper's classifier, URL blacklists, or MyPageKeeper as a whole, we consider here that a technique classified a particular URL as socware if that URL was flagged by that technique upon any of the URL's occurrences. Correspondingly, we consider a URL to have not been classified as socware if it was not identified as such during any of its occurrences.

Checking the validity of socware identified by MyPageKeeper's classifier is not straightforward, since there is no ground truth for what represents socware and what does not. However, here we attempt to evaluate the positive samples of socware identified by MyPageKeeper's classifier using a combination of a host of complementary techniques (we later discuss in Section 7 the validation of posts that are deemed safe by MyPageKeeper). To do so, we use an instrumented Firefox browser to crawl the 4,972 URLs flagged by MyPageKeeper at the end of the four month period of MyPageKeeper operation. For every URL that we crawl, we record the landing URL, the IP address and other whois information of the landing domain, and contents of the landing page. To verify the reputation of every URL, we then apply several techniques in the order summarized in Table 5.

- *Blacklisted URLs:* First, we check if any of the URLs or the corresponding landing URLs are found in any

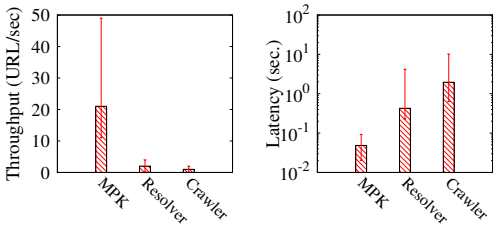
URL blacklists. Note that, though we use blacklists in the operation of MyPageKeeper itself, we use only those that can be stored and queried locally. Therefore, here we use for validation other external blacklists for which we have to issue remote queries. Further, even for blacklists used in MyPageKeeper, they may not identify some instances of socware when they initially appear because blacklists have been found to lag in keeping up with the viral propagation of spam on OSNs [44]. Hence, we check if a URL identified as socware by MyPageKeeper's classifier appeared in any of the blacklists used by MyPageKeeper at a later point in time, even though it did not appear in any of those blacklists initially when MyPageKeeper spotted posts containing that URL.

- *Flagged by fb.me URL shortener:* Many URLs posted on Facebook are shortened using Facebook's URL shortener `fb.me`. When Facebook determines any link shortened using their service to be unsafe, the corresponding shortened URL thereafter redirects to Facebook's home page—`facebook.com/home.php`—instead of the actual landing page. Of the URLs flagged by MyPageKeeper's classifier, we check if those shortened using Facebook's URL shortening service redirect to Facebook's home page.
- *Content deleted from Facebook:* If Facebook determines any URL hosted under the `facebook.com` domain to be unsafe (e.g., the page for a spamming Facebook application), it thereafter redirects that URL to `facebook.com/4oh4.php`. We use this as another source of information to validate URLs flagged by MyPageKeeper's classifier.
- *Blacklisted apps:* If the URLs in posts made by a Facebook app are flagged due to any of the above reasons, we consider that app to be malicious and declare all other URLs posted by it as unsafe, thus helping validate some of the URLs declared as socware by MyPageKeeper's classifier.
- *Blacklisted IPs:* For every URL flagged by any of the above techniques, we record the IP address when that URL is crawled and blacklist that IP. Of the URLs flagged by MyPageKeeper's classifier, we then consider those that lead to one of these blacklisted IP addresses as correctly classified.
- *Domain deleted:* Malicious domains are often deleted once they are caught serving malicious content. Therefore, we deem MyPageKeeper's positive classification of a URL to be correct if the domain for that URL no longer exists when we attempt to crawl it.
- *Obfuscation of app installation page:* Posts made by Facebook applications to attract users to install them typically include an un-shortened URL pointing to a Facebook page that contains information about the application. Once a user visits this page, she can read the application's description and then click on a link on this page if she decides to install it. However, posts from some surreptitious applications contained shortened URLs that directly take the user to a page where they request the user to grant permissions (e.g., to post on the user's wall) and install the application. We have found all instances of such applications to be spamming applications. Therefore, if any of the URLs flagged by MyPageKeeper's classifier is a shortened URL that directly points to the installation page for a Facebook app, we declare that classification correct.
- *Spamming app:* From our dataset, we manually identified several Facebook applications that try to spread on Facebook by promising free money to users and make posts that point to the application page. Once installed by a user, such applications periodically post on the user's wall (without requesting the user's authorization for each post) in an attempt to further propagate by attracting that user's friends; Table 6 shows some such applications that frequently appear in our dataset. Any URLs classified as socware by MyPageKeeper's classifier that happen to be posted by one of these manually identified spamming apps are deemed correct.
- *Manual analysis*: Finally, over the operation of MyPageKeeper during the four months, we periodically verified a subset of URLs flagged by the classifier. These provide an additional source of validation.

Source	# (%) of URLs	# (%) of posts	Overlap with classifier (# of URLs)
Google SBA2	221 (6.8%)	378 (0.4%)	0
Phishtank	12 (0.4%)	435 (0.5%)	1
Malware Norm	69 (2.1%)	154 (0.2%)	0
Joewein	240 (7.4%)	652 (0.7%)	11
APWG	56 (1.7%)	569 (0.6%)	0
Spamcop	232 (7.1%)	921 (1.0%)	0
All blacklists	830 (25.6%)	3104 (3.4%)	12
MyPageKeeper classifier	2405 (74.4%)	89389 (96.6%)	

Table 7: Comparison of contribution made by blacklists and classifier to MyPageKeeper's identification of socware during the four month period of operation.

