Measurement and Early Detection of Third-Party Application Abuse on Twitter

Shehroze Farooqi
The University of Iowa

Zubair Shafiq
The University of Iowa

ABSTRACT

Third-party applications present a convenient way for attackers to orchestrate a large number of fake and compromised accounts on popular online social networks. Despite recent high-profile reports of third-party application abuse on popular online social networks, prior work lacks automated approaches for accurate and early detection of abusive applications. In this paper, we perform a longitudinal study of abusive third-party applications on Twitter that perform a variety of malicious and spam activities in violation of Twitter's Terms of Service (ToS). Our measurements spanning over a period of 16 months demonstrate an ongoing arms race between attackers continuously registering and abusing new applications and Twitter trying to detect them. We find that hundreds of thousands of abusive applications remain undetected by Twitter for several months while posting tens of millions of tweets. We propose a machine learning approach for accurate and early detection of abusive Twitter applications by analyzing their first few tweets. The evaluation shows that our machine learning approach can accurately detect abusive application with 92.7% precision and 87.0% recall by analyzing their first seven tweets. The deployment of our machine learning approach in the wild shows that attackers continue to abuse third-party applications despite Twitter's recent countermeasures targeting third-party applications.

CCS CONCEPTS

• **Security and privacy → Social network security and privacy**.

KEYWORDS

Abuse; API; Online Social Networks; Spam; Twitter

ACM Reference Format:
Shehroze Farooqi and Zubair Shafiq. 2019. Measurement and Early Detection of Third-Party Application Abuse on Twitter. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313515

1 INTRODUCTION

Background. Popular social networking sites, including Twitter, allow developers to use third-party applications to enhance user experience. Millions of applications use Twitter's third-party developer platform to support news, gaming, entertainment, analytics, research, and publishing solutions [2]. Third-party Twitter applications (or simply Twitter applications) use OAuth [33] for getting permissions from users to read/write/message on their behalf [11, 15]. Twitter applications have perpetual access to user accounts unless users explicitly revoke their permissions. Naturally, an attacker can control a large number of accounts by tricking users into installing a malicious application [45] or compromising a popular legitimate application [36, 51].

Motivation. Third-party Twitter applications present a convenient way for attackers to orchestrate fake or compromised accounts through Twitter API [12]. Attackers can install third-party applications on fake accounts that they themselves create or buy in bulk from underground marketplaces [35, 58]. Attackers can also trick users (e.g., phishing [27], malicious browser extensions [41]) into installing their applications to compromise accounts. Attackers can even recruit real users on crowdturfing marketplaces to install their applications in exchange for monetary and non-monetary incentives (e.g., free followers) [36, 58, 60]. On several occasions during the last couple of years, Twitter has disclosed large-scale abuse by hundreds of thousands of third-party applications on their platform [7, 10, 45].

Limitations of Prior Art. Prior research has paid little attention to directly mitigate the role of third-party applications in propagating malware and spam on Twitter [34, 43, 52, 54, 55, 55, 57, 61]. While some prior research has reported the spread of malware and spam by third-party Twitter applications [34, 52, 55, 57], most efforts are focused on detecting the *sources* (fake and compromised accounts) and *targets* (retweets) of malicious activities on Twitter. We believe that directly targeting such abusive third-party Twitter applications is crucial for robust detection of increasingly sophisticated malicious activity on Twitter. Our belief is in line with Twitter's recently announced plans to target fake, coordinated, and automated account activities conducted by third-party applications on their platform [9, 10].

Measuring abusive Twitter Applications. In this work, we conduct a 16-month long longitudinal measurement study of abusive third-party applications that perform a variety of malicious and spam activities in violation of Twitter's Terms of Service (ToS) [20]. To collect a comprehensive ground truth of abusive Twitter applications, we retrospectively check whether tweets by third-party applications are removed by users or Twitter's abuse detection systems [5, 8, 58]. Prior work has also leveraged retrospective analysis of deleted/suspended tweets/accounts to study spam and malware campaigns on Twitter [56, 57]. We are able to identify 167,013 abusive third-party Twitter applications through retrospective analysis of tweets collected over a period of 16 months using Twitter's APIs [12, 17]. Our measurements reveal that there is an ongoing

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313515

arms race between attackers registering and abusing new applications and Twitter actively trying to detect and remove them. More specifically, we show that attackers are able to use a large pool of hundreds of thousands of abusive applications to post tens of millions of tweets. Abusive applications often evade detection for several months while posting millions of tweets for astroturfing [32, 52, 55], phone spam [39], and tricking users with deceiving claims to compromise accounts [4, 26, 57].

Early Detection of Abusive Twitter Applications. Accurate and early detection of abusive third-party applications can help in significantly mitigating malware and spam on Twitter. To this end, we propose a machine learning approach for the early detection of abusive third-party Twitter applications by analyzing their first few tweets. Specifically, we extract a variety of user-based (e.g., account age) and tweet-based features (e.g., retweets to tweets ratio) on the first-$k \in \{2, 3, ..., 25\}$ tweets to train a supervised machine learning classification model to distinguish between abusive and benign Twitter applications as early as possible. We implement and evaluate our machine learning approach before and after Twitter's recent countermeasures targeting third-party application abuse [9]. The evaluation shows that our machine learning approach is able to accurately detect abusive applications with 92.7% precision and 87.0% recall. We also show that our machine learning approach detects abusive applications months before Twitter does, during which time they are able to post tens of millions of tweets.

Key Contributions. We summarize our contributions as follows.

• *Longitudinal Measurement Study of Abusive Applications on Twitter.* We perform a longitudinal measurement study to establish a ground truth of abusive Twitter applications that captures diverse malicious and spamming behaviors. We showed that these abusive applications stay undetected for a long time posting tens of millions of tweets despite Twitter's ongoing efforts to detect them.

• *Machine Learning Approach for Early Detection.* We propose a machine learning approach to accurately and early detect these abusive applications by analyzing their first-$k \in \{2, 3, ..., 25\}$ tweets. We select $k = 7$ as a suitable trade-off between classification accuracy and early detection. Our machine learning model detects abusive applications with a precision of 92.7% and a recall of 87.0% using 10-fold cross validation as soon as they post their first seven tweets. We show that our model detects a large fraction of these abusive applications several months before Twitter detects them while they post tens of millions of tweets during this time period.

• *In The Wild Deployment.* The deployment of our machine learning model in the wild shows that attackers are still able to register and abuse third-party applications despite Twitter's new countermeasures [9]. We show that our machine learning model accurately detects these abusive third-party applications as soon as they post their first seven tweets while they evade detection by Twitter for a long time. Finally, we show that our machine learning model detects a large fraction of new abusive applications that are missed by Twitter's existing abuse detection systems.

We have disclosed our findings to Twitter's Site Integrity team, who is actively trying to mitigate abuse of third-party applications on their platform [9]. Our machine learning approach can complement Twitter's abuse detection systems for accurate and early detection of Twitter API abuse by third-party applications.

2 BACKGROUND

In this section, we first provide an overview of third-party application support on popular online social networks. We then discuss our threat model for third-party applications and the prevalence of their abuse on Twitter.

2.1 Third-Party Applications

Online social networks provide APIs to develop third-party applications such as games, entertainment, education, and utilities. To allow third-party application development, online social networks implement authorization frameworks such as OAuth [33]. For example, Twitter uses the OAuth 1.0a authorization framework [33], which enables third-party applications to gain access to Twitter's streaming and REST APIs as well as Twitter's Single Sign-On (SSO) service [12, 19]. When creating a new third-party application, developers have to specify a set of permissions required from users who would install the application on their accounts. OAuth supports both *read* and *write* permissions. The read permissions allow a third-party application to retrieve data (e.g., timeline tweets/posts, list of followers/friends) from a user's account. The write permissions allow a third-party application to perform write actions (e.g., posting tweets/posts, following users or liking pages) on a user's behalf. Popular online social networks such as Twitter and Facebook have millions of third-party applications that are regularly used by hundreds of millions of users [3, 23].

