Understanding Promotion-as-a-Service on GitHub

Kun Du
Tsinghua University
dk15@tsinghua.edu.cn

Hao Yang
Tsinghua University
h-yang@tsinghua.edu.cn

Yubao Zhang
University of Delaware
ybzhang@udel.edu

Haixin Duan*
Tsinghua University
QI-ANXIN Group
duanhx@tsinghua.edu.cn

Haining Wang
Virginia Tech
hnw@vt.edu

Shuang Hao
University of Texas at Dallas
shao@utdallas.edu

Zhou Li
University of California, Irvine
zhou.li@uci.edu

Min Yang
Fudan University
m_yang@fudan.edu.cn

ABSTRACT

As the world's leading software development platform, GitHub has become a social networking site for programmers and recruiters who leverage its social features, such as star and fork, for career and business development. However, in this paper, we found a group of GitHub accounts that conducted promotion services in GitHub, called "promoters", by performing paid star and fork operations on specified repositories. We also uncovered a stealthy way of tampering with historical commits, through which these promoters are able to fake commits retroactively. By exploiting such a promotion service, any GitHub user can pretend to be a skillful developer with high influence.

To understand promotion services in GitHub, we first investigated the underground promotion market of GitHub and identified 1,023 suspected promotion accounts from the market. Then, we developed an SVM (Support Vector Machine) classifier to detect promotion accounts from all active users extracted from GH Archive ranging from 2015 to 2019. In total, we detected 63,872 suspected promotion accounts. We further analyzed these suspected promotion accounts, showing that (1) a hidden functionality in GitHub is abused to boost the reputation of an account by forging historical commits and (2) a group of small businesses exploit GitHub promotion services to promote their products. We estimated that suspicious promoters could have made a profit of $3.41 million and $4.37 million in 2018 and 2019, respectively.

CCS CONCEPTS

• **Security and privacy → Network security**.

KEYWORDS

GitHub, Promoter Detection, Promotion-as-a-Service

ACSAC 2020, December 7–11, 2020, Austin, USA

ACM ISBN 978-1-4503-8858-0/20/12…$15.00
https://doi.org/10.1145/3427228.3427258

ACM Reference Format:
Kun Du, Hao Yang, Yubao Zhang, Haixin Duan*, Haining Wang, Shuang Hao, Zhou Li, and Min Yang. 2020. Understanding Promotion-as-a-Service on GitHub. In *Annual Computer Security Applications Conference (ACSAC 2020), December 7–11, 2020, Austin, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3427228.3427258

1 INTRODUCTION

GitHub was founded in 2008 and has now become the most important code management and sharing website. According to a 2019 GitHub Report [18], there are more than 40 million developers, more than 44 million repositories created, and 2.9 million organizations within GitHub. In addition to being used as a code repository, GitHub also integrates several functionality for online socialization, resembling those of Facebook and Twitter. In fact, developers can watch, star, and fork repositories of others, introducing social communications. By watching a repository, developers can receive notifications for new pull requests and issues that are created. Starring a repository means a developer is interested in this project and would pay sustained attention on it. Forking a repository enables other developers to build their own repositories based on the current one. This means the code in this repository can be reused effectively. Those functions encourage the contribution of high-quality code and lower the obstruction of developing new open-source projects.

GitHub's Impact on Job Recruiting. Due to its prominent role in the software community, the number of stars, watches, and forks attached to a GitHub user or repository has been considered a strong indicator of coding skills, and it is used as a metric when screening job applicants. For example, Devskiller, a developer screening and online interview platform, states that "Stars and forks are a sign of good, usable code" and "good code is forked and starred a lot, so pay attention to these elements" [9]. In Zhaopin [50], the most popular online recruitment service provider in China, many job advertisements related to software development require applicants to have more than a certain number of stars on their owned repositories. The most common requirement is to own a repository with at least 100 GitHub stars.

GitHub Abuse. With such similar requirements, some developers attempt to manipulate the social statistics of their own GitHub

accounts by purchasing stars and forks, which then boosts the underground "Promotion-as-a-Service" business for GitHub that sells stars and forks for profit. In fact, since 2018, there have been some scattered reports about such fraudulent activities [32, 52]. In 2019, even SK Telecom, the biggest mobile service provider in Korea, is reported abusing GitHub stars by giving free drinks to accounts for starring a specified repository [21]. Although this Promotion-as-a-Service on GitHub is another type of fraud and is not allowed in most instances, so far, there is no systematic study on this issue, not to mention a deep understanding of the problem's scale and fraudsters' strategies.

Our Studies. In this paper, we performed the first large-scale measurement and analysis of Promotion-as-a-Service on GitHub. First, we crawled GitHub logs from 2015 to 2019 in GH Archive [14], which is a project recording public user events on GitHub. The log files consist of more than 20 event types, such as commits, forks, watches, tickets, comments, and member changes. These events are aggregated into buckets separated by hours. The total size of the log files is 4.79TB.

Our first task is to identify activities related to this type of fraud. Although the problem is similar to *crowdturfing* attacks, in which human workers are paid to commit fraudulent online activities for a buyer, existing detection systems like those used on social networks [36, 42, 49, 51] cannot be directly applied because the user activities in GitHub are far more complicated than just "post," "like," "follow," and "comment". For GitHub promoters, they have more choices to conceal their promotion tracks by forking, watching, issuing in a popular repository, or even faking updates to their own repositories. As such, we decided to build a new detection system tailored to this problem. To obtain ground-truth datasets, we created a repository in GitHub with only a few script files and then ordered 1,023 stars and forks by taking advantage of GitHub promotion services. Tracing back from these paid stars and forks, we identified a list of promotion accounts. The activity histories of these promotion accounts were also extracted from log files that we crawled from GH Archive.

After a pilot analysis, we trained an SVM (Support Vector Machine) classifier by using the data related to these promotion accounts and reputable GitHub accounts we sampled elaborately from normal GitHub accounts. We applied the SVM classifier on all of the accounts extracted from the log files and detected 63,872 suspected promotion accounts. We checked their homepages in GitHub and found that a large ratio of suspected promotion accounts had not yet been banned by GitHub during our study.

Next, we conducted a comprehensive analysis on these suspected promotion accounts to understand how they operate and gain profit.

We analyzed the organization distribution of these suspected accounts, clustering them into groups to understand their topological structure and relations. Then we examined suspected promotion accounts to check if they were banned by GitHub itself, and found that most of them had not been detected yet. We further analyzed the characteristics of fake stars and forks, profile, and the registration time of these suspected accounts, revealing more intrinsic characteristics of these suspected promotion accounts.

Moreover, we identified different features between normal and suspicious promoted repositories. Regarding the business and operational models of this GitHub fraud, there are two main interesting observations. First, we witnessed that a hidden functionality in GitHub can be abused to boost the reputation of a promotion account by forging the time and frequency of historical commits. Second, we observed that some software companies published parts of their products' source code or instructions on GitHub and paid promotion services to boost these repositories, targeting the GitHub trending list, in order to attract potential customers to purchase their products.

Contributions. We summarize our main contributions below:

(1) We performed the *first* comprehensive study on GitHub promotion and uncovered the strategies used by suspicious promoters. We found that stars and forks of a repository are not trustworthy indicators of a developer's coding skill, due to the use of fraudulent promotion services. Based on the GitHub log data in GH Archive, we estimate that suspicious promoters made a profit of about $3.41 million and $4.37 million in 2018 and 2019, respectively.

(2) We conducted a large-scale measurement on more than 40 million GitHub accounts by examining the data from 2015 to 2019. We developed an SVM classifier and trained it through the publicly available user historical activity data. We evaluated our classifier using an F1-measure, achieving an accuracy of 99.1% on the ground-truth dataset. We identified 63,872 suspected promotion accounts from GitHub accounts.

(3) We shed new light on how this promotion service is operated. We also disclosed a hidden functionality of GitHub that allows a user to pretend to be a skillful developer retroactively. We reported this type of abuse to GitHub, and they indicated that they would pass our request on to the right team for remediation.