In all, the union of the above techniques validates that 58,388 out of 60,191 posts declared as socware by the MyPageKeeper classifier are indeed so. Therefore, 97% of the socware identified by MyPageKeeper's classifier are true positives. On the other hand, the 1,803 posts incorrectly classified as socware constitute less than 0.005% of the over 40 million posts in our dataset. Note that, though all of the above techniques could be folded into MyPageKeeper itself to help identify socware, we do not do so because all of these techniques require us to crawl a URL in order to evaluate it; we cannot afford the latency of crawling.



(a) Throughput (b) Latency

Figure 2: Comparison of MyPageKeeper's throughput and latency in classifying URLs with a short URL resolver and a crawler-based approach. The height of the box shows the median, with the whiskers representing 5^{th} and 95^{th} percentiles.

4.2 Comparison with blacklists

Though we see that the identification of socware by MyPageKeeper's classifier is accurate, the next logical question is: what is the classifier's contribution to MyPageKeeper in comparison with URL blacklists? Table 7 provides a breakdown of the URLs and posts classified as socware by MyPageKeeper during the four month period under consideration. There are two main takeaways from this table. First, we see that the classifier finds 74.4% of socware URLs and 96.6% of socware posts identified by MyPageKeeper. Thus, the classifier accounts for a large majority of socware identified by MyPageKeeper and is thus critical to the system's operation. Second, there is very little overlap between the URLs flagged by blacklists and those flagged by the classifier. The typically low frequency of occurrence of URLs that match blacklists is another reason that the classifier's share of identified socware posts is significantly greater than its corresponding share of flagged URLs.

4.3 Efficiency

Beyond accuracy, it is critical that MyPageKeeper's identification of socware be efficient, so as to minimize the costs that we need to bear in order to keep the delay in identifying socware and alerting users low. The matching of a URL against a whitelist or a local set of blacklists incurs minimal computational overhead. In addition, we find that execution of the classifier also imposes minimal delay per URL verified.

To demonstrate the efficiency of MyPageKeeper, we compare the rate at which it classifies URLs with the classification throughput that two other alternative classes of approaches would be able to sustain. Our first point of comparison is an approach that relies only on locally queriable URL whitelists and blacklists but resolves all shortened URLs into the corresponding complete URL. Our second alternative crawls URLs to evaluate them, e.g., using the content on the page or the IP address of the target website. Figure 2(a) compares the throughput of classifying URLs with the three approaches, using data from

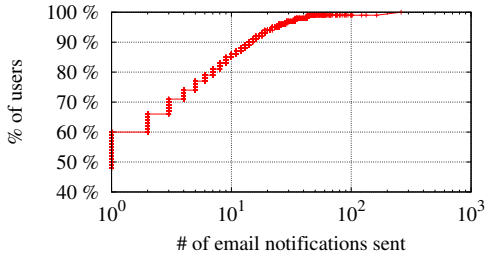


Figure 3: 49% of MyPageKeeper's 12,456 users were notified of socware at least once in four months of MyPageKeeper's operation.

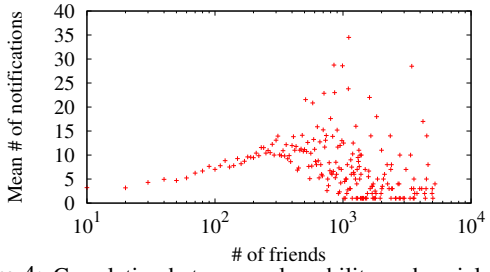


Figure 4: Correlation between vulnerability and social degree of exposed users.

two weeks of MyPageKeeper's execution. We see that the throughput with MyPageKeeper is almost an order of magnitude greater than the alternatives, with all three approaches using the same set of resources on EC2. As we see in Figure 2(b), MyPageKeeper's better performance stems from its lower execution latency to check an URL; the median classification latency with MyPageKeeper is 48 ms compared to a median of 426 ms when resolving short URLs and 1.9 seconds when crawling URLs. Thus, we are able to significantly reduce MyPageKeeper's classification latency, compared to approaches that need to resolve short URLs or crawl target web pages, by keeping all of its computation local.

Furthermore, a crawler-based approach will be significantly more expensive than MyPageKeeper. Thomas et al. [54] found that crawler-based classification of 15 million URLs per day using cloud infrastructure results in an expense of $800/day. Therefore, we estimate that it would cost approximately $1.5 million/year to handle Facebook's workload; 1 million URLs are shared every 20 minutes on Facebook [35]. Since MyPageKeeper's classification latency is 40 times less than a crawler-based approach, we estimate that the expense incurred with MyPageKeeper would be at least 40 times lower than a system that classifies URLs by crawling them.

5 Analysis of Socware

Thus far we described how MyPageKeeper detects socware efficiently at scale. In this section, we analyze the socware that we have found during MyPageKeeper's operation to throw light on characteristics of socware on Facebook.

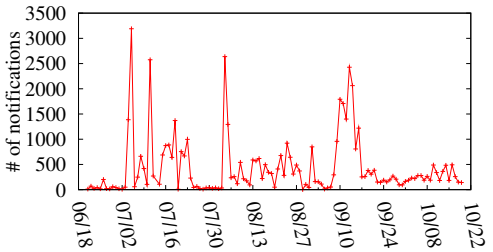


Figure 5: No. of socware notifications per day. On 11th July, 19th Sep, and 3rd Oct, socware was observed in large scale.