2.2 Threat Model

While third-party applications are widely used for benign purposes, unfortunately, they can also be exploited by attackers to compromise and orchestrate a large number of accounts for nefarious purposes. Prior work has reported several instances of widespread abuse of third-party applications for spreading spam and malware on online social networks [34, 52, 55, 57].

The typical modus operandi of attackers is as follows. Attackers register a new third-party application with the aim of installing it on as many fake/compromised accounts as possible. Attackers install the registered application on fake accounts that they themselves create or buy in bulk from underground marketplaces [35, 58]. Attackers may also compromise an account by tricking its user into installing the application. After installing the application on a sufficient number of accounts, attackers can use the access tokens [1] via the APIs to conduct malicious activities at scale.

The abuse of third-party applications has been shown time and again on popular online social networks [36, 50, 52, 55, 57]. For example, prior work reported the abuse of third-party applications to escalate the reputation of a target account by retweeting/liking/following from compromised/fake accounts on Twitter [52, 55] and Facebook [36]. Prior work has also reported the abuse of third-party applications to run spam or malware campaigns from compromised/fake accounts on Twitter [57] and Facebook [50].

In this paper, we specifically focus on investigating abuse by third-party applications on Twitter. It is noteworthy that Twitter recently disclosed a large-scale abuse of third-party applications on their platform [10] in the aftermath of a congressional investigation into Russian interference in the 2016 U.S. election [7]. Specifically, Twitter announced that they removed hundreds of thousands of third-party applications that were abusing their API during 2017 through 2018 [10, 45]. Also noteworthy is Twitter's recently announced policy to vet new third-party applications at registration [9]. However, despite Twitter's existing detection systems and new countermeasures, we will show later that attackers continue to abuse third-party applications on Twitter to this day.

2.3 Abusive Twitter Applications

We refer to a third-party Twitter application as "abusive" if it violates Twitter's rules [20]. The violations of these rules mostly include malicious and spammy behaviors such as posting links to malicious content, aggressive following and un-following behavior, abusing reply or mention function, hijack trending topics or hashtags, duplicate updates, etc. Other violations of these rules, unrelated to malicious and spammy behaviors, include the unlawful use of Twitter platform such as illegitimate distribution of copyrighted or hacked material, graphical violence, and harassment.

It is challenging to manually establish the ground truth for third-party Twitter applications (e.g., manual detection of the violation of Twitter's rules) because of the scale and diversity of abusive behavior. To automatically get a comprehensive ground truth of abusive Twitter applications, we retrospectively check whether tweets by third-party applications are removed by users or Twitter's abuse detection systems due to the violation of their rules [20]. Prior work has exploited similar retrospective analysis of deleted tweets and suspended accounts to study spam and malware campaigns on Twitter by fake/compromised accounts [56, 57]. In particular, if all tweets by a Twitter application are removed, it is highly likely that the application is violating Twitter's rules and can be labeled as abusive. In addition to retrospective ground truth labeling of abusive Twitter applications, we further expand our ground truth out-of-band by including follower market applications that compromise accounts by incentivizing users to install the applications in exchange for "free followers" [13, 55]. We next explain our methodology to collect data of these abusive applications.

3 MEASURING ABUSIVE APPLICATIONS

In this section, we conduct a longitudinal measurement study of third-party Twitter applications to demonstrate that abusive applications are able to evade detection by Twitter's abuse detection systems for extended time periods while they post tens of millions of tweets during this time.

3.1 Data collection

We leverage Twitter's streaming API to gather tweets by different third-party Twitter applications and REST API to establish a ground truth of benign and abusive applications.

Streaming API. We rely on Twitter's streaming API to collect publicly available tweets. Twitter's streaming API offers a `sample` stream that returns 1% sample of all public tweets [48]. Each tweet contains the tweet's text and metadata, which includes timestamp, user's screen name, and the source field that contains the name of the application used to post the tweet. We collect 1.5 billion tweets by 112 million users from 456,987 applications from September 2016 to December 2017. We refer to this collection of tweets as the *Twitter sample dataset.*

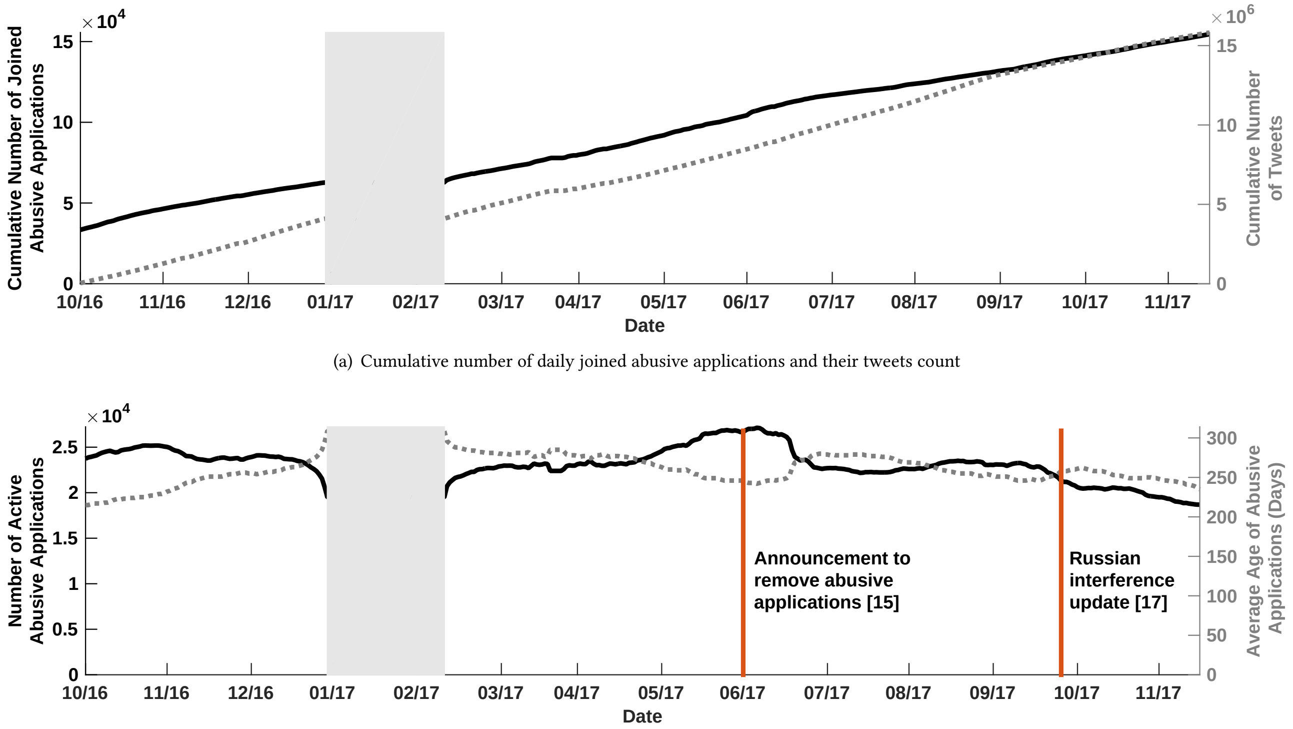
Retrospective Analysis of Removed Tweets. We retrospectively query the current status of tweets of all third-party applications in the Twitter sample dataset to check whether they are removed using Twitter's REST API. We provide sufficient time to Twitter's abuse detection systems to remove tweets by abusive third-party applications. Specifically, we start querying Twitter's REST API to check the deletion status of tweets in August 2018, which is at least 8 months apart from the tweets in the Twitter sample dataset. Note that we cannot query the deletion status of tweets for all applications due to the rate limits imposed by Twitter's REST API [16]. Hence, for each application, we select a random sample of at most 100 tweets whose deletion status is queried using Twitter's REST API. Since 87% of applications have less than or equal to 100 tweets, we sample tweets for only 13% applications and consider all tweets of the remaining applications. In total, we query the deletion status of 12 million tweets posted by 456,987 applications of which 36% tweets are removed. 49% of applications have no removed tweets while 37% applications have all of their tweets removed. To minimize false positives in our labeling, we conservatively label the 37% applications with all of their tweets removed as abusive. We next explain our methodology to identify and crawl follower markets to expand our ground truth of abusive applications.