The rest of this paper is organized as follows. Section 2 briefs the background of GitHub. Section 3 elaborates on how we processed data and built an SVM classifier to detect promotion accounts. Section 4 presents the large-scale measurement study to uncover the characteristics of suspected promotion accounts. Section 5 demonstrates how promoters help their clients to forge a hard-working account by tampering with historical commits and how small businesses exploit GitHub to promote their products. Section 6 discusses related issues and possible countermeasures. Section 7 surveys related works, and finally, Section 8 concludes our work.

2 BACKGROUND

GitHub provides a web-based hosting service for code hosting and version control by using Git. Therefore, it offers all the functions of Git (*i.e.*, distributed version control and source code management) as well as its own features, including access control and collaboration features [6]. GitHub has different kinds of plans for enterprise and individual users. In general, an individual developer prefers to use a free account to host open-source or private repositories. It was reported that GitHub has more than 40 million developers as of December 2019 [18].

Since GitHub has social-networking functions, *e.g.*, starring and forking, it facilitates social interactions among developers. The number of a repository's stars and forks indicates a developer's skills to some extent. During job screenings, a candidate could

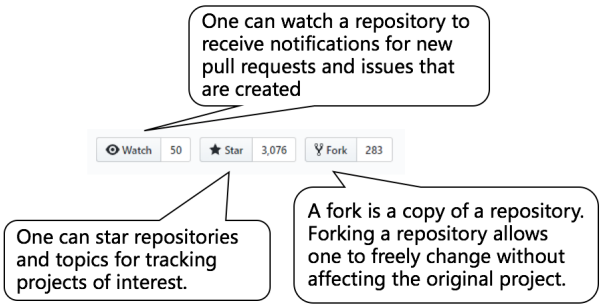


Figure 1: Implications of watch, star, and fork

be evaluated in part based on the number of stars and forks of her repositories. Therefore, promotion services have emerged and been exploited by developers to promote their repositories with paid stars and forks, especially for job screenings. In this section, we will explain how stars and forks work in GitHub, how GitHub promotion services operate and make a profit. At the end, we will also discuss the problem scope of this work.

2.1 How Stars and Forks Work on GitHub

Generally speaking, starring a repository is considered a technical endorsement on the repository. Therefore, the code quality of a repository has a positive correlation with the number of stars. Developers can obtain a considerable number of stars by writing code with a superior quality [13]. Starring a repository also helps a GitHub user keep track of changes. Figure 1 shows the user interface of these features.

Forking a repository is similar to creating a copy, which allows developers to modify code without directly changing the original repository. After forking a repository, developers can either propose changes to the repository that will be reviewed later or create a new project based on it. Therefore, the number of forks can indicate the popularity and re-usability of a repository.

The focus of this work is on stars and forks in a repository, because these two operations are abused by dishonest developers to increase their career prospects in software development [32, 52]. Other GitHub social-networking functions such as "watch" and "follow" are not considered, as to the best of our knowledge, they have not been used as factors for job screenings so far.

2.2 How GitHub Promotion Operates

We observed the GitHub promotion services from search engines, public websites, web blogs, online shops, and instant messaging (IM) tools including Telegram, QQ, and WeChat (the last two are mainly used in China). The underground market in Darknet was also included for this purpose [1]. Here we focus on the entities behind promotions and make the following observations.

First, there are a small number of merchants selling GitHub accounts in the Darknet. One of them claimed that these selling accounts can be successfully logged into and even offered a lifetime warranty. The price is about $2.07 per account, which implies that it is not hard for promotion service providers to set up enough promotion accounts and operate for a long time in GitHub.

Second, in addition to individual sellers, we also discovered a few websites dedicated to GitHub promotion services. One of them is called GitStar [23], which serves as a platform for users to exchange their stars and forks. We inspected the website and found some interesting characteristics, which are described below: (1) This website is not opened on the publicly known port 80 but 88. We speculate that the website owner attempted to avoid public attention. Moreover, we queried the domain name `gitstar.top` record in passive DNS provided by Farsight [12] and found that it had pointed to various IP addresses over time. This shows that the site migrates more often than normal ones. (2) This website enforces web cloaking. We received no response when we visited the website from IP addresses out of China. This shows that the main customers of this website are located in China. (3) During the registration process, the website requires the same username as the one used in GitHub and checks the ownership by asking the registrant to star a famous repository [19] in GitHub. After registration, developers can post the repositories that solicit stars and forks. GitStar acts as a bulletin board that lists all of these repositories. All members of GitStar can star and fork the repositories listed, regardless of the repository's content or quality. The publisher is supposed to return the favor to other repositories when receiving stars or forks. If not, the website will treat it as an "owe." All of the owe information is open and members of the website can determine if a repository is worthy of starring or forking by checking the publisher's owe information. (4) We also observed that GitHub API is leveraged by GitStar to query an account's star and fork information. During the registration process, GitStar can check the ownership of an account by examining if the account indeed performed the star operation on the repository required. During operation step, if user A has finished a star operation on user B's repository, GitStar is able to validate whether user B performs the same on A's repository through GitHub API. This is another abuse of GitHub API in this type of blackhat promotion.

Third, there are also a number of IM groups through which GitHub users can exchange stars and forks for free or profit. We identified more than 20 groups by searching "GitHub star each other" (translated from Chinese) in QQ and WeChat (IM tools like whatsapp). The largest group has more than 1,020 members and charges a "membership" fee. To investigate the market, we paid $1.49 to join the group. The owner of this group can earn more than $1,520 by just collecting the "membership" fees. We also joined three other IM groups for more information and comparison. After monitoring these groups for more than one year, we found that on average, there were about 20 repositories asking for promotion every day. About 20 to 30 members in the chat group actively confirmed that they had given stars or forks for the promoted repositories. We also contacted the users in these groups who operated promotion services and found that it costs $0.40 to purchase one star and $0.50 to purchase one fork.

Fourth, there are a few online shops selling GitHub stars and forks (*e.g.*, the websites illustrated in [15] and [35]). We found that the online shops in different locations preferred different charging modes. In China, operators preferred online third-party payment like Alipay or WeChat Pay, although this type could be tracked and

[1] We examined the most famous market, *Dream Market*, in Darknet.

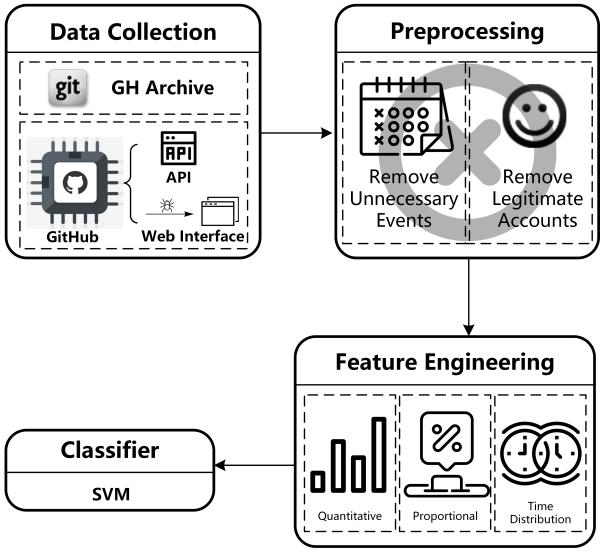


Figure 2: System Architecture.

Figure 3: An example of GH Archive.

linked to a specific person. In other regions, especially in North America or Europe, operators preferred to use Paypal or Bitcoin.

In the end, to further understand the business model of GitHub promotion, we ordered services from a few promotion service providers and performed an *infiltration* study by following a similar methodology of previous works that inspect underground business [10, 24, 28, 30].

2.3 Problem Scope

It is worthwhile to note that GitHub promotion is a worldwide problem rather than a regional one (*e.g.*, China), even though the way that promotion service providers operate in different countries may vary. For example, promotion service providers in China utilize both websites and IM groups to attract customers and accept payment mainly through the payment channels of IM tools (*e.g.*, WeChat). By contrast, in North America, promotion service providers sell GitHub stars and watchers mainly through websites and accept payment only via Paypal or Bitcoin [15, 35]. Some people in the other countries have also discussed this problem on Internet [3, 38].