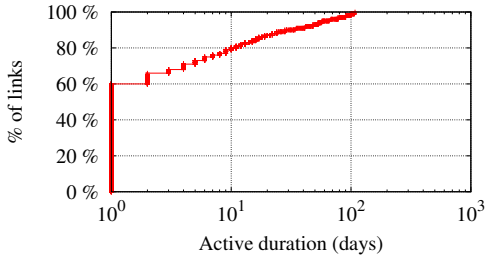


Figure 6: Active-time of socware links. 20% of socware links were observed more than 10 days apart.

5.1 Prevalence of socware

49% of MyPageKeeper's users were exposed to socware within four months. First, we analyze the prevalence of socware on Facebook. To do so, we define that a user was exposed to a particular socware post if that post appeared in her wall or news feed. As shown in Figure 3, 49% of MyPageKeeper's users were exposed to at least one socware post during the four month period we consider here. Though this already indicates the wide reach of socware on Facebook, we stress that 49% is only a lower-bound due to a couple of reasons. First, many of MyPageKeeper's users subscribed to our application at some time in the midst of the four month period and therefore, we miss socware that they were potentially exposed to prior to them subscribing to MyPageKeeper. Second, Facebook itself detects and removes posts that it considers as spam or pointing to malware [36, 38, 52]. All the socware detected by MyPageKeeper is after such filtering by Facebook.

Given that some users are exposed to more socware than others, we analyze if the social degree of a user has any impact on the probability of a user being exposed to socware. Figure 4 shows the number of socware notifications received by MyPageKeeper users as a function of the number of friends they have on Facebook. We bin users with the number of friends within 10 of each other and plot the average number of notifications per bin; we consider here only those users who were subscribed to MyPageKeeper for at least three months. We see that the probability of users being exposed to socware is largely independent of their social degree. This indicates that whether a user is more likely to be exposed to socware is not simply a function of how many friends she has, but likely depends on the susceptibility of those friends to becoming victims of scams and helping propagate them.

Shortening service	% of socware URLs
bit.ly	21.9%
tinyurl.com	18.8%
goo.gl	5.1%
t.co	3.16%
tiny.cc	1.6%
ow.ly	1.1%
on.fb.me	1.0%
is.gd	0.7%
j.mp	0.4%
0rz.com	0.3%
All shortened URLs	54%

Table 8: Top URL shortening services in our socware dataset.

Domain Name	% of URLs	% of posts
facebook.com	20.7%	26.3%
blogspot.com	6.3%	8.7%
miessass.info	1.9%	3.2%
shurulburul.tk	1.8%	1.2%
tomoday.info	0.8%	0.13%

Table 9: Top two-level domains in our socware dataset.

We also find that socware on Facebook is prevalent over time. Figure 5 shows the number of socware notifications sent per day by MyPageKeeper to its users. We see a consistently large number of notifications going out daily, with noticeable spikes on a few days. On 11^{th} July 2011, a scam that conned users to complete surveys with the pretext of fake free products went viral and posts pointing to the scam appeared 4,056 times on the walls and news feeds of MyPageKeeper's users. Two other scams, that promised *'Free Facebook shoes'* and conned users to fill out surveys, also caused MyPageKeeper to send out a large number of notifications on that day. On 19^{th} Sep. 2011, different variants of the *'Facebook Free T-Shirt'* scam [9] were spreading on Facebook and was spotted 2,040 times by MyPageKeeper. On 3^{rd} Oct. 2011, a video scam was spreading on Facebook and MyPageKeeper observed it in 1,739 posts.

We next analyze the prevalence and impact of socware from the perspective of individual socware links. For each link, we define its "active-time" as the difference between the first and last times of its occurrence in our dataset. Figure 6 shows that we did not see 60% of socware links beyond one day. Subsequent posts containing these links may have been filtered by Facebook once it recognized their spammy or malicious nature, or our dataset may miss those posts due to MyPageKeeper's limited view into Facebook's 850 million users. Further, we do not attempt any clustering of links into campaigns here. However, even with these caveats, 20% of socware links were seen in multiple posts separated by at least 10 days, suggesting that a significant fraction of socware eludes Facebook's detection mechanisms and lasts on Facebook for significant durations.

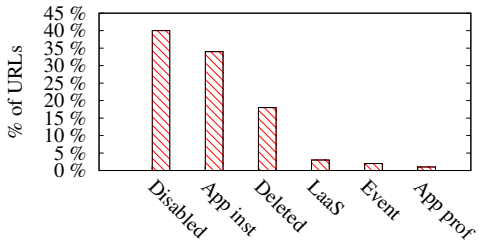


Figure 7: Breakdown of socware links, when crawled in Nov. 2011, that originally point to web pages in the `facebook.com` domain.

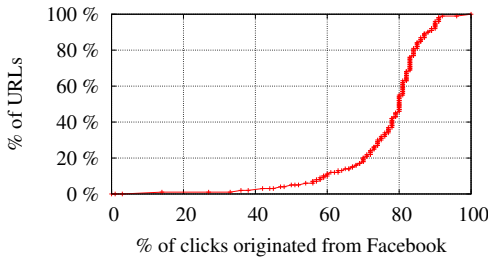


Figure 8: For most socware links shortened with bit.ly or goo.gl, a large majority of the clicks came from Facebook.