Follower Markets. We query Google and Twitter search to identify follower markets. First, we search Google using keywords such as "free followers" and "increase followers". We manually analyze search results to identify popular follower markets. Second, we search Twitter using hashtags such as "followers" and "increase-Followers". We manually analyze URLs in tweets to find follower markets. Using this methodology, we are able to identify 50 follower markets that ask Twitter users to install third-party applications in exchange for "free followers". Our eyeball analysis shows that abusive applications used by follower markets change over time. Therefore, we periodically crawl follower markets to extract the names of their abusive applications. To automate this process, we use Selenium WebDriver [18] to open each follower market website every 15 minutes. Upon clicking the sign-in button to install the application, we are redirected to Twitter's application authorization page. We extract the name of the abusive application, without installing it, from the authorization page. We crawl these 50 follower markets from September 2016 to August 2017 and identify names of 14,150 distinct abusive applications. Out of these 14,150, we find 6,437 abusive applications in our Twitter sample dataset.

Table 1 summarizes the statistics of third-party Twitter applications in our Twitter sample dataset. In total, we are able to identify 168,227 distinct abusive applications through retrospective analysis of removed tweets and crawling follower markets.

3.2 Arms Race

We investigate the arms race between attackers continuously creating new abusive applications and Twitter trying to detect and remove them [10, 45]. Specifically, we study how quickly Twitter detects and removes abusive applications while they post tens of



(a) Cumulative number of daily joined abusive applications and their tweets count

(b) Count and average of daily active abusive applications

Figure 1: Illustration of the arms race between attackers and Twitter in registering and removing abusive Twitter applications. Attackers use a large pool of abusive applications to post tens of millions of tweets. While Twitter's existing countermeasures detect some abusive applications, attackers always have tens of thousands of active abusive applications that stay undetected for several months.

All Applications	Abusive (Retrospective)	Abusive (Follower markets)	Abusive (Combined)
456,987	167,013	6,437	168,227

Table 1: Summary of abusive applications identified from retrospective analysis of removed tweets and follower markets in our Twitter sample dataset.

millions of tweets observed in the Twitter sample dataset. Figure 1(a) plots the cumulative number of new abusive applications observed every day and the cumulative number of tweets posted by them [1]. We find that attackers use a large pool of applications to post abusive tweets. On average, attackers daily register 392 new abusive applications and post 41,705 tweets. In total, we observe more than 17 million tweets posted by 168,227 abusive applications. Since our Twitter sample dataset is limited to at most 1% sample of daily tweets, the actual number of tweets posted by abusive application is likely higher by roughly two orders of magnitude. Thus, we estimate the number of daily and total tweets posted by abusive applications to be in the order of millions and billions, respectively. Our estimates are close to the recent disclosure by Twitter that mentioned 2.2 billion tweets posted by abusive applications [10].

As Twitter detects and removes some of these abusive applications, we expect attackers to register new applications to make up for the removed applications. To evaluate Twitter's existing abuse detection systems, we plot the distribution of number of daily active abusive applications and their average age in Figure 1(b). Note that we estimate the *age* of an abusive application by calculating the difference between the first time and the last time the application appeared in Twitter sample dataset. We say that an application is removed by Twitter's abuse detection systems [5, 8, 56, 58], when it stops appearing in our dataset. Since there is no definitive way for us to know whether or not an application is removed by Twitter, we optimistically assume that these applications are removed by Twitter. We observe that attackers always have a substantial number of active abusive applications ranging between 18,687-27,132. These active applications stay undetected for a long time with an average age of more than six months.

While recently announced countermeasures by Twitter detect some abusive applications, we note that a vast majority of abusive applications still go undetected for a long time period. Specifically, Twitter announced plans to implement new machine learning based approaches to detect and remove abusive applications in June 2017 [5]. In Figure 1(b), we observe a sharp but small decline in the number of active abusive applications. It is interesting to note that attackers seem to adapt to these new countermeasure and the number of active applications stabilizes by August 2017. We observe another decline in the number of active applications after Twitter announced additional countermeasures against abusive third-party applications in September 2017 [7]. However, we note that a vast

[1] Our data collection stopped during the time period represented by the grey shaded region due to an error in the data collection script.

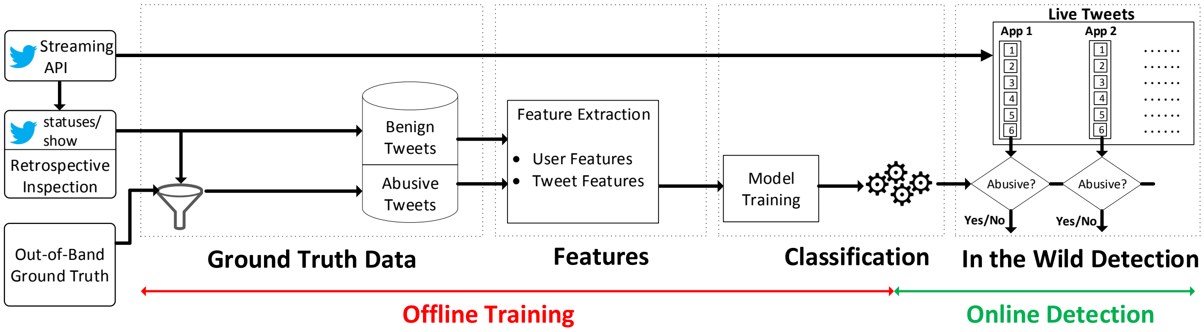


Figure 2: Our proposed approach for early detection of abusive third-party applications on Twitter. In the offline phase, we analyze the first-k tweets of each application to extract user-based and tweet-based features. We then train a supervised machine learning model to classify benign and abusive applications. In the online phase, we use the trained model to detect abusive applications by analyzing their first-k tweets from Twitter's streaming API.

majority of the abusive application still remain active despite Twitter's countermeasures.

It is noteworthy that the applications detected during these sharp declines in Figure 1(b) are relatively new because the average age of active applications increased after the decline. This shows that while Twitter detects some abusive applications, a large number of long-lived abusive applications remain undetected. Specifically, 5,404 abusive applications remain undetected during the 16 month period and are able to post 2.9 million tweets.

Takeaway: Our results indicates an ongoing arms race between attackers and Twitter on the registration and removal of abusive Twitter applications. Essentially, attackers are able to use a large pool of abusive applications to post tens of millions of tweets while being resilient to Twitter's countermeasures. While Twitter's existing countermeasures detect some abusive applications, a large number of abusive applications stay undetected for a long time. As we discuss next, we propose a machine learning approach for the early detection of abusive applications.

4 PROPOSED APPROACH

Since abusive third-party Twitter applications are able to evade detection for a long time, we are interested in detecting these abusive applications as early as possible before they are able to post many tweets. In this section, we present a machine learning approach for early detection of abusive third-party Twitter applications. Figure 2 provides an overview of our approach. In the offline phase, we train a supervised machine learning model that analyzes the first-k tweets of an application to detect abusive applications. More specifically, we extract a variety of user-based and tweet-based features to distinguish between benign and abusive applications. Using a labeled repository of tweets for benign and abusive applications, we then train a supervised machine learning classifier to detect abusive applications. In the online phase, we use the trained supervised machine learning model to detect abusive applications *in the wild* by analyzing their first-k tweets from Twitter's streaming API.