3 PROMOTION ACCOUNT DETECTION

In this section, we describe the detection methodology used in this paper. Figure 2 shows the architecture of the detection system. We first describe the dataset used to understand GitHub promotion. Then we introduce how to conduct data preprocessing and data analysis. Later, we present how to train the classifier and the detection results of the classifier. Finally, we show how we validated these results.

3.1 Datasets

We collected the GitHub data from the following three sources:

GH Archive [14]. GH Archive records the public GitHub timeline and archives it for further analysis. Each line of an archive file represents a JSON encoded event record reported by the GitHub API, as shown in Figure 3. Each event record includes seven properties, such as "event_id," "event_type," and "actor." In this study, we collected all archive files from 2015 to 2019 and used four properties, including "actor.login_name," "repo.name," "created_at," and "event_type."

GitHub API [1]. GH Archive only archives the history since 2015 and also misses the registration and profile information. So, GitHub API is utilized to collect these missing data, such as the profile of a specific GitHub account. For avoid being abused, the API interface has a rate limit, 60 requests per hour for unauthenticated users.

GitHub Web Interface [22]. Through the web interface, we can crawl the information directly from GitHub web pages, such as a specific GitHub account's avatar, the popular repository list, and the hot trend list. There is no visit limit in most cases.

3.2 Data Preprocessing

We extracted a total of 23, 375, 824 accounts in GH Archive from 2015 to 2019. To build a classification model in an effective and efficient manner, we preprocessed the dataset as follows:

Removing Events With No Accounts. We grouped each account's event logs by event type and found that not every event type contains a valid record. Therefore, we removed all such event types if all accounts extracted have no valid records between 2015 and 2019 (32 event types removed in total) and kept 14 event types. We list all these 14 event types and their meanings in Table 1. We highlight both fork and watch events since they are closely related to the fork and star promotion services in GitHub.

Removing Legitimate Accounts. As illustrated in Section 2.2, promoters make the most of their profit by starring and forking repositories. Therefore, we ruled out inactive accounts that have no star or fork action records between 2015 and 2019, since those accounts were not engaged in the promotion we targeted.

Table 1: Event meanings and Example of User log count in some event type.

Event Type	Explanation	User Examples		
		adieuadi**	anrf**	xzrunn**
ForkEvent	a user forks a repository.	6	3	1
WatchEvent	someone stars a repository	881	97	51
CommitCommentEvent	a commit comment is created.	3	4	0
CreateEvent	create repository, branch, or tag.	42	53	93
DeleteEvent	delete branch or tag.	58	1	0
GollumEvent	a Wiki page is created or updated.	0	0	0
IssueCommentEvent	an issue comment is created, edited, or deleted.	115	646	0
IssuesEvent	an issue is opened, edited, deleted or etc.	63	204	0
MemberEvent	a user accepts or is removed as a collaborator to a repository.	5	2	0
PublicEvent	a private repository is open sourced.	2	0	0
PullRequestEvent	a pull request is assigned, unassigned etc.	523	46	0
PullRequestRevi-ewCommentEvent	a comment on a pull request's unified diff is created, edited and etc.	43	110	0
PushEvent	Triggered on a push to a repository branch.	770	727	1728
ReleaseEvent	a release is published.	5	9	0

To further filter out other irrelevant accounts, we developed a simple heuristic that considers the cost of maintaining a promotion account and the gain of selling forks and stars. We denote the prices for a single star and fork as P_s and P_f, the number of stars and forks that have been given by the promotion account as C_s and C_f, respectively. The value of the account is denoted as V. An account is considered a possible promotion account if it holds $P_s \times C_s + P_f \times C_f > V$. We obtain C_s and C_f of an account by counting the number of its watch and fork events, and set P_s, P_f, and V based on the infiltration study above in Section 2.2. Specifically, we set V to 2.07, P_s to 0.4, and P_f to 0.5, in light of the observations in DreamMarket and IM groups. The price is similar across different promotion service providers. After ruling out these accounts, we obtained more than 14 million possible dedicated accounts for further data analysis.

3.3 Characteristics of GitHub Promotion and Obstacles in Distinguishing

In order to acquire the ground truth, we contacted 10 promotion service providers in the IM groups, and purchased 1,023 star and fork promotions for our test repository. To avoid these providers coming from a single group, we contacted 4 from the QQ group that is the most popular IM tool in China, 3 from WeChat that is another popular IM tool in China in recent years, and the rest 3 from telegram that is used by promoters worldwide. Each single fork or star corresponds to an individual promotion account. Therefore, we collected 1,023 distinct promotion accounts via the test repository. We crawled all of the promotion accounts' information and extracted the events of each promotion account from the archive files. There are three main characteristics and obstacles in distinguishing promotion accounts from normal ones.

(1) The purpose of a promotion account is to generate profit by performing star and fork operations on a customer's repository. The owners of promotion accounts rarely use these accounts for project development. As such, most of the promotion accounts' operations are star and fork. By contrast, the operations on normal accounts are much more diverse, just like creating a repository, pulling and modifying code. However, there exist some normal users whose action pattern is quite similar to that of promotion accounts, since they star and fork a number of repositories likely for future concerns. This similarity can result in increasing the false positive rate in our detection. To overcome this difficulty, we employ more reliable features (*e.g.*, *time distribution*) to address this problem, which will be discussed later.

(2) Many customers who purchase promotion services will ask service providers to finish the star and fork promotions as soon as possible. Therefore, promotion accounts are usually associated with a large number of star and fork operations in a short time period. However, this pattern could also exhibit on normal users as well. A visitor to another account's homepage can easily star all the repositories without taking a long time to search. In fact, we have been informed by some GitHub users that they prefer to star *all* the projects under a reputable developer account (*e.g.*, Facebook) in order to watch any related code changes. This also increases the complexity of differentiating legitimate and promotion accounts. To overcome this challenge, we used the time interval of adjacent operation pair as a feature, which will be illustrated in Section 3.5.

(3) Most promotion accounts perform star and fork operations on well-known repositories in order to disguise as normal accounts for evading detection. Table 2 shows an operation sequence in a short interval from a promotion account. We observed that all operations are "watch events," which means starring a repository, and the interval between two operations is less than 20 seconds. The fifth record is the test repository we used in our study. The promotion account performed star operations on not only our test repository but also other repositories that belong to Facebook, Alibaba, *etc* for evading detection. However, this pattern cannot last for a long time, because the number of popular repositories is limited and one repository can be starred or forked only once.

To distinguish promotion accounts from legitimate accounts, we consider distinct characteristics of promotion accounts and develop reliable features for training classifiers in the following illustration.

Table 2: Operation Sequence of a Promotion Account

No.	Type	Created_at	Repo	Star Count	Repo Type
1	WatchEvent	2018-12-23T07:37:38Z	mahmoud/awesome-python-applications	6,369	Popular Author
2	WatchEvent	2018-12-23T07:37:44Z	FavioVazquez/ds-cheatsheets	3,597	Popular Author
3	WatchEvent	2018-12-23T07:38:02Z	facebookresearch/flashlight	762	Organization
4	WatchEvent	2018-12-23T07:39:19Z	trekhleb/homemade-machine-learning	8,749	Popular Author
5	WatchEvent	2018-12-23T07:39:30Z	ghost1****/t****	716	Promotion Target
6	WatchEvent	2018-12-23T07:39:38Z	FAQGURU/FAQGURU	3,529	Popular Author
7	WatchEvent	2018-12-23T07:39:46Z	alibaba/x-deeplearning	2,145	Organization
8	WatchEvent	2018-12-23T07:40:00Z	alash3al/redix	689	Popular Author

Table 3: Features Used by Our Classifier

Name	Description
NO14	Number of Different Operations (14-Dimension)
NSFE	Number of Star and Fork Operations
RSFA	Ratio of Sum of Star and Fork Operations in All the Operations
AAOPM1	Average of Adjacent Operations Pair in One Minutes
AAOPM2	Average of Adjacent Operations Pair in Two Minutes
AAOPM3	Average of Adjacent Operations Pair in Three Minutes
AAOPH1	Average of Adjacent Operations Pair in One Hour

Table 4: Comparison of Classifier Performance

Classifier	Precision (%)	Recall (%)	F1
Naive Bayes	95.4	64.5	0.77
KNN	96.2	87.3	0.915
Logistic Regression	95.7	98.7	0.972
Decision Tree	97.9	98.7	0.983
Random Forest	98.3	99.5	0.989
SVM	98.5	99.7	**0.991**

3.4 Dataset for Training

By using our honeypot repository, we identified promotion accounts that had made stars and forks to the honeypot repository. Since we never advertised our honeypot projects and the projects had no meaningful code or documents at all, all stars and forks must come from promotion services we infiltrated. These identified promotion accounts serve as positive samples for training the classifier.