5.2 Domain name characteristics

20% of socware links are hosted inside Facebook. In the next section of our analysis, we focus on the domain-level characteristics of socware links. First, Table 8 shows the top ten URL shortening services used in socware links observed by MyPageKeeper. In all, shortened URLs account for 54% of socware links in our dataset. Our design of MyPageKeeper's classifier to rely solely on social context, and to not resolve short URLs, hence makes a significant difference (as previously seen in the comparison of classification latency).

Further, we find it surprising that a large fraction of socware links (46%) are not shortened, given that shortening of URLs enables spammers to obfuscate them. On further investigation, we find that many Facebook scams such as *'free iPhone'* and *'free NFL jersey'* use domain names that clearly state the message of the scam, e.g., http://iphonefree5.com/ and http://nfljerseyfree.com/. These URLs are more likely to elicit higher click-through rates compared to shortened URLs. On the other hand, most of the shortened URLs were used by malicious or spam applications (e.g., 'The App', 'Profile Stalker') that generate shortened URLs pointing to their application's installation page. We find that 89% of shortened URLs in our dataset of socware links were posted by Facebook applications.

Next, based on our crawl of the socware links in our dataset, we inspect the top two-level domains found on the landing pages pointed to by these links. First, as shown in Table 9, we find that a large fraction of socware (over 20% of URLs and 26% of posts) is hosted on Facebook itself. Second, a sizeable fraction of socware uses sites such as `blogspot.com` and `wordpress.com` that enable the spammers to easily create a large number of URLs without going through the hassle of registering new domains. Further, all of these domains are of good repute and are unlikely to be flagged by traditional website blacklists.

5.3 Analysis of socware hosted in Facebook

Hackers use numerous channels in Facebook to spread socware. Given the large fraction of socware hosted on Facebook itself, we next analyze this subset of socware. First, in early November 2011, we crawled every socware link in our dataset that had pointed to a landing page in the `facebook.com` domain at the time when MyPageKeeper had initially classified that link as socware. Figure 7 presents a breakdown of the results of this crawl. If Facebook disables a URL, it redirects us to `facebook.com/home.php`. Similarly, if crawling a URL points us to `facebook.com/4oh4`, it implies that Facebook has deleted the content at that URL. Therefore, as seen in Figure 7, a large fraction of socware links that were originally pointing to Facebook have now been deactivated. However, we also see that a significant fraction of these links—over 40%—were still live. Further, the figure shows that spammers use several different channels, such as applications, events, and pages to propagate their scams on Facebook. In the figure, *'App inst'* and *'App prof'* refer to the installation and profile pages of Facebook applications, and *'LaaS'* refers to campaigns intended to increase the number of Likes on a Facebook page (described in detail in Section 6).

In our dataset, we see 257 distinct socware links shortened with the `bit.ly` and `goo.gl` URL shorteners that point to landing pages in the `facebook.com` domain. Using the APIs [5, 16] offered by these URL shortening services, we computed the number of clicks recorded for these 257 links in two cases—1) where the Referrer was Facebook, and 2) where the Referrer was any other domain. Figure 8 shows that Facebook is the dominant platform from which most of these links received most of their clicks; 80% of links received over 70% of their clicks from Facebook. This seems to indicate that most socware hosted on Facebook is propagated solely on Facebook and tailored for that platform.

5.4 Comparison of socware to email spam

Socware keywords exhibit little (10%) overlap with spam email keywords. As we saw earlier in Section 4, spam keyword score is a key feature in MyPageKeeper's classifier. Therefore, in the final section of our analysis, we investigate the overlap in 'spam keywords' that we observe in socware on Facebook with those seen in another medium targeted by spammers, specifically email.

Socware word	Likelihood ratio	Spam email word	Likelihood ratio
free	12.1	money	11.5
< 3	∞	price	26.6
iphone	∞	free	0.08
awesome	31.3	account	9.6
win	24.3	stock	9.7
wow	90.8	address	5.2
hurry	36.8	bank	56.4
omg	332.3	pills	∞
amazing	4.9	viagra	∞
deal	1.9	watch	1.9

Table 10: Top keywords from socware posts and spam emails.

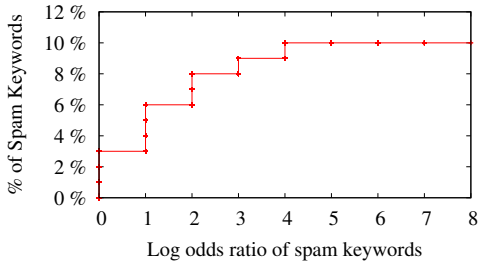


Figure 9: Overlap of keywords between email and Facebook.

We investigate whether spammers use similar keywords on Facebook as they use in email spam.