4.1 Ground Truth

Next, we explain our method to establish the ground truth for benign and abusive applications in our Twitter sample dataset. Recall from Section 3.1 that we may strictly label an application as abusive if all of its tweets are removed or benign if none of the tweets are removed. However, this strict definition would result in mislabeling many abusive and benign applications. For instance, a user may remove tweets posted through a benign application due to spelling or grammatical mistakes [22]. Similarly, a user may remove tweets posted through an abusive application after recovering a compromised account [14, 57]. Moreover, Twitter's abuse detection systems may remove a subset of tweets by an abusive application, unrelated to detection of the abusive application [6]. Therefore, we need to relax our labeling criterion from all-or-nothing. To this end, we define two thresholds, α and β, to label an application as abusive or benign, respectively. We label an application as abusive if the percentage of removed tweets is more/less than α/β. On one extreme, we select the value of $\alpha = 90\%$ as 37% of applications have at least 90% of their tweets removed. On the other extreme, we select $\beta = 30\%$ as 58% of applications have less than 30% removed tweets. To reaffirm this selection of β, note that the percentage of removed tweets for several popular benign applications (e.g., Twitter for iPhone, Twitter for Android, Twitter Web Client) is less than 30%. To conclude, we are able to label 95% applications as benign or abusive using 90%-30% α and β selections. Note that we filter applications with only one user/tweet representing less than 2% tweets in our Twitter sample dataset. Overall, our ground truth contains 19,343 benign and 24,588 abusive applications in our Twitter sample dataset.

Next, we present case studies of a select spam and malicious campaigns. We manually analyze the content of a sample of labeled abusive applications to identify these campaigns.

Scam installs. We find several known malicious campaigns that deceive users into installing their abusive applications to compromise accounts and post spam tweets from these accounts. For example, we identify a malicious campaign that claims to inform users who visited their timeline [26, 47]. This is a deceiving claim to trick users into installing abusive applications since Twitter does not provide timeline visit information to third-party applications. As another example, attackers register applications with names that impersonate Twitter (e.g., `Twitter Age Confirmation` and `Twitter Age Verification`) to trick users into installing their abusive applications [4]. We find over 250 abusive applications that are part of such malicious campaigns.

Phone spam campaign. We find a malicious campaign promoting phone spam [39]. In this campaign, spammers mislead victims by making false promises and expect users to contact them on listed phone numbers in the posted tweets. We find 47 abusive applications that are part of this spam campaign.

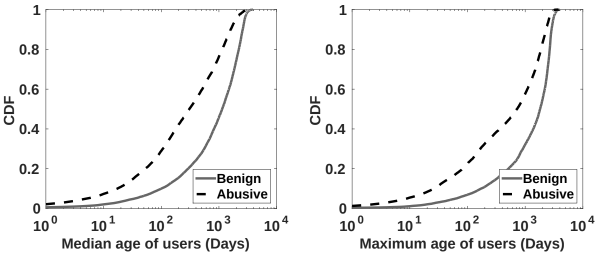
Astroturfing campaigns. We find several astroturfing campaigns that exploit abusive applications to run their operations. Some examples of these campaigns provide fake followers [55] and retweets [52]. We identify thousands of abusive applications that participate in such astroturfing camapaigns.

4.2 Features

We extract a comprehensive set of features to capture distinguishing characteristics of benign and abusive applications. Our feature extraction has two key differences compared to prior work. First, while prior research computed per-tweet features to detect spam/malicious tweets or per-user features to detect fake/compromised user accounts, we compute features on a per-application basis to directly detect abusive applications. Second, unlike prior research, we compute features from the first-k tweets of each application for early detection of abusive applications.

We compute a variety of user-based and tweet-based features to detect abusive applications. For user-based features we focus on following characteristics of users: (1) **number of followers*, (2) **number of followings*, (3) average number of followers to followings ratio, (4) ratio of users with default images set as profile, (5) ratio of verified users, (6) **age of user accounts*, (7) **number of tweets*, and (8) average ratio of total tweets to age of user accounts . For tweet-based features we focus on following characteristics of tweets: (1) ***number of user mentions*, (2) ***number of hashtags*, (3) percentage of tweets with hashtags, (4) percentage of unique hashtags, (5) entropy of hashtags, (6) average of retweet-to-tweet ratio, (7) entropy of URLs, (8) percentage of URLs, and (9) percentage of unique URLs. For many of the user-based and tweet-based features, we compute various summary statistics for italicized features across all tweets of an application. Features with * represent mean, median, minimum, and maximum whereas features with ** represent mean, median, minimum, maximum, and standard deviation. In total, we extract 38 user-based and tweet-based features. We next analyze the effectiveness of a select subset of these features in distinguishing between abusive and benign applications.

Account Age. Figure 3(a) plots the distribution of the median age of user accounts of abusive and benign applications. We note that the user accounts of benign applications are significantly older than those of abusive applications. More specifically, 68% abusive applications have median user account age of two years or less. In contrast, 37% benign applications have median user account age of two years or less. We surmise that the median age of user accounts of abusive applications is low because attackers continuously create fresh user accounts which are, sooner or later, suspended due to violation of Twitter rules [56].



(a) Average

(b) Maximum

Figure 3: We observe that the user accounts of benign applications are older than the user accounts of abusive applications.

Figure 3(b) plots the distribution of maximum user account age for abusive and benign applications. We note that a large fraction of benign applications has at least one account that is very old. Specifically, more than 75% of benign applications have at least one user account aged two years or more. In contrast, more than 50% of abusive applications have no user account aged two years or more. While attackers can obfuscate maximum user account age feature by adding an aged user account, the median age of user accounts is more robust to obfuscation because it relies on the whole user account population of an application.

Retweet-to-tweet Ratio. Figure 4 plots the distribution of retweet-to-tweet ratio of abusive and benign applications. We note that a large fraction of abusive applications posts only retweets while this behavior is quite uncommon among benign applications. Specifically, 32% abusive applications have retweet-to-tweet ratio of 1 whereas only 5% benign application have retweet-to-tweet ratio of 1. Such abusive applications are likely being used to artificially boost reputation of tweets [52]. Attackers can try to post original tweets to obfuscate retweet-to-tweet ratio feature. To generate original tweets, attackers can use duplicate content but this will also be detected due to the violation of Twitter rules. Attackers can also recruit crowdturfing workers [44] to generate organic content to obfuscate ratio of retweet-to-tweet feature but it may prohibitively increase their cost for large-scale spam operation [30].

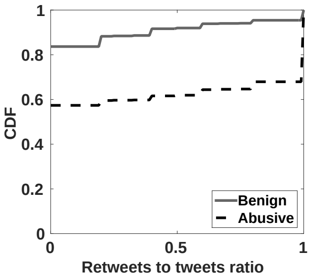


Figure 4: We observe that a large fraction of abusive applications only post retweets while this behavior is quite uncommon among benign applications.

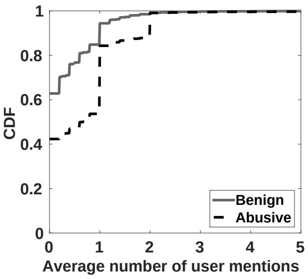


Figure 5: We observe that abusive applications mention more users in their tweets as compared to benign applications.

Number of User Mentions. Figure 5 plots the distribution of average number of user mentions in tweets posted by abusive and benign applications. It is interesting to note that abusive applications mention more users than benign applications. Specifically, 46% abusive applications have an average of one or more user mentions while only 5% benign applications have an average of one or more user mentions. Abusive applications mention more users in tweets to either lure other victims into installing their applications or increase the reach of their tweets. Attackers can try to reduce or stop mentioning users altogether to manipulate their average number of mentions; however, it would negatively impact their ability to reach to other users.