For the negative samples, we consider authentic users who made major active contributions in well-known repositories. Specifically, we collected two kinds of accounts: the contributors of popular GitHub repositories and those who have proposed valuable issues on popular repositories. The former accounts have a considerable proportion of commit and push events, while the latter have a number of operations, including push, issue, and comment events. Considering the diversity of negative samples, we selected the top 10 programming languages reported by GitHub [17], and found the most popular repository in https://gitstar-ranking.com/ [2] for each different language. From these repositories, we collected 1,550 users as negative samples, including 200 high-profile users who are quite active all the time and 1,350 normal users.

[2]A site in which one can see top 1,000 users, organizations, and repositories.

3.5 Features

In Section 3.3, we discuss the differences between promotion accounts and normal ones, which can help us to select proper features for developing classifiers. First of all, promotion accounts have much more frequent operations than normal users in terms of 14 different operation types, especially in `WatchEvent` and `ForkEvent`. Therefore, we selected the number of these operations as features. Moreover, promoters basically focus on star and fork. Thus, we computed the total number and ratio of fork and star operations among all the operations. Meanwhile, we also need to capture the burst of promotion accounts (*i.e.*, a number of star and fork operations in a short time period). The burst is associated with the promotion nature. To this end, we calculated the average time interval between successive operations of promotion accounts. Specifically, this average time interval is set to different timing granularity, ranging from one minute, two minutes, three minutes, to one hour. The detailed description of the features is presented in Table 3. The features are normalized by dividing the largest feature value.

3.6 Classifiers

We used the features mentioned in Table 3 for model training. We employed six different popular classification algorithms for promoter detection, including Naive Bayes, K-Nearest Neighbors (KNN), Logistic Regression, Decision Tree, Random Forest, and SVM (Support Vector Machine). Based on the training dataset, we evaluated their classification performance with 10-fold cross-validation, and used the standard binary classification metrics of recall (R), precision (P), and F1-measure (F1) to measure classification accuracy.

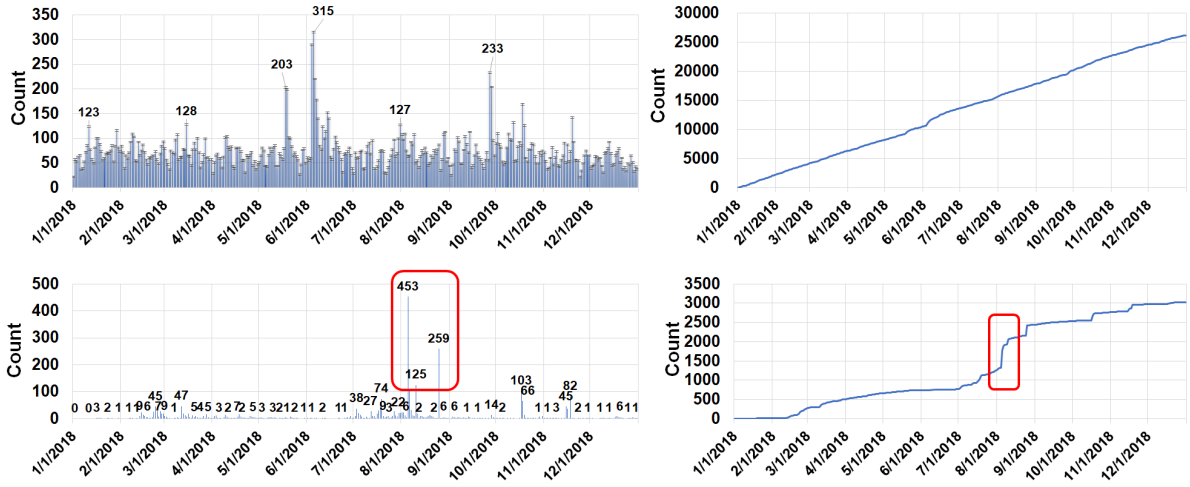


Figure 4: Star distribution over time for a normal repository (top) and a suspiciously promoted repository (bottom).

In this step, "Recall" refers to the ratio of the amount of correctly classified promotion accounts over the total amount of real promotion accounts. "Precision" is the ratio of the number of correctly classified promotion accounts to the total number of classified promotion accounts. "F1-measure" combines precision and recall, and it is defined in equation 1 as the harmonic mean of precision and recall. The F1 score reaches its best value at 1 and its worst at 0. Our classification results are listed in Table 4.

$$F1 = 2 \times (P \times R)/(P + R) \quad (1)$$

Among these six different classification algorithms, SVM achieves the highest classification accuracy, as all three classification metrics of SVM are higher than those of the other classifiers. Finally, we applied the SVM-based classifier on the dataset in Section 3.2, and detected 63,872 suspected promotion accounts.

When we finished modeling the algorithm comparison, we further attempted to determine the effectiveness of different features in Table 3. Therefore, we divided them into three categories according to their characteristics and evaluated them individually. The first category is quantitative, including NO14 and NSFE. The second is proportional category, including RSFA. The third is time-distributed category, including AAOPM1-3 and AAOPH1. When only the first category features are included, the accuracy of the classifier is 73.3%. After adding the second category features, it reaches 83.9%. When the time distribution type is included, it reaches 98.5% (All these results are from the SVM classifier). However, if only the proportional or time-distributed features are included, the accuracy is less than 80%. This shows that all these three types of features are necessary for the classifier.

3.7 Account Validation

Due to the lack of ground-truth data, it is difficult to validate suspected promotion accounts. In contrast to validating malware or spam samples, there is no malicious content associated with suspected promotion accounts. Here we verified whether an account is used for promotion by checking the associated repositories. We notice when a repository rising in popularity receives many stars in a short time, it has a long-tail distribution and shrinks slowly. However, a promoted repository will shrink sharply because promoters tend to complete the promotion tasks as fast as possible for maximizing their profits. If a repository shows a sharp increase in the number of stars or forks, followed by a sharp decrease, we would consider the repository abnormal and the related accounts suspicious.

To demonstrate these distinct patterns between normal repositories and promoted ones, we plot the time distribution of stars between a normal repository and a suspicious promoted repository in Figure 4. The top is the VScode repository published by Microsoft, which obtained a total of 26,211 stars in 2018. The bottom is related to a suspicious promoted repository identified by us. We can see that in the promoted scenario, there are extraordinary star spikes in August 2018 with 453, 259, and 125 stars. On the right side, we can see the star cumulative distributions of these two repositories. The top one linearly increases, while the bottom one has a big jump, followed by an insignificant increase. This indicates that promoted repositories usually experience a drastic increase in the number of stars and forks within a short time period, significantly deviating from a normal increase pattern. After the drastic increase, the owner of the promoted repository stopped to use the promotion service, and then the repository will seldomly receive any stars or forks from other users.