To perform this analysis, we collected over 17,000 spam emails from [50]. For Facebook spam, we use 92,493 socware posts collected by MyPageKeeper. We transform posts in either dataset to a bag of words with their frequency of occurrence. Similar to [54], we then compute the log odds ratio for each keyword to determine its overlap in Facebook socware and spam email. Here, the log odds ratio for a keyword is defined by $ratio = |log(p_1 q_2 / p_2 q_1)|$ where p_i is the likelihood of that keyword appearing in set i and $q_i = 1 - p_i$. A value of 0 for the log odds ratio indicates that the keyword is equally likely to appear in both datasets, whereas an infinite ratio indicates that the keyword appears in only one of the datasets. In Figure 9 (infinite values are omitted), we see only a 10% overlap in spam keywords between email and Facebook. This indicates that Facebook spam significantly differs from traditional email spam.

Further, Table 10 shows the likelihood ratio (defined earlier in Section 3.3) for the top keywords in either dataset. The higher the likelihood ratio of a socware keyword, the stronger the bias of the keyword appearing more in Facebook socware than in email spam; an infinite ratio implies the keyword exclusively appears in Facebook socware. The word *'omg'* is 332 times more likely to be used in Facebook socware than in email spam. On the other hand, words such as *'pills'* and *'viagra'* are restricted solely to email spam.

6 Like-as-a-Service

Facebook has now become the premier online destination on the Internet. Over 900 million users, half of whom visit the site daily, spend over 4 hours on the site every

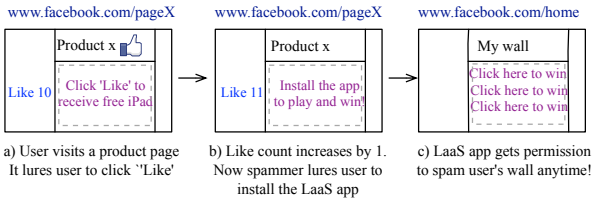


Figure 10: A representation of how a Like-as-a-service Facebook application collects Likes for its client's page and gains access to the user's wall for spamming. Dotted region of the page is controlled by the spammer.

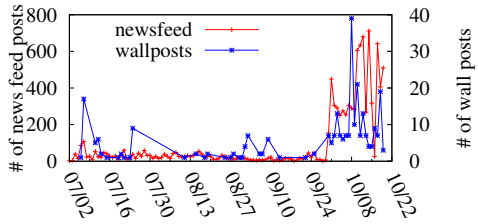


Figure 11: Timeline of posts made by the Games LaaS Facebook application seen on users' walls and news feeds.

month [10]. To leverage user activity on Facebook, an increasingly large number of businesses have Facebook pages associated with their products. However, attracting users to their page is a challenge for any business. One way of doing so is to make users who visit a Facebook page click the 'Like' button on the page. A large number of Likes has two significant implications. First, the number of Likes associated with a page has begun to represent the reputation associated with a page, e.g., a higher number of Likes improves the page's rank in Bing [3]. Second, a link to the product page appears in the news feed of the friends of the user who clicked Like on the page, thus enabling the link to the page to spread on Facebook.

Based on our view of Facebook socware through the MyPageKeeper lens, we see an emerging Like-as-a-Service [3] market to help businesses attract users to their pages. We identify several Facebook apps (e.g., 'Games' [15], 'FanOffer' [13], and 'Latest Promotions' [21]) which are hired by the owners of Facebook pages to help increase the number of Likes on their pages. These applications, which offer Likes as a service, presumably get paid on a 'Pay-per-Like' model by the owners of Facebook pages that make use of their services.

Figure 10 shows how a Like-as-a-Service (LaaS) application typically works. First, a customer of the LaaS application integrates the application into their Facebook page. When users visit the page, the LaaS application entices the user to click Like on page. Typically, the re-

[3] Note that 'Like-as-a-Service' differs from 'Likejacking' [22], where users are tricked into clicking the Like button without them realizing they are doing so, e.g., by enticing the user to click on a Flash video, within which the Like button is hidden.

Page Name	Application Message	No. of Likes
Raging Bid	Just got a better score on Raging Bid's Bouncing Balls contest and I am now in 12297th place. I am getting closer to winning a Sony Bravia 3D HDTV. Who thinks they can beat my score? Click here to try: URL	168,815
www.WalkerToyota.com	DAILY CONTEST UPDATE: I am currently in 7573rd place in Walker Toyota's Tetris contest. There is still plenty of time to try and win a 16GB iPad2. Who thinks they can get a better score than me? Click here to try: URL	136,212
Chip Banks Chevrolet Buick	DAILY CONTEST UPDATE: I am currently in 310th place in Chip Banks Chevrolet Buick's Gem Swap II contest. There is still plenty of time to try and win a 16GB iPad2. Who thinks they can get a better score than me? Click here to try: URL	2,190
Casey Jamerson	DAILY CONTEST UPDATE: I am currently in 6234th place in Casey Jamerson Music's Gem Swap II contest. There is still plenty of time to try and win a 16GB iPad2. Who thinks they can get a better score than me? Click here to try: URL	47,496
Tara Gray	DAILY CONTEST UPDATE: I am currently in 10213th place in Tara Gray's Gem Swap II contest. There is still plenty of time to try and win a Burma Ruby Ring. Who thinks they can get a better score than me? Click here to try: URL	231,035

Table 11: Five example Facebook pages integrated with the Games LaaS application to spam users' walls for propagation.