4.3 Classification

We leverage previously discussed user-based and tweet-based features to train a supervised machine learning model to classify an application as abusive or benign. For classification, we tried several classification algorithms using Python's scikit library and ended up selecting the Random Forest classifier because it outperformed other algorithms. For training the model, we use the ground truth of 19,343 benign and 24,588 abusive applications. We first use 10-fold cross-validation to evaluate the accuracy of our trained Random Forest classification model. We then use our trained Random Forest classification model to detect abusive applications in the wild on new data collected from Twitter's streaming API.

5 EVALUATION

In this section, we evaluate the effectiveness of our machine learning approach in detecting abusive applications as early as possible. First, we use cross-validation to study the classification accuracy for varying values of first-k tweets. We show that our machine learning approach is able to detect abusive applications with very high accuracy several weeks before they are detected by Twitter. Second, we employ our machine learning approach to detect abusive applications based on their first few tweets on a new Twitter dataset, which is collected after Twitter's recently announced countermeasures were implemented. We show that attackers still register new abusive applications that go undetected by Twitter while our machine learning approach detects them quite early.

5.1 Cross-Validation

Early Detection. We train and test a Random Forest classifier on varying values of first-$k \in \{2, 3, \ldots, 25\}$ tweets using 10-fold cross-validation. For each value of k, we sample from the set of abusive and benign applications with at least k tweets. Since number of benign applications are slightly less than abusive applications, we randomly sample benign applications for each value of k to match the number of abusive applications to create a balanced dataset for cross-validation. We train and test our model using 100 random samples of benign applications for each value of k and report averages and standard deviations of precision and recall metrics. Figure 6 plots precision and recall as a function of k. We observe the best average precision and recall of 94.7% and 89.7% at $k = 25$, respectively. Recall that our objective is to accurately detect abusive applications as early as possible. We observe that precision and recall improve as k increase but they start to plateau beyond $k = 7$ in Figure 6(a). Specifically, precision increases from 90.9% for $k = 2$ to 92.7% for $k = 7$ and recall increases from 83.2% for $k = 2$ to 87.0% for $k = 7$. Therefore, we select $k = 7$ as a suitable tradeoff between early detection and classification accuracy.

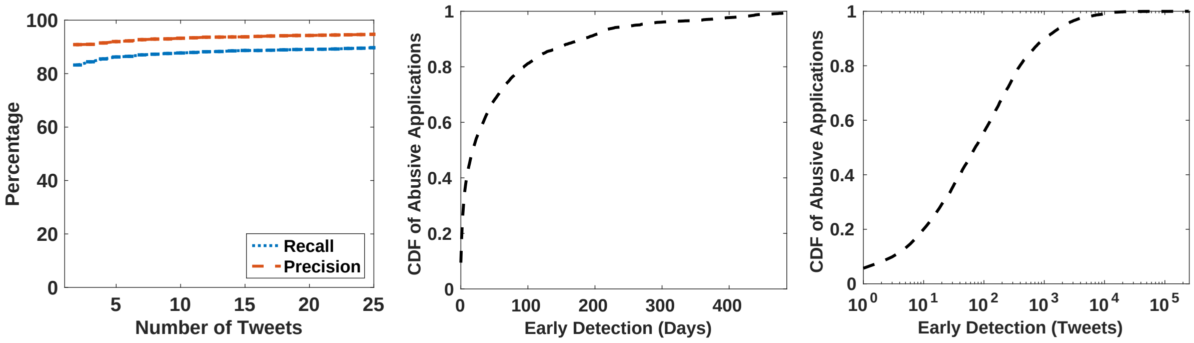
We quantify the early detection of abusive applications correctly detected by our model for $k = 7$ in terms of days and tweets. Early detection in terms of days is defined as the difference between the estimated age of application (defined in Section 3.2) and the age of application when they are detected by our model. Early detection in terms of tweets is defined as the difference between the total number of tweets posted by the applications and the number of tweets posted by applications when they are detected by our model.

Figure 6(b) shows that our trained model is able to detect abusive applications weeks and sometimes months before Twitter does so. Specifically, 42% abusive applications are detected at least a month earlier by our model whereas 21% abusive applications are detected at least 3 months earlier by our model. It is noteworthy that our model has detected 60 abusive applications on the first day of their appearance that otherwise remain undetected throughout the data collection period of 16 months. Figure 6(c) shows that our trained model detects abusive applications before they continue posting hundreds of millions of tweets. Specifically, 45% abusive applications posted 100 or more tweets where 10% abusive applications posted 1000 or more tweets after detection by our model. In total, all abusive applications posted 9,146,439 tweets after detection by our model. Since our Twitter sample dataset is limited to at most 1% sample of daily tweets, the actual number of tweets posted by abusive applications is likely higher by roughly two orders of magnitude. Thus, we estimate that abusive applications posted tweets in the order of hundreds of millions after detection by our model.

We manually analyze a small sample of false positives i.e., benign applications incorrectly classified as abusive by our machine learning model during cross-validation. Our manual analysis uncovers that several applications labeled as benign in our ground truth are in fact abusive. This indicates that our precision is actually more than our estimate reported in Figure 6(a). We surmise that activities of many abusive applications remain undetected by Twitter's existing abuse detection systems during our data collection period; hence, these abusive applications are incorrectly labeled as benign in our ground truth. This also shows that our ground-truth is a conservative estimate of abusive applications. Despite being a conservative estimate, as discussed in Section 4.1, we argue that our ground truth captures a diverse set of abusive behaviors.

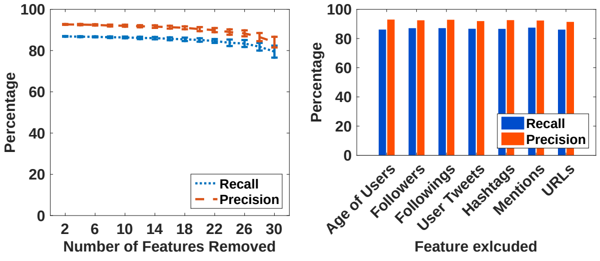
We also acknowledge that abusive applications that post less than 7 tweets will not be detected by our model. However, we argue that these low-activity applications do not pose a significant threat due to their low tweet volume. In other words, while posting fewer tweets allows abusive applications to go undetected, it also limits their ability to conduct abuse on a large scale. Moreover, if needed, we can detect these low-activity abusive applications using our machine learning model trained for smaller values of k with reasonably high precision and recall.

Feature Ablation. We perform an ablation experiment to understand the impact of removing features on the classification accuracy of our model in Figure 7. We randomly remove a varying number of features from 0 to 30 to train and test our model. We repeat this experiment 100 times for each number of removed features. Figure 7(a) shows that precision and recall decreases as we remove more features but this decrease is not substantial. Specifically, the average precision and recall decrease by only 8.6% and 7.4%, respectively, when the number of removed features increase from 2 to 30. We perform another ablation experiment to understand the contribution of individual feature sets of user-based and tweet-based features. Specifically, we remove one feature set at a time to train and test our model. Figure 7(b) shows that the classification accuracy does not



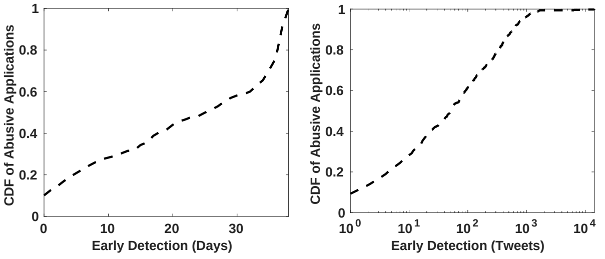
(a) Ten-fold cross validation (b) Early Detection at $k = 7$ (Days) (c) Early Detection at $k = 7$ (Tweets)

Figure 6: We achieve best precision and recall of 94.7% and 89.7%, respectively, at $k = 25$. We select $k = 7$ as a suitable trade-off between early detection and classification accuracy, where we detect 42% of abusive applications at least a month before Twitter during which time these abusive applications post millions of tweets.