Based on the feature mentioned above, we randomly sampled 1,000 out of those 63,872 suspected promotion accounts and focused on the repositories that had been starred or forked. First, we checked the number variation of the stars or forks of the repositories against time. Then, we counted the increase of stars and forks per day and employed the standard deviation of star and fork increments of the repository per day to determine whether the growth rates of stars and forks are much larger than normal. Specifically, we classified those repositories with standard deviations larger than 25 as suspected repositories. Note that we empirically selected a rather large threshold 25 to lower the false positive rate. Finally, we found

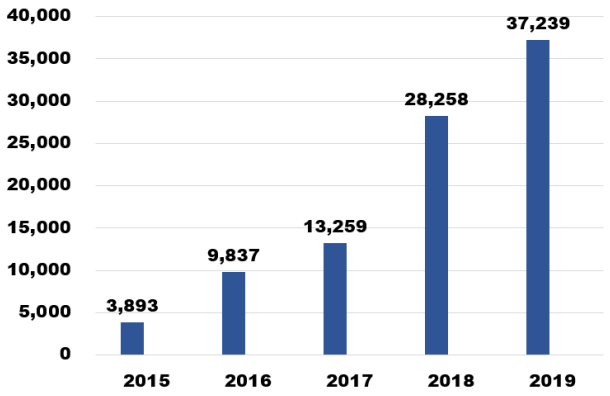


Figure 5: Suspected promotion accounts' time distribution.

that 973 of the sampled 1,000 suspected promotion accounts had starred or forked more than three suspected repositories, indicating the effectiveness of our approach.

3.8 Ethical Issues

To better understand how GitHub promotion service providers work and gain profits, as well as to learn the providers' strategies from the inside, we purchased two GitHub promotion services at the cost of about $295 in total. This may raise an ethical concern that the GitHub promotion service providers were funded by us. However, we argue that our influence is very limited compared to the projected total profit of $3.41 million, as shown in Section 4.8. Another concern is that our test accounts and repositories may have brought negative impact on GitHub. To eliminate such a potential negative impact, we deleted these two accounts and all related repositories at the end of our study.

4 MEASUREMENT AND ANALYSIS

Using our SVM classifier trained in Section 3, we identified 63,872 suspected promotion accounts. In this section, we first investigate the temporal dynamics and organization distributions of these suspected promotion accounts. Second, we cluster them into groups to study the topological structure in every group and the relationship among different groups. Third, we check how many suspected promotion accounts are officially banned by GitHub, observing that only less than 5% of these accounts are detected and banned by the existing defense mechanism in GitHub. Next, we compare average star and fork counts between normal users and suspected promotion accounts, as well as the profile information of both types. Finally, we examine the lifespan of suspected promotion accounts in GitHub, exploring when these accounts are registered and estimating the revenue of this kind of promotion service.

4.1 Temporal Dynamics

The number of suspected promotion accounts against time ranging from 2015 to 2019 is shown in Figure 5, which shows a rapid growth trending. We can see that there were only 3,893 suspected promotion accounts about five years ago. In 2016 and 2018, the number of suspected promotion accounts was more than doubled than the previous year. By 2019, the total number has increased to 37,239.

Table 5: Organization distribution

Organization	Location	Member Count	Suspected Promoter count
foss****	Singapore	1,353	101
EpicG****	United States	2,021	98
b3***	China	820	38
github-****	-	507	32
NV**** GameW****	-	359	26
co***	-	315	15
gats****	-	336	15
no****	-	358	14
phi****	-	107	14
fashio****	-	85	13

"*****" is for anonymization to protect the organization privacy.

4.2 Organization Distribution

We further inspected how many suspected promotion accounts belong to a developer organization. We extracted the organization information from user profiles. We observed that 4,122 (14.59%) of suspected promotion accounts have organization information. The top 10 organizations with the most suspected promotion accounts are listed in Table 5. From the table, we can see that the first organization has more than 100 suspected promotion accounts, which has a total of 1,353 members and is located in Singapore. The second organization is located in United States and has 2,021 members with only two repositories. The last commit from this organization happened in 2017, and there were only 11 commits in total. We speculate that the organization was closed for business.

4.3 Clustering Analysis

Since only 14.59% of suspected promotion accounts claim their organizations, we clustered distinct groups of suspected promotion accounts for further analysis. Note that our measurements are only associated with 28,258 suspected promotion accounts detected in 2018, due to the strict rate limit of GitHub API interface. We first grouped these 28,258 suspected promotion accounts into different groups by the "following" relationship between them. Then we calculated if these groups share any common suspected promotion accounts. The clustering analysis of suspected promotion accounts can provide complementary information about these accounts.

Although suspected promotion accounts may actively follow others to make a profit, it is not easy for them to be followed by normal accounts. After analyzing the communications with promotion service providers being infiltrated, we learned that suspected promotion accounts may follow one another to form a group. This is because these service providers attempt to give others an impression that these accounts are reliable and reputable. By analyzing the clustered groups of promotion accounts, we can have a deep understanding of their topological structure.

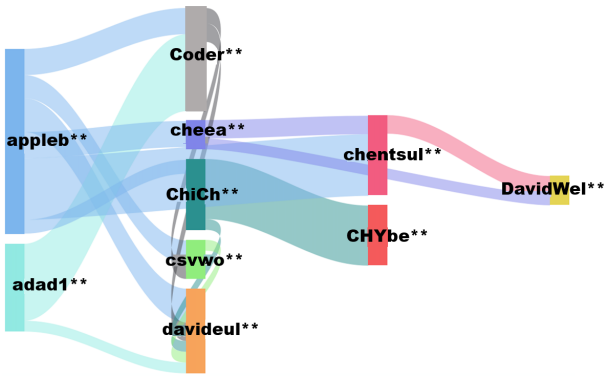


Figure 6: Top 10 Groups of Suspected Promotion Accounts. Every bar represents a group. The line between two groups represents the amount of `shared accounts`. The thicker the line, the more accounts they share.

In detail, we analyzed the "following" relationship among suspected promotion accounts in four steps. (1) For each suspected promotion account a, we crawled all its followers, and conducted the logic "`AND`" operation with those followers that are also suspected promotion accounts, and form a's follower group, F_a, which includes accounts that are both a's followers and suspected promotion accounts. (2) We labeled "a" as a core element and merged "a" into its followers' group F_a, resulting a group node A. It is possible that some accounts can be grouped into more than one group, and we call them "`shared accounts`." (3) For any two groups F_x and F_y, if one of them is a subset of others or their "`shared accounts`" have more than 20% occupancy in their individual follower group "$F_{x/y}$", we merged them together. (4) Finally, we sorted the generated groups by counting group members, while the number of "`shared accounts`" between any two groups determines the number of connections between these two group nodes.

Then, let $G =< V, E >$ denote the undirected graph of suspected promotion account groups. V represents each group of suspected promotion accounts, and E denotes each connection among these groups.

We show our result in Figure 6. From the figure, we can see that these groups are almost fully connected to each other, nearly forming a completely connected graph. Some suspected promotion accounts belong to more than two groups. This implies that some suspected promotion accounts join in multiple promotion groups for making more profit. The largest group shown in the figure contains 400 suspected promotion accounts.

4.4 Banned Accounts

GitHub must have various detection mechanisms for detecting malicious activities. Once detected, GitHub would ban the related accounts and respond with a `404` page when others visit their homepages or repositories. But by receiving only the `404` page, one cannot determine whether the corresponding account does not exist at all or is banned. However, we can distinguish these two scenarios by checking the log file, because the recently banned accounts must have been recorded in the log file.

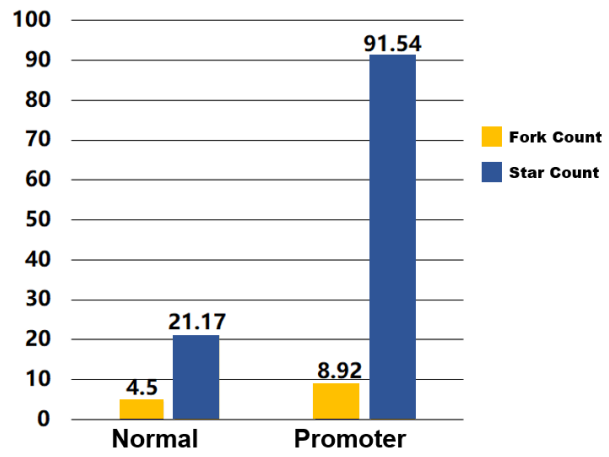


Figure 7: Average number of fork and star operations between normal and promotion accounts.