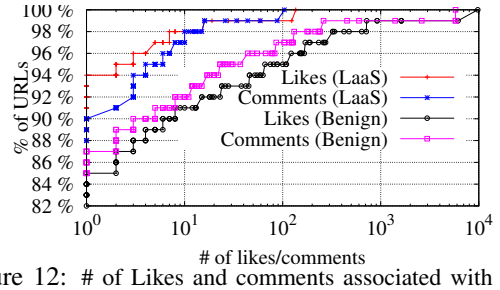


Figure 12: # of Likes and comments associated with URLs posted by the Games Facebook app.

ward promised to the user in return for his Like is that the user can play some games on the page or have a chance of winning free products. However, once the user clicks Like on the page to access the promised reward, the LaaS application then demands that the user add the application to his profile in order to proceed further. In the process of getting the user to add the LaaS application, the application requests the user to grant permission for it to post on the user's wall. Once the application obtains such permissions, it periodically spams the user's wall with posts that contain links to the Facebook page of the customer who enrolled the LaaS application for its services. These posts will appear in the news feeds of the unsuspecting user's friends, who in turn may visit the Facebook page and go through the same cycle again. The LaaS application thus enables the Facebook pages of its customers to accumulate Likes and increase their reputation, even though users are clicking Like on these pages with the promise of false rewards rather than because they like the products advertised on the page.

Here, we analyze the activity of one such LaaS application—Games [15]. Figure 11 shows that posts made by this application appear regularly in the walls and news feeds of MyPageKeeper's users. Even with our small sample of roughly 12K users from Facebook's total population of over 850 million users, we see that 40 users have posts made by Games on their walls, which implies that these users have installed the application and granted it permission to make posts on their wall at any time. We also see that the number of users who installed Games significantly rose around mid-September 2011. Further, from the news feeds of MyPageKeeper users, we see that Games posted links to as many as 700 Facebook pages on a single day; each link points to the Facebook page of a different customer of this LaaS application. Table 11 shows the posts made by Games for some of its customers, the variation in text messages across these posts, and the large number of Likes garnered by the Facebook pages of these customers.

We next analyze the Likes and comments received by 721 URLs posted by the Games app. As shown in Figure 12, we see that over 95% of these URLs have less than 100 Likes and less than 100 comments; this fraction is significantly lesser on a dataset of randomly chosen 721 URLs from benign posts. However, over 20% of the URLs posted by the Games app do receive Likes and comments, thus enabling them to propagate on Facebook. Real users may be unknowingly helping to spreading spam in these cases; such users have been previously referred to as creepers [52].

7 Discussion

Client-based solution. An alternative to MyPageKeeper's server-side detection of socware would be to identify socware on client machines. In such an approach, a client-side tool can classify a post at the instant when the user accesses the post. However, we choose not to use such an approach for multiple reasons. First, a server-side solution is more amenable to adoption; it is easier to convince users to add an app to their Facebook profile than to convince them to download and install an application or browser extension on their machines. Second, users can access Facebook from a range of browsers and even from different device types (e.g., mobile phones). Developing and maintaining client-side tools for all of these platforms is onerous. Finally, and most importantly, many of the features used by our socware classifier (e.g., message similarity score) fundamentally depend on aggregating information across users. Therefore, a view of Facebook from the perspective of a

single client may be insufficient to identify socware accurately.

Estimating false negatives. While we evaluated the accuracy of socware identified by MyPageKeeper by cross-validating with other techniques, evaluating the accuracy of MyPageKeeper's classifier in cases where it declares a URL safe is much harder. Not only do we lack ground truth, but since the highly common case is that a Facebook post is benign, manual verification of a randomly chosen subset of the classifier's negative outputs is insufficient.

We therefore evaluate whether MyPageKeeper's classifier misses any socware by using data from user-reported samples of socware. As shown in Table 3, 533 distinct MyPageKeeper users have submitted 679 such reports and we have received 333 unique URLs across these reports. Based on manual verification, we find that 296 of these 333 URLs indeed point to spam or malware. The remaining 37 URLs point to sites like `surveymonkey.com` (fill out surveys) and `clixsense.com` (get paid to view advertisements), which though abused by spammers have legitimate uses as well. We suspect that our users did come across socware, but reported the URL of the landing page, rather than the URL that they originally found in a socware post.

Of the 296 instances of true socware reported by users, MyPageKeeper's classifier flagged all but 17 of them, independently of users reporting them to us. This translates into a false negative rate of 5% for the classifier. However, 16 of these 17 URLs had been found to match against one of the URL blacklists used by MyPageKeeper. Thus, the false negative rate for the whole MyPageKeeper system, which combines blacklists and the classifier to detect socware, is 0.3%.

Arms race with spammers. Though our current techniques seem to suffice to accurately identify socware on Facebook, we speculate here on how spammers may evolve socware, given the knowledge of how MyPageKeeper works. One option for spammers to evade MyPageKeeper is to use different shortened URLs for a single malicious landing URL. In such cases, MyPageKeeper would consider every posted shortened URL seperately even though they are all part of the same campaign. Thus, if any of these shortened URLs does not appear on the walls/news feeds of several users, MyPageKeeper may fail to flag it. Another option for socware to evade MyPageKeeper is for spammers to slow down its rate of propagation; as we found in Section 4.2, MyPageKeeper sometimes misses socware which is observed only a few times in our dataset. However, slowing down a socware epidemic makes it likely that it will be flagged by other techniques, such as URL blacklists. Moreover, spammers may often be unable to control how fast a socware epidemic spreads. In the case where an epidemic spreads by luring users into installing a Facebook app, the spammer can control how often the app posts spam on the user's wall. However, in cases where users are asked to 'Like' or 'Share' a post to access a fake reward, the socware is self-propagating and its viral spread cannot be controlled by spammers.