(a) Removing features randomly (b) Excluding a feature set

Figure 7: Ablation experiment at $k = 7$ shows that our features are resilient to obfuscation attempts by attackers. More specifically, (a) shows that our feature set is resilient to obfuscation attempts against a combination of features and (b) shows that our classification results are not dependent on individual user-based and tweet-based features.



(a) Early Detection In the Wild (Days) (b) Early Detection In the Wild (Tweets)

Figure 8: The deployment of our machine learning approach in the wild shows that attackers continue to register new applications that go undetected for a long time despite Twitter's new countermeasures while our machine learning approach detects them several weeks before Twitter by analyzing their first seven tweets.

significantly degrade without any individual feature set. The lowest precision of 91.2% and recall of 85.9% is observed when we remove the *URLs* feature set which includes entropy of URLs, unique percentage of URLs, and ratio of URLs to tweets. We conclude that our trained machine learning model is resilient against attempts to obfuscate a specific feature set or a combination of features.

5.2 In the Wild Detection

Next, we show that attackers are able to register new applications and abuse them despite Twitter's most recent countermeasures against them. We also show that our machine learning approach can detect these new abusive applications that are missed by Twitter's countermeasures.

Overview of Twitter's new countermeasures. Recall from Section 3.2 that Twitter has several countermeasures in place to detect and remove abusive applications [5, 10]. However, these countermeasures have not sufficiently deterred attackers from abusing third-party applications. To mitigate the abuse of third-party applications, Twitter recently enforced several new countermeasures [9]. Specifically, Twitter introduced a new policy to review use cases and check policy compliance of new third-party Twitter applications at registration to mitigate abuse. Twitter also introduced new rate limits on the use of POST endpoints (e.g., tweet/retweet, like/follow). Finally, Twitter introduced new ways for users to detect and report abusive applications. Since these countermeasures were implemented after our data collection period that ended in December 2017, we are interested in studying whether they are effective in mitigating abuse of third-party applications. More specifically, we want to find out whether attackers can still register new applications and evade detection and whether our machine learning approach can accurately detect them as early as possible.

In the Wild Deployment. To study the effectiveness of Twitter's newly introduced countermeasures, we use Twitter's streaming API to collect a new tweet dataset during September-October 2018 which is after Twitter's new countermeasures went into effect. To focus on new high-activity applications, we filter the applications that also appeared in our older dataset and those with fewer than seven tweets. We use Twitter's REST API to retrospectively query the deletion status of tweets of 2,225 new applications. Using the approach outlined in Section 3, we find that Twitter removed 532 new applications. In other words, Twitter removed about a quarter (24%) of new applications, which were able to bypass vetting at registration and post hundreds of thousands of tweets before being eventually detected by Twitter.

Next, we use our machine learning model trained on Twitter sample dataset to classify the 2,225 new applications. Our model is able to detect 93% (495 out of 532) of the applications removed by Twitter. It is noteworthy that our model detects these abusive applications that evade detection for weeks while posting hundreds of thousands of tweets. Figure 8 shows that 40% of these abusive applications are detected by our model at least a month before Twitter's new countermeasures. Figure 6(c) shows that 62% of these abusive applications posted at least 100 tweets after detection by our model, which uses only the first-7 tweets. In total, these abusive applications posted 239,485 tweets after detection by our model and before being detected by Twitter's countermeasures. Note that we estimate that abusive applications posted tweets in the order of tens of millions after detection by our model since our tweet collection from Twitter's streaming API is an approximately 1% sample of all tweets. In addition to these 495 abusive applications, our model also classifies 390 new abusive applications that go undetected by Twitter. We manually inspect 10% of these applications sorted in the descending order of our model's detection probabilities. We find that 95% of all the inspected applications are clearly abusive. Among the abusive applications missed by Twitter, we find applications that are part of various spam campaigns such as astroturfing, and profile visit scam [26] (also discussed in Section 4.1).

Takeaway. The deployment of our machine learning approach in the wild shows that attackers continue to register new third-party application and post tens of millions of tweets despite Twitter's new countermeasures. We also show that our machine learning approach accurately and early detect these abusive applications that go undetected by Twitter. We believe that our proposed machine learning approach can complement Twitter's existing efforts for accurate and early detection of abusive third-party applications.

5.3 Limitations & Discussion

Next, we address some limitations of our machine learning approach, discuss its deployment to complement Twitter's existing abuse detection systems, as well as ideas for future extensions.

Evasion and Countermeasures. Like any machine learning based system, our approach is susceptible to evasion if attackers become aware of the details of our machine learning framework. Attackers can attempt to manipulate the features used by our machine learning model to evade detection. However, as demonstrated in Figure 7, our approach is resilient to obfuscation attempts against a particular feature or even different combinations of user-based and tweet-based features. Hence, attackers would need to manipulate multiple feature sets, some of which would likely be cost prohibitive for them. For example, our machine learning model captures the pattern that average account age for abusive applications is less as compared to benign applications. To obfuscate average account age, attackers would need to either discard newer fake/compromised accounts limiting the scale of their operations or purchase "aged" accounts that are reportedly much more expensive than newly created accounts [58]. Even if attackers are able to successfully manipulate multiple features and evade detection, we can periodically retrain our machine learning model using new ground truth to capture the evolving behavior of abusive applications. We can further design new features to better capture the changing behavior of abusive applications since our machine learning framework is readily amenable to the addition of new features as needed. Finally, after becoming aware of our early detection system, attackers can mimic the behavior of benign applications initially and delay abusive activities to evade early detection by our machine learning approach. To address this issue, our machine learning approach can be adapted to continuously monitor an application's tweets in a streaming fashion. Since our work focuses on the early detection of abusive applications, the implementation and evaluation of continuous application monitoring is outside the scope of this paper.

Low-volume Abusive Applications. Our machine learning approach will not be able to detect low-volume abusive applications that post only a few tweets because our machine learning model needs at least seven tweets for detection. First, we surmise that attacker could deliberately post very few tweets to evade detection by our machine learning approach. However, this would significantly reduce the scale of abusive activities, especially given Twitter's revamped application registration process that limits automation [9]. Second, it is also likely that Twitter's existing abuse detection systems [5, 8, 9] are able to detect many abusive applications before they post seven tweets needed by our machine learning approach. In other words, we only observe sophisticated abusive applications via Twitter's streaming API that bypass Twitter's abuse detection systems. Thus, we believe that our machine learning approach nicely complements Twitter's existing countermeasures by early detection of abusive applications that otherwise remain undetected by Twitter.

Handling False Positives While our machine learning approach detects abusive applications with a seemingly non-negligible false positive rate of approximately 6%, we argue that it is sufficiently low to be practical at Twitter's scale. Twitter can employ a review system that allows developers to submit an appeal to review incorrectly flagged applications. Note that Twitter already employs a review process to vet developers [9] which can be extended to review appeals for potentially mislabeled abusive applications. Recall that we observed 2,225 applications through Twitter's streaming API with sufficient activity over the duration of 39 days in September-October 2018. Thus, we argue that a false positive rate of 6% translates into a very manageable 5 false positives per day.

Third-Party Application Abuse on Other Online Social Networks. There have been several high-profile reports of third-party application abuse on popular online social networks including Twitter [10], Facebook [29, 36], and Google+ [46]. Our machine learning approach provides a footprint for other popular online social networks for accurate and early detection of third-party application abuse. Unfortunately, we are unable to investigate abuse of third-party applications on other online social networks due to lack of publicly available data. Nonetheless, online social networks operators can replicate our proposed machine learning framework shown in Figure 2. Most of our features (e.g., age of user accounts, number of followers, number of followings) can be directly translated to other online social networks such as Facebook, Instagram, and Snapchat. Moreover, online social network operators can further design new features based on their proprietary data to better capture behavior of abusive applications on their platform. Our machine learning framework is readily amenable to addition of new features.