To check whether a suspected promotion account is banned by GitHub, we visited each of these 28,258 suspected promotion accounts' homepage and observed that only 1,184 of them are actively banned by GitHub. In other words, only 4.19% are banned by GitHub. This implies that only few of these suspected promotion accounts trigger the detection mechanism of GitHub while most of them can evade. Therefore, it is obvious that GitHub needs to significantly improve its detection mechanisms for more effective promotion detection, and our work is able to help GitHub to reach this goal.

4.5 Fake Stars and Forks

Since the profit of GitHub promoters mainly comes from stars and forks, they must perform numerous star and fork operations. Thus, on average, a promotion account is likely to perform more star and fork operations than a normal account. More importantly, since promotion accounts need to finish the promotion target within a short time period, their star and fork operation behaviors are more bursty than those of normal accounts. Therefore, we investigated the average operation number of stars and forks for suspected promotion accounts and compared with that of normal accounts in 2018. For normal accounts, the average operation number of forks is 4.50 and that of stars is 21.17, while for suspected promotion accounts, the average operation number of forks is 8.92 and that of stars is 91.54. Suspected promotion accounts have much more (1.98 times) fork operations and (4.32 times) star operations than normal accounts. Note that the difference in stars between promotion and normal accounts is clearly larger than that in forks. We believe that this is because star is a more straightforward indicator during recruitment, thus it is more widely used and becomes more popular in promotion services. Our observation is also supported by [9] and [50]. The result is shown in Figure 7.

4.6 Profile Comparison

To better distinguish promotion accounts from normal ones, we further inspected and compared the profile information of suspected

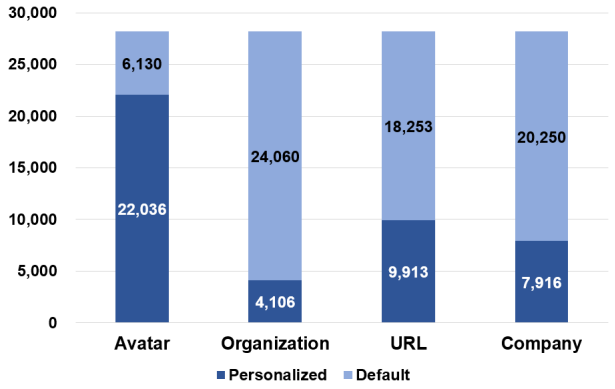


Figure 8: User profile setting of promotion accounts.

promotion accounts with that of normal accounts. In GitHub, for avatar, if an account has not set its personalized avatar, GitHub would set a default one. This default avatar uses only two different colors and its size is 420 x 420. We extracted the avatar URL of each suspected promotion account by visiting a promotion account's homepage via web interface. We detected if an account's avatar is default by checking the avatar's size and colors, using the python library Pillow [33]. Finally, we found 6,130 (21.69%) of suspected promotion accounts using default avatars and 22,036 (78.31%) of them using personalized avatars.

It is impossible to crawl all normal accounts' avatars, and so we randomly sampled 100K accounts from the user list and inspected them. Surprisingly, only 25,698 (25.70%) accounts had personalized avatars while 74,302 (74.30%) used default avatars. This is surprising because normal accounts have a higher likelihood of using default avatars than suspected promotion accounts. This also indicates that suspected promotion accounts are more likely to utilize personalized avatars to disguise as normal.

As for organization information in a profile, we found that 4,106 (14.58%) of suspected promotion accounts have no such information while 24,060 (85.42%) of them do. For the personal homepage link and company information, we found 9,913 (35.19%) and 7,916 (28.10%) of suspected promotion accounts have valid information, respectively. This result shows that suspected promotion accounts do not care much about profile information settings, but most of them use personalized avatars to indicate that they are not bot accounts. Figure 8 shows these results.

4.7 Registration Time

We examined the registration time of these 28,258 suspected promotion accounts in 2018 to check when they were registered. We cannot retrieve an account's registration time precisely from its web interface because there is no such information provided, so we had to query GitHub APIs [1]. We queried all 28,258 suspected promotion accounts through API and extracted their registration time. From the result we find that the registration time ranges from 2008 to 2018, and most of them were registered between February 2013 and July 2016. This means that promotion service providers prefer to use mature accounts rather than new ones, which is in accordance with our common sense that mature accounts are more reputable.

However, these mature accounts are also likely not created by promotion service providers. We speculate that promotion service providers need to leverage different ways to convert these aged accounts into promotion accounts. For example, some of these mature accounts could be compromised. We found that there were at least two security accidents that happened in 2016 and 2018 respectively [47, 48]. Moreover, some of these mature accounts are paid or leased by promotion service providers to become promotion accounts. This has been confirmed by our observation that there are a number of merchants selling GitHub accounts in Darknet to promotion service providers, as presented in Section 2.

4.8 Revenue Estimation

Since the profit of a GitHub promoter is related to the number of star and fork operations being performed, we count the number of star and fork operations of suspected promotion accounts. Note that there are a few uncertain factors in estimating the total revenue of this underground economy. The first factor is that the existing number of star and fork operations in a repository is not accurate, because suspected promotion accounts can star or fork the repository for only a specific time period, then cancel this operation. The second factor is that some promotion accounts perform star and fork operations not only on customers' repositories, but also on popular projects for disguising themselves. We assume that the most authoritative developers and organizations do not need promotion services. So, we first calculated all fork and star operation counts of these suspected promotion accounts. Then we crawled the popular repositories list in https://gitstar-ranking.com/, obtaining 1,000 repositories. Therefore, we can estimate the lower bound of the revenue of the promotion service. According to our observations about price (*i.e.*, $0.50 for each fork operation and $0.40 for each star operation) in Section 2.2, the lower bound of the revenue of all promotion service providers could have been up to $3.41 million in 2018 and $4.37 million in 2019.

5 CASE STUDIES

Here we show case studies of the suspected promotion accounts and the associated repositories that are potentially promoted by these accounts. We first focus on forging retroactive commits, which help suspected promotion accounts to disguise themselves as normal accounts and can be abused by promotion service providers to help customers pretend to be hard-working developers. Then, we describe some small businesses that utilize GitHub promotion as advertisements for selling their products.

5.1 Forging Retroactive Commits

GitHub provides a functionality that allows accounts to modify a committed message in case the owner has submitted private information, and permits accounts to add retroactive commits if he uploads important files (*e.g.*, database IP and password) or if there is no Internet access during the development period.

However, we found that promotion service providers can abuse this functionality and *forge retroactive commits*. As shown in Figure 11 in Appendix A, the left sub-figure is the original commit

history of our honeypot repository. It shows that it had only two contributions in 2016. The right sub-figure is modified by forging historical commits from promotion service providers, and it shows that we had 190 contributions in 2016, which can deceive others into thinking that we are productive developers.

To forge retroactive commits, one must first change historical records stored separately in two local files named "pack-*.idx" and "pack-*.pack." Then, one must forge a temporary file and tamper with its content, such as adding and deleting characters. By uploading these files back to GitHub via API, the promotion service providers can change historical records at will. Figure 12 in Appendix A shows the forging process.

We reported this abuse to GitHub, and they indicated that they would pass our findings on to the corresponding team for serious consideration. Note that though this feature has been discovered before for creating pixel art on a personal homepage in GitHub [16], we are the first to report how this can be abused for promotion.

5.2 Promotion by Small Businesses

In our research, we found that some small businesses, especially IT companies, exploit GitHub to promote their businesses. In such a situation, a company can publish the repository on GitHub and place a link to their homepage on GitHub. By exploiting GitHub's web ranking, their links are also promoted, similar to "blackhat SEO" illustrated in [10, 46]. Here we give an example of GitHub "blackhat SEO" utilized, in which we inspect the strategy for this company to promote their business via GitHub. As shown in Figure 9, the company first publishes the repository on GitHub and places the link in their source code, homepage hyperlink or "readme.md" to their online shop's homepage, showing a hyperlink from GitHub. In most cases, the source code they published on GitHub is a free trial version that provides only limited services, specifically, supporting only few users with restricted functionalities.