Another option is for spammers to change the keywords that they use in socware posts, thus affecting the spam keyword score used by MyPageKeeper's classifier. Though spammers are constrained in their choice of keywords by the need to attract users, some of the keywords may evolve over time as popular colloquial expressions (e.g., 'OMG') change. To evaluate MyPageKeeper's ability to cope with such change, we identified the top keywords (those with high likelihood ratio compared to benign posts among frequently occurring keywords) distinctive to user-reported socware posts. We find that the spam keywords that we use in MyPageKeeper's classifier (identified from manually identified samples of socware) match those computed here. Though this captures data only across four months, MyPageKeeper can similarly recompute the set of spam keywords over time.

8 Related Work

Motivated by the increasing presence of spam and malware on OSNs, there have been several recent related efforts. Here, we contrast our work with these prior efforts.

Studies of spam on OSNs. Gao et al. [43] analyzed posts on the walls of 3.5 million Facebook users and showed that 10% of links posted on Facebook walls are spam, with a large majority pointing to phishing sites. They also presented techniques to identify compromised accounts and spam campaigns. In a similar study on Twitter, Grier et al. [44] showed that at least 8% of links posted on Twitter are spam while 86% of the involved accounts are compromised. In contrast to this study, Thomas et al. [55] show that the majority of suspended accounts in Twitter are created by spammers as opposed to compromised users. All of these efforts however focus on post-mortem analysis of historical OSN data and are not applicable to MyPageKeeper's goal of identifying socware soon after it appears on a user's wall or news feed.

Detecting spam accounts. Benevenuto et al. [39] and Yang et al. [57] developed techniques to identify accounts of spammers on Twitter. Others have proposed a honeypot based approach [53, 47] to detect spam accounts on OSNs. Yardi et al. [58] analyzed behavioral patterns among spam accounts in Twitter. Instead of focusing on accounts created by spammers, MyPageKeeper enables socware detection on the walls and news feeds of legitimate Facebook users.

Real-time spam detection in OSNs. Thomas et al. [54] developed Monarch, a real-time system that

crawls URLs submitted from services such as Twitter to determine whether a URL directs to spam. Monarch relies on the network and domain level properties of URLs as well as the content of the web pages obtained when URLs are crawled. Interestingly, Monarch's classification accuracy is shown to be independent of the social context on Twitter. MyPageKeeper distinguishes itself from Monarch in several ways—1) we study socware on Facebook, which we see significantly differs in its characteristics from traditional spam messages, 2) to make MyPageKeeper efficient, our socware classifier operates without crawling of links found in posts, and 3) we find that the use of social context based features is crucial to efficient detection of socware. In another study, Gao et al. [42] perform online spam filtering on OSNs using incremental clustering. Their technique however relies on having the whole social graph as input, and so, is usable only by the OSN provider. MyPageKeeper instead relies only on the view of the OSN as seen by MyPageKeeper's users. Lee et al. [48] built Warningbird, a system to detect suspicious URLs in Twitter; their system however relies on following the HTTP redirection chains of URLs, thus making their approach less efficient than MyPageKeeper.

Wang et al. [56] propose a unified spam detection framework that works across all OSNs, but they do not have an implementation of such a system in practice. Stein et al. [52] describe Facebook's Immune System (FIS), a scalable real-time adversarial learning system deployed in Facebook to protect users from malicious activities. However, Stein et al. provide only a high-level overview about threats to the Facebook graph and do not provide any analysis of the system. Similarly, other Facebook applications [6, 25, 4] that defend users against spam and malware are proprietary with no details available about how they work. Abu-Nimeh et al. [37] analyze the URLs flagged by one of these applications, Defenseio, but they do not discuss Defenseio's classification techniques and their analysis is restricted to that of the hosting infrastructure (country and ASN) underlying Facebook spam. To the best of our knowledge, we are the first to provide classification of socware on Facebook that relies solely on social context based features, thus enabling MyPageKeeper to efficiently detect socware at scale.

Social context based email spam. Jagatic et al. [45] discuss how email phishing attacks can be launched by using publicly available personal information (e.g., birthday) from social networks, and Brown et al. [40] analyzed such email spam seen in practice. However, due to revisions in Facebook's privacy policy over the last couple of years, only a user's friends have access to such information from the user's profile, thus making such email spam no longer possible. Further, MyPageKeeper focuses on spam propagated on Facebook rather than via email.

9 Conclusions

Facebook is becoming the new epicenter of the web, and we showed that hackers are adapting to this change by designing new types of malware suited to this platform, which we call socware. In this paper, we presented the design and implementation of MyPageKeeper, a Facebook application that can accurately and efficiently identify socware at scale. Using data from over 12K Facebook users, we found that the reach of socware is widespread and that a significant fraction of socware is hosted on Facebook itself. We also showed that existing defenses, such as URL blacklists, are ill-suited for identifying socware, and that socware significantly differs from email spam. Finally, we identified a new trend in aggressive marketing of Facebook pages using "Like-as-a-Service" applications that spam users to make money based on a "Pay-per-Like" model.

References

[1] Anti-phishing working group. http://www.antiphishing.org/.

[2] Application authentication flow using oauth 2.0. http://developers.facebook.com/docs/authentication/.