6 RELATED WORK

We divide prior work into three categories. First, we discuss prior work on the abuse of third-party applications on Twitter and Facebook. Second, we discuss prior work on detection of fake or compromised accounts on Twitter. Third, we discuss prior work on the measurement and detection of spam and malicious activities on Twitter.

Third-Party Application Abuse on Twitter. Prior work has reported on the exploitation of third-party Twitter applications for nefarious purposes [31, 34, 52, 55, 57]. Chu et al. [31] reported the abuse of third-party applications by bots on Twitter. Stringhini et al. [55] also reported the abuse of third-party applications by Twitter follower markets. Egele et al. [34] and Thomas et al. [57] each independently reported more than 9,000 abusive third-party applications being used to spread spam on Twitter. Prior research has also used third-party application information to aid detection of compromised accounts [34] and spam [52] on Twitter. Twitter tries to block applications used by attackers; who periodically register new applications to avoid the shutdown. While Twitter has recently removed more than 240,000 abusive applications [8], as we showed in this paper, the cat-and-mouse game between Twitter trying to detect abusive applications and attackers continuously creating new applications is still ongoing. To the best of our knowledge, we are the first to attempt to directly detect abusive third-party Twitter applications.

Third-Party Application Abuse on Facebook. Prior work has also reported on the exploitation of third-party Facebook applications for nefarious purposes [36, 50]. Most closely related to our work is that of Rahman et al. [50]. The authors proposed a machine learning approach to detect abusive third-party applications on Facebook. They found a collusion network of 5,307 abusive Facebook applications that promote each other. A key difference between our work and theirs is that we focus on *early* detection of abusive applications but their detection is post hoc. More recently, Farooqi et al. [36] reported that spammers exploit legitimate third-party Facebook applications to provide fake likes and comments. While they employed temporal clustering and IP rate limits to mitigate the abuse of legitimate third-party Facebook applications, we propose a supervised machine learning approach for early detection of abusive third-party Twitter applications.

Fake/Compromised Accounts. There is a large body of prior work on the detection of fake or compromised accounts in online social networks [21, 24, 25, 28, 34, 40, 42, 43, 49, 53, 54, 59, 61, 62]. First, researchers have leveraged account information such as demographics and number of followers/friends to detect fake or compromised accounts [21, 25, 43, 53, 54]. For example, Stringhini et al. [54] trained machine learning models using account features such as number of friends and messages to detect spamming accounts on Facebook and Twitter. Second, researchers have leveraged social connectivity information to detect fake or compromised accounts [28, 61, 62]. For example, Cai and Jermaine [28] used the latent community model to detect Sybil communities that are linked relatively loosely with the rest of the social graph. Third, researchers have leveraged activity patterns to detect fake or compromised accounts [24, 34, 40, 42, 59]. For example, Egele et al. [34] detected compromised accounts by identifying synchronized changes in account behavior within a short time period. It has been shown time and again that more sophisticated attackers can mimic account information, social connectivity, and activity patterns of real accounts to evade detection by such approaches. While our work is complementary to prior research on the detection of fake or compromised accounts, we believe that it may be more effective to directly target the mechanisms used by attackers to orchestrate fake or compromised accounts. Therefore, we focus on detecting abusive third-party Twitter applications that are used by attackers to control fake or compromised accounts.

Spam/Malware Activities. A large body of prior work has focused on the detection and characterization of spam activities on online social networks [37, 38, 49, 52, 55]. Grier et al. [38] characterized different types of spam activities such as phishing, malware, and scam on Twitter. Gao et al. [37] clustered user activity based on URL and textual similarity to detect and characterize spam campaigns on Facebook. Stringhini et al. [55] trained a machine learning model using activity based features such as the rate of change of followers/followings to detect Twitter follower market customers. Song et al. [52] trained a machine learning model using activity based features such as re-tweet time distribution to detect crowdturfing targets (e.g., tweets) on Twitter. Nilizadeh et al. [49] distinguished between benign and spam activities (e.g., tweets) based on their dissemination patterns in communities that share some topics of interest. While prior research has focused on the detection and characterization of malicious activities in online social networks, we aim to detect abusive applications as early as possible to minimize their malicious activities.

7 CONCLUSION

In this paper, we presented a machine learning based approach for accurate and early detection of abusive third-party applications on Twitter. First, we performed a longitudinal measurement study of abusive third-party Twitter applications over a period of 16 months. Our measurements demonstrated an ongoing arms race between attackers registering and abusing new applications and Twitter actively trying to detect and remove them. Second, since abusive applications go undetected for several months, we proposed a machine learning approach for accurate and early detection of abusive applications by analyzing their first few tweets. The evaluation showed that our machine learning approach can accurately detect abusive applications several months before Twitter's existing abuse detection systems, preventing these abusive applications from posting millions of spam and malicious tweets. Third, the deployment of our machine learning model in the wild showed that attackers continue to abuse third-party applications despite new countermeasures enforced by Twitter. Our machine learning approach can complement Twitter's existing abuse detection systems for accurate and early detection of abusive third-party applications.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation under grant numbers 1715152 and 1815131.

REFERENCES

[1] OAuth Core 1.0 Revision A. https://oauth.net/core/1.0a/, June 2009.

[2] One Million Registered Twitter Apps. https://blog.twitter.com/official/en_us/a/2011/one-million-registered-twitter-apps.html, July 2011.

[3] Facebook Apps Leaderboard - AppData. https://web.archive.org/web/20161022132414/http://www.appdata.com/leaderboard/apps, 2016.

[4] Malicious Twitter Applications and Abuse of the Twitter API. https://www.slickrockweb.com/malicious-twitter-applications.php, June 2017.

[5] Our approach to bots and misinformation. https://blog.twitter.com/official/en_us/topics/company/2017/Our-Approach-Bots-Misinformation.html, June 2017.

[6] Twitter is sweeping out fake accounts like never before, putting user growth at risk. https://www.washingtonpost.com/technology/2018/07/06/twitter-is-sweeping-out-fake-accounts-like-never-before-putting-user-growth-risk/, June 2017.

[7] Update: Russian interference in the 2016 US presidential election. https://blog.twitter.com/official/en_us/topics/company/2017/Update-Russian-Interference-in-2016--Election-Bots-and-Misinformation.html, Sept. 2017.

[8] How Twitter is fighting spam and malicious automation. https://blog.twitter.com/official/en_us/topics/company/2018/how-twitter-is-fighting-spam-and-malicious-automation.html, July 2018.

[9] New developer requirements to protect our platform. https://blog.twitter.com/developer/en_us/topics/tools/2018/new-developer-requirements-to-protect-our-platform.html, July 2018.

[10] Update on Twitter's Review of the 2016 U.S. Election. https://blog.twitter.com/official/en_us/topics/company/2018/2016-election-update.html, Jan. 2018.

[11] Application Permission Model - Twitter Developers. https://dev.twitter.com/oauth/overview/application-permission-model, 2019.

[12] Docs - Twitter Developers. https://dev.twitter.com/rest/public, 2019.

[13] "Free followers" apps. https://support.twitter.com/articles/20171936, 2019.

[14] Help with my compromised account. https://help.twitter.com/en/safety-and-security/twitter-account-compromised, 2019.

[15] OAuth with the Twitter API - Twitter Developers. https://dev.twitter.com/oauth, 2019.

[16] Rate limits - Twitter Developers. https://developer.twitter.com/en/docs/basics/rate-limits, 2019.

[17] Sample realtime Tweets - Twitter Developers. https://developer.twitter.com/en/docs/tweets/sample-realtime/overview/GET_statuse_sample, 2019.

[18] Selenium - Web Browser Automation. http://www.seleniumhq.org/, 2019.

[19] Sign in with Twitter. https://dev.twitter.com/web/sign-in, 2019.

[20] The Twitter Rules. https://support.twitter.com/articles/18311, 2019.