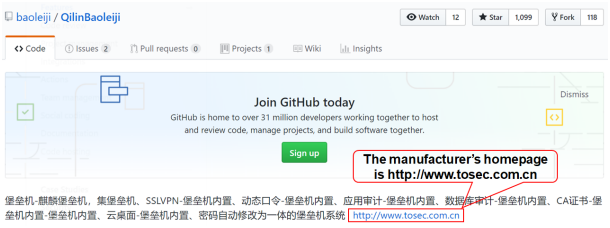


Figure 9: Promotion of small business

5.3 Potentially Compromised Accounts

It has been reported that there are more than 2.1 million organizations in GitHub [18]. However, we found 4,122 (14.59% of 28,258 in 2018) suspected promotion accounts belong to at least one organization. We present an example in Figure 10, where one account from IBM was identified as a suspected promotion account since it forked our honeypot repository during the test period. Moreover, we observed that this account forked more than 100 repositories from December 30th, 2018 to January 20th, 2019. Therefore, we speculate that this account has been compromised.

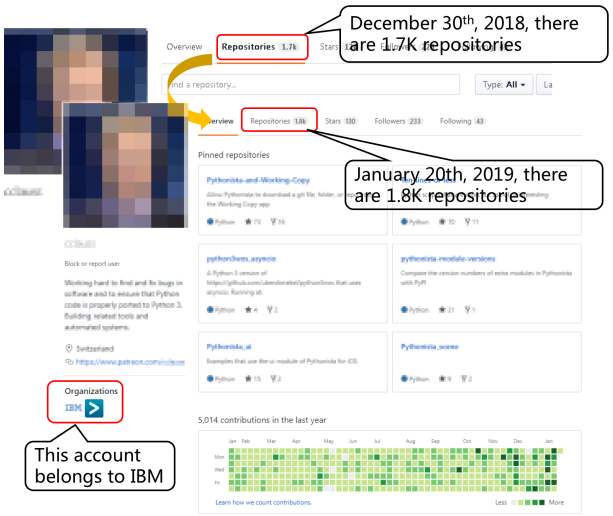


Figure 10: A suspected promotion account from IBM

6 DISCUSSION

Responsible Disclosure and Feedback. We have reported our findings through the "Report abuse" page of GitHub [20] and they promised to pass our request on to the appropriate team for consideration.

Limitations. We develop our detection method by analyzing behavior patterns and then used this method to perform a large-scale detection. However, if adversaries knew our detection algorithms in advance, they would have adjusted their behavior patterns for evasion. For example, they can conduct more common activities such as commit, push, and pull, just like normal users do. However, these actions would increase the cost for maintaining these promotion accounts. An alternative to avoid being detected is to reduce the frequency of starring and forking, but this would also increase the cost of promotion service providers and decrease their profit prominently. Another evasion method may fork or star other popular repositories during their promotion service. This can confuse our detection due to the addition of more actions that cannot be identified in an accurate manner. For our future work, we will investigate possible ways for evasion and then improve our detection by making it more adaptive.

Recommendations to GitHub. We suggest that GitHub should take action to regulate the existing promotion. One possible action is to send emails with confirmed information to suspected accounts and ask them for an active response. The inquiries can depend on the actual activities that have happened on the GitHub repositories, and the responses should be relevant to the inquires. According to our observations, most of these suspected promotion accounts are acquired through selling and buying. Thus, they are likely to ignore emails received in their GitHub accounts or auto-reply to the inquiries. In such a case, GitHub can put special flags on these accounts. Industry recruiters will notice the accounts with special flags, and could potentially reject the applications whose repositories have stars or forks from those special accounts. Thus, the negative impacts of promotion services can be alleviated. Another

action could be requiring suspected promotion accounts to complete a more complicated captcha. This can prevent automatic login and automatic action like forking and starring.

7 RELATED WORKS

Collaboration on GitHub. GitHub is a cloud-based service that helps developers collaborate on a variety of projects. The collaboration plays a vital role in GitHub. Researchers have done a set of works on this collaboration platform [6, 27, 29, 31, 40]. Developers make contributions [8, 34, 41] to improve their reputation. While developers and their community can benefit from the legitimate use of bots [43], the misuse of bots in a promotion service could harm the community but has not yet been noticed by the community.

Manipulating Reputation. Reputation manipulation has become a serious security problem in recent years. In online e-commerce markets, dishonest sellers have been reported to manipulate the reputation system by faking their transaction history. Xu *et al.* [45] investigated five underground markets for reputation manipulation in Taobao, referred to as Seller-Reputation-Escalation (SRE) markets. Within the SRE underground market, sellers can easily harness human labors to conduct fake transactions for improving their stores' reputation. In [4], Cai *et al.* employed reinforcement learning methods to detect the reputation manipulation in online e-commerce markets. Xie *et al.* [44] examined the underground market where mobile app developers can illegally misuse positive reviews and hence boost their reputation. In [5], Chen *et al.* exploited the unusual ranking change patterns of apps to identify promoted apps and detect the collusive promotion groups that engaged in reputation manipulation.

Promotion Services in Online Social Networks. Stringhini *et al.* [37] inspected the Twitter follower markets, which provide promotion services for helping users create a large number of followers in Twitter. Similarly, the underground market for boosting page likes has emerged in Facebook and attracted considerable attentions [2, 7, 25]. Zheng *et al.* [51] demonstrated the collusive promoters who generate seemingly-trustworthy reviews in Dianping, a user-review social network. Jiang *et al.* [26] proposed a graph-mining approach based on catching synchronized behaviors in a large network. A number of studies have also been done to detect crowdturfing [36, 42] and suspicious accounts [11, 39]. Song *et al.* [36] detected crowdturfing through the detection of target objects, such as post content, pages and URLs. Wang *et al.* [42] performed the detection in the context of malicious crowdsourcing systems, where sites connect both paying users and promoters. However, these detection approaches are ineffective in GitHub. GitHub promotion service providers do not need to post content for promotion. Many of them use IM tools to gain profits, and there is no valid site yet. Besides, GitHub promotion service providers take two main actions to conceal their characteristics: (1) They forge retroactive commits in their own accounts actively. Consequently, the account itself can be treated as a normal account that has more complicated action patterns include forking, starring, and issuing. (2) They mix their promotion actions with normal actions. For example, during our infiltration described in Section 3.3, we observed that promotion accounts performed star and fork operations on well-known repositories as a disguise.

8 CONCLUSION

In this paper, we have conducted the *first* comprehensive investigation on a new promotion service on GitHub called "Promotion-as-a-Service," which helps developers increase the number of stars and forks on a repository, so as to improve social status and earn advantages in career development. In this work, we have developed a behavior pattern model by purchasing services from actual GitHub promotion service providers and performed a large-scale scan on all those accounts with star and fork operations from 2015 to 2019. We have detected that 63,872 accounts are promotion accounts that star and fork for profit. Moreover, we have conducted a large-scale measurement on these suspected promotion accounts and the repositories that they starred or forked. We believe that our findings will help the security community to pay more attention to all kinds of fraudulent promotion methods. More importantly, our work will help to achieve fair and objective recruitment in the IT industry.

ACKNOWLEDGMENTS

We thank our shepherd Ting-Fang Yen and anonymous reviewers for their insightful feedback, which helped us improve the quality of this paper. This work was supported in part by the National Natural Science Foundation of China (U1836213 and U1636204) and the BNRist Network and Software Security Research Program (Grant No. BNR2019TD01004).