[3] Bing gets friendlier with Facebook. http://www.technologyreview.com/web/37585/.

[4] Bitdefender Safego. http://www.facebook.com/bitdefender.safego.

[5] bit.ly API. http://code.google.com/p/bitly-api/wiki/ApiDocumentation.

[6] Defensio Social Web Security. http://www.facebook.com/apps/application.php?id=177000755670.

[7] Escrow-fraud. http://escrow-fraud.com/.

[8] Experts: Facebook crime is on the rise. http://www.zdnet.com/blog/facebook/experts-facebook-crime-is-on-the-rise/2632.

[9] Facebook birthday T-shirt scam steals secret mobile email addresses. http://bit.ly/Kvax0t.

[10] Facebook is the web's ultimate timesink. http://mashable.com/2010/02/16/facebook-nielsen-stats/.

[11] Facebook Phishing Scam Costs Victims Thousands of Dollars. http://www.hyphenet.com/blog/2011/10/04/facebook-phishing-scam-costs-victims-thousands-of-dollars/.

[12] Facebook scam involves money transfers to the Philippines. http://profitscam.com/facebook-scam-involves-money-transfers-to-the-philippines-post/.

[13] Fan Offer. https://www.facebook.com/apps/application.php?id=107611949261673.

[14] FBML- Facebook Markup Language. https://developers.facebook.com/docs/reference/fbml/.

[15] Games. https://www.facebook.com/apps/application.php?id=121297667915814.
[16] goo.gl API. http://code.google.com/apis/urlshortener/v1/getting_started.html.
[17] Google Safe Browsing API. http://code.google.com/apis/safebrowsing/.
[18] Hackers selling $25 toolkit to create malicious Facebook apps. http://zd.net/M2WNe1.
[19] How to spot a Facebook Survey Scam. http://facecrooks.com/Safety-Center/Scam-Watch/How-to-spot-a-Facebook-Survey-Scam.html.
[20] Joewein: Fighting spam and scams on the Internet. http://www.joewein.net/.
[21] Latest Promotions. https://www.facebook.com/apps/application.php?id=174789949246851.
[22] Likejacking takes off on Facebook. http://www.readwriteweb.com/archives/likejacking_takes_off_on_facebook.php.
[23] MalwarePatrol- Malware is everywhere! . http://www.malware.com.br/.
[24] MyPageKeeper. https://www.facebook.com/apps/application.php?id=167087893342260.
[25] Norton Safe Web. http://www.facebook.com/apps/application.php?id=310877173418.
[26] Phishtank. http://www.phishtank.com/.
[27] Rihanna video scam. "http://www.virteacon.com/2011/11/sick-i-just-hate-rihanna-after-watching.html".
[28] Spamcop. http://www.spamcop.net/.
[29] Spamhaus. http://www.spamhaus.org/sbl/index.lasso.
[30] Steve Jobs death scams are just the greedy exploiting the gullible. http://bit.ly/M67Zme.
[31] SURBL. http://www.surbl.org/.
[32] SVM Tutorials. http://svms.org/tutorials/.
[33] URIBL. http://www.uribl.com/.
[34] Web-of-trust. http://www.mywot.com/.
[35] What 20 Minutes On Facebook Looks Like. http://tcrn.ch/KytqzB.
[36] Facebook becomes partner with Web of Trust (WOT). https://www.facebook.com/notes/facebook-security/keeping-you-safe-from-scams-and-spam/10150174826745766, May 2011.
[37] S. Abu-Nimeh, T. M. Chen, and O. Alzubi. Malicious and spam posts in online social networks. In *IEEE Computer Society*, 2011.
[38] F. becomes partner with WebSense. http://www.thetechherald.com/article.php/201139/7675/Facebook-implements-malicious-link-scanning-service, Oct 2011.
[39] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on Twitter. In *CEAS*, 2010.
[40] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders. Social networks and context-aware spam. In *ACM CSCW*, 2008.
[41] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2, 2011.
[42] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary. Towards online spam filtering in social networks. 2012.
[43] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *IMC*, 2010.
[44] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The underground on 140 characters or less. In *CCS*, 2010.
[45] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Commun. ACM*, 2007.
[46] A. Le, A. Markopoulou, and M. Faloutsos. Phishdef: Url names say it all. In *Infocom*, 2010.
[47] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots + machine learning. In *SIGIR*, 2010.
[48] S. Lee and J. Kim. Warningbird: Detecting suspicious urls in twitter stream. 2012.
[49] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious urls. In *KDD*, 2009.
[50] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive bayes – which naive bayes? In *CEAS*, 2006.
[51] Z. Qian, Z. M. Mao, Y. Xie, and F. Yu. On network-level clusters for spam detection. In *NDSS*, 2010.
[52] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *SNS*, 2011.
[53] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *ACSAC*, 2010.
[54] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and Evaluation of a Real-Time URL Spam Filtering Service. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2011.
[55] K. Thomas, C. Grier, V. Paxson, and D. Song. Suspended accounts in retrospect: An analysis of twitter spam. In *IMC*, 2011.
[56] D. Wang, D. Irani, and C. Pu. A social-spam detection framework. In *CEAS*, 2011.
[57] 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.
[58] S. Yardi, D. Romero, G. Schoenebeck, et al. Detecting spam in a twitter network. *First Monday*, 2009.