[21] A. Aggarwal and P. Kumaraguru. What they do in shadows: Twitter underground follower market. In *13th IEEE Annual Conference on Privacy, Security and Trust (PST)*, 2015.

[22] H. Almuhimedi, S. Wilson, B. Liu, N. Sadeh, and A. Acquisti. Tweets are forever: a large-scale quantitative analysis of deleted tweets. In *ACM CSCW*, 2013.

[23] S. Bennett. 14 percent Use Third Party Apps for Twitter. http://www.adweek.com/digital/twitter-third-party-ads/, Adweek, Aug. 2014.

[24] A. Beutel, W. Xu, Wenkatesan, Chirstopher, and Christos. CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks. In *WWW*, 2013.

[25] Y. Boshmaf, D. Logothetis, G. Siganos, J. Leria, J. Lorenzo, M. Ripeanu, and K. Beznosov. Integro: Leveraging Victim Prediction for Robust Fake Account Detection in OSNs. In *NDSS*, 2015.

[26] C. Boyd. "Who visits your Twitter profile" spam app brings week of chaos. https://blog.malwarebytes.com/cybercrime/2018/01/who-visits-your-twitter-profile-spam-app- brings-week-of-chaos/, Jan. 2018.

[27] R. Brandom. The Google Docs spam attacks played off Google's most fundamental weakness. https://www.theverge.com/2017/5/4/15544608/google-docs-spam-phishing-email-hack, The Verge, May 2017.

[28] Z. Ca and C. Jermaine. The Latent Community Model for Detecting Sybil Attacks in Social Networks. In *NDSS*, 2012.

[29] C. Cadwalladr and E. Graham-Harrison. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election, The Guardian, 2018.

[30] N. Christin. Security Economics: From Game Theory to Field Measurements. https://www.sigmetrics.org/sigmetrics2017/Christin2017_SIGMETRICS_tutorial.pdf SIGMETRICS Tutorial, 2017.

[31] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *Annual Computer Security Applications Conference (ACSAC)*, 2010.

[32] E. D. Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, and M. Z. Shafiq. Paying for Likes?: Understanding Facebook Like Fraud Using Honeypots. In *ACM Internet Measurement Conference (IMC)*, 2014.

[33] E. E. Hammer-Lahav. The OAuth 1.0 Protocol. IETF RFC 5849, April 2010.

[34] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. COMPA: Detecting Compromised Accounts on Social Networks. In *NDSS*, 2013.

[35] S. Farooqi, M. Ikram, E. D. Cristofaro, A. Friedman, G. Jourjon, M. Kaafar, Z. Shafiq, and F. Zaffar. Characterizing Key Stakeholders in an Online Black-Hat Marketplace. In *IEEE/APWG Symposium on Electronic Crime Research (eCrime)*, 2017.

[36] S. Farooqi, F. Zaffar, N. Leontiadis, and Z. Shafiq. Measuring and Mitigating OAuth Access Token Abuse by Collusion Networks . In *ACM IMC*, 2017.

[37] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao. Detecting and Characterizing Social Spam Campaigns. In *ACM IMC*, 2010.

[38] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The Underground on 140 Characters or Less. In *ACM CCS*, 2010.

[39] S. Gupta, A. Khattar, A. Gogia, P. Kumaraguru, and T. Chakraborty. Collective Classification of Spam Campaigners on Twitter: A Hierarchical Meta-Path Based Approach. In *WWW*, 2017.

[40] M. Ikram, L. Onwuzurike, S. Farooqi, E. D. Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, and Z. Shafiq. Measuring, Characterizing, and Detecting Facebook Like Farms. *ACM Transactions on Privacy and Security*, 20(4):1–28, 2017.

[41] N. Jagpal, E. Dingle, J.-P. Gravel, P. Mavrommatis, N. Provos, M. Rajab, and K. Thoma. Trends and Lessons from Three Years Fighting Malicious Extensions. In *USENIX Security*, 2015.

[42] M. Jian, P. Cui, A. Beutel, C. Faloutsos, and S. Yang. CatchSync: catching synchronized behavior in large directed graphs. In *ACM KDD*, 2014.

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

[44] K. Lee, P. Tamilarasan, and J. Caverlee. Crowdturfers, Campaigns, and Social Media: Tracking and Revealing Crowdsourced Manipulation of Social Media. In *ICWSM*, 2013.

[45] I. Lunden. Twitter revoked API access for 142K apps covering 130M 'low-quality' tweets in 1 week under new terms. https://techcrunch.com/2018/04/25/twitter-axed-142k-apps-violating-tos-in-q1-accounting-for-130m-low-quality-tweets/, Apr. 2018.

[46] D. MacMillan and R. McMillan. Google Exposed User Data, Feared Repercussions of Disclosing to Public. https://www.wsj.com/articles/google-exposed-user-data-feared-repercussions-of-disclosing-to-public-1539017194, The Wall Street Journal, 2018.

[47] A. Mak. Beware: A Twitter Scam That Appeals to Users' Vanity Is Hijacking Accounts. http://www.slate.com/blogs/future_tense/2017/10/10/a_fake_app_is_making_twitter_accounts_post_spam.html, Oct. 2017.

[48] F. Morstatter, J. Pfeffer, H. Liu, and K. Carley. Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In *ICWSM*, 2013.

[49] S. Nilizadeh, F. Labreche, A. Sedighian, A. Zand, J. Fernandez, C. Kruegel, G. Stringhini, and G. Vigna. POISED: Spotting Twitter Spam Off the Beaten Paths. In *ACM CCS*, 2017.

[50] M. S. Rahman, T.-K. Huang, H. V. Madhyastha, and M. Faloutsos. FRAppE: Detecting Malicious Facebook Applications. In *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2012.

[51] J. Russell. Prominent Twitter accounts compromised after third-party app Twitter Counter hacked. https://techcrunch.com/2017/03/15/twitter-counter-hacked/, TechCrunch, Mar. 2017.

[52] J. Song, S. Lee, and J. Kim. CrowdTarget: Target-based Detection of Crowdturfing in Online Social Networks. In *ACM CCS*, 2015.

[53] G. Stringhini, G. Jacob, M. Egele, C. Kruegel, and G. Vigna. EVILCOHORT: Detecting Communities of Malicious Accounts on Online Services. In *USENIX Security Symposium*, 2015.

[54] G. Stringhini, C. Kruegel, and G. Vigna. Detecting Spammers on Social Networks. In *Annual Computer Security Applications Conference (ACSAC)*, 2010.

[55] G. Stringhini, G. Wang, M. Egele, C. Kruegel, G. Vigna, H. Zheng, and B. Y. Zhao. Follow the Green: Growth and Dynamics in Twitter Follower Markets. In *ACM Internet Measurement Conference (IMC)*, 2013.

[56] K. Thomas, C. Grier, V. Paxson, and D. Song. Suspended Accounts in Retrospect: An Analysis of Twitter Spam . In *ACM SIGCOMM*, 2011.

[57] K. Thomas, F. Li, C. Grier, and V. Paxson. Consequences of Connectivity: Characterizing Account Hijacking on Twitter. In *ACM CCS*, 2014.

[58] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson. Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse. In *USENIX Security Symposium*, 2013.

[59] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Towards Detecting Anomalous User Behavior in Online Social Networks. In *USENIX Security Symposium*, 2014.

[60] G. Wang, C. Wilson, X. Zhao, Y. Zhu, M. Mohanlal, H. Zheng, and B. Y. Zhao. Serf and Turf: Crowdturfing for Fun and Profit. In *WWW*, 2012.

[61] Z. Yang, C. Wilson, T. G. Xiao Wang, B. Zhao, and Y. Dai. Uncovering Social Network Sybils in the Wild. *ACM Transactions on Knowledge Discovery*, 2014.

[62] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against sybil attacks via social networks. In *ACM SIGCOMM*, 2006.