REFERENCES

[1] api.github.com. 2019. GitHub API Interface. https://api.github.com/.
[2] Prudhvi Ratna Badri Satya, Kyumin Lee, Dongwon Lee, Thanh Tran, and Jason Zhang. 2016. Uncovering Fake Likers in Online Social Networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.*
[3] brunch.co.kr. 2019. SKT-Github-Abuse. https://brunch.co.kr/@supims/595.
[4] Qingpeng Cai, Aris Filos-Ratsikas, Pingzhong Tang, and Yiwei Zhang. 2018. Reinforcement Mechanism Design for Fraudulent Behaviour in E-commerce. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence.*
[5] Hao Chen, Daojing He, Sencun Zhu, and Jingshun Yang. 2017. Toward Detecting Collusive Ranking Manipulation Attackers in Mobile App Markets. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security.*
[6] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work.*
[7] Emiliano De Cristofaro, Arik Friedman, Guillaume Jourjon, Mohamed Ali Kaafar, and M Zubair Shafiq. 2014. Paying for Likes?: Understanding Facebook Like Fraud Using Honeypots. In *Proceedings of the ACM Internet Measurement Conference.*
[8] Giuseppe Destefanis, Marco Ortu, David Bowes, Michele Marchesi, and Roberto Tonelli. 2018. On Measuring Affects of Github Issues' Commenters. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering.* ACM.
[9] devskiller.com. 2019. Devskiller. https://devskiller.com/.
[10] Kun Du, Hao Yang, Zhou Li, Hai-Xin Duan, and Kehuan Zhang. 2016. The Ever-Changing Labyrinth: A Large-Scale Analysis of Wildcard DNS Powered Blackhat SEO. In *USENIX Security Symposium.*
[11] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. COMPA: Detecting Compromised Accounts on Social Networks. In *Proceedings of the Symposium on Network and Distributed System Security.*
[12] farsightsecurity.com. 2019. Passive DNS historical internet database: Farsight DNSDB. https://www.farsightsecurity.com/solutions/dnsdb/.
[13] freecodecamp.org. 2016. How I Got 1,000 Stars on My GitHub Project, and the Lessons Learned Along the Way. https://medium.freecodecamp.org/how-i-got-1000–on-my-github-project-654d3d394ca6.
[14] gharchive.org. 2019. GH Archive. https://www.gharchive.org/.
[15] gimhub.com. 2019. GimHub - Buy GitHub Stars and Followers. https://gimhub.com/.
[16] github.com. 2018. Abusing Github commit history for the lulz. https://github.com/gelstudios/gitfiti.

[17] github.com. 2018. Projects | The State of the Octoverse. https://octoverse.github.com/projects.

[18] github.com. 2018. The State of the Octoverse | The State of the Octoverse celebrates a year of building across teams, time zones, and millions of merged pull requests. https://octoverse.github.com/.

[19] github.com. 2019. GitHub - torvalds/linux: Linux kernel source tree. https://github.com/torvalds/linux.

[20] github.com. 2019. GitHub Report Abuse. https://github.com/contact/report-abuse/.

[21] github.com. 2019. Stop abuse GitHub Star metatron-app/metatron-discovery · GitHub. https://github.com/metatron-app/metatron-discovery/issues/2405.

[22] github.com. 2019. The world leading software development platform GitHub. https://github.com/.

[23] gitstar.org. 2018. GitStar. http://218.241.135.34:88/.

[24] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, M. Zubair Rafique, Moheeb Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. 2012. Manufacturing Compromise: The Emergence of Exploit-as-a-service. In *Proceedings of the ACM Conference on Computer and Communications Security*.

[25] Muhammad Ikram, Lucky Onwuzurike, Shehroze Farooqi, Emiliano De Cristofaro, Arik Friedman, Guillaume Jourjon, Mohammed Ali Kaafar, and M Zubair Shafiq. 2017. Measuring, Characterizing, and Detecting Facebook Like Farms. *ACM Transactions on Privacy and Security* (2017).

[26] Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. 2016. Catching Synchronized Behaviors in Large Networks: A Graph Mining Approach. *ACM Transactions on Knowledge Discovery from Data* (2016).

[27] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M German, and Daniela Damian. 2014. The Promises and Perils of Mining GitHub. In *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM.

[28] Chris Kanich, Nicholas Weaver, Damon McCoy, Tristan Halvorson, Christian Kreibich, Kirill Levchenko, Vern Paxson, Geoffrey M Voelker, and Stefan Savage. 2011. Show Me the Money: Characterizing Spam-advertised Revenue. In *USENIX Security Symposium*.

[29] Paul M Leonardi. 2014. Social media, Knowledge sharing, and Innovation: Toward a Theory of Communication Visibility. *Information Systems Research* (2014).

[30] Kirill Levchenko, Andreas Pitsillidis, Neha Chachra, Brandon Enright, Márk Félegyházi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, Damon McCoy, Nicholas Weaver, Vern Paxson, Geoffrey M. Voelker, and Stefan Savage. 2011. Click Trajectories: End-to-End Analysis of the Spam Value Chain. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[31] Jennifer Marlow, Laura Dabbish, and Jim Herbsleb. 2013. Impression Formation in Online Peer Production: Activity Traces and Personal Profiles in Github. In *Proceedings of the Conference on Computer Supported Cooperative Work*. ACM.

[32] oschina.net. 2018. Github's fake industry chain is exposed. You can buy Stars when you spend money (Translated from Chinese). https://www.oschina.net/news/99612/fake-star-on-github?from=20180909.

[33] pillow.readthedocs.io. 2018. Pillow. https://pillow.readthedocs.io/en/stable/.

[34] Jinglei Ren, Hezheng Yin, Qingda Hu, Armando Fox, and Wojciech Koszek. 2018. Towards Quantifying the Development Value of Code Contributions. In *Proceedings of the 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

[35] smexpt.com. 2018. SMEXPT | Github Stars. https://www.smexpt.com/shop/github-stars/.

[36] Jonghyuk Song, Sangho Lee, and Jong Kim. 2015. Crowdtarget: Target-Based Detection of Crowdturfing in Online Social Networks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.

[37] Gianluca Stringhini, Manuel Egele, Christopher Kruegel, and Giovanni Vigna. 2012. Poultry Markets: On the Underground Economy of Twitter Followers. *ACM SIGCOMM Computer Communication Review* (2012).

[38] theregister.co.uk. 2019. Drinks-for-stars. https://www.theregister.co.uk/2019/07/30/would_you_star_a_github_project_for_a_free_drink/.

[39] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. 2011. Suspended Accounts in Retrospect: An analysis of Twitter Spam. In *Proceedings of the ACM Internet Measurement Conference*.

[40] Ferdian Thung, Tegawende F Bissyande, David Lo, and Lingxiao Jiang. 2013. Network Structure of Social Coding in GitHub. In *Proceedings of the 17th European Conference on Software Maintenance and Reengineering*. IEEE.

[41] Jason Tsay, Laura Dabbish, and James Herbsleb. 2014. Influence of Social and Technical Factors for Evaluating Contribution in GitHub. In *Proceedings of the 36th International Conference on Software Engineering*. ACM.

[42] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y Zhao. 2014. Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers. In *USENIX Security Symposium*.

[43] Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S. Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A. Gerosa. 2018. The Power of Bots: Characterizing and Understanding Bots in OSS Projects. *Proc. ACM Hum.-Comput. Interact.* (2018).

[44] Zhen Xie and Sencun Zhu. 2015. AppWatcher: Unveiling the Underground Market of Trading Mobile App Reviews. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*.

[45] Haitao Xu, Daiping Liu, Haining Wang, and Angelos Stavrou. 2015. E-commerce Reputation Manipulation: The Emergence of Reputation-Escalation-As-A-Service. In *International Conference on World Wide Web*. ACM.

[46] Hao Yang, Xiulin Ma, Kun Du, Zhou Li, Haixin Duan, Xiaodong Su, Guang Liu, Zhifeng Geng, and Jianping Wu. 2017. How to Learn Klingon without a Dictionary: Detection and Measurement of Black Keywords used by the Underground Economy. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[47] zdnet.com. 2016. GitHub warns. https://www.zdnet.com/article/github-warns-some-accounts- compromised-after-reused-password-attack/.

[48] zdnet.com. 2018. GitHub says bug exposed some plaintext passwords | ZDNet. https://www.zdnet.com/article/github-says-bug-exposed-account-passwords/.

[49] Xianchao Zhang, Shaoping Zhu, and Wenxin Liang. 2012. Detecting Spam and Promoting Campaigns in the Twitter Social Network. In *Proceedings of the IEEE 12th International Conference on Data Mining*.

[50] zhaopin.com. 2019. Zhaopin. https://www.zhaopin.com/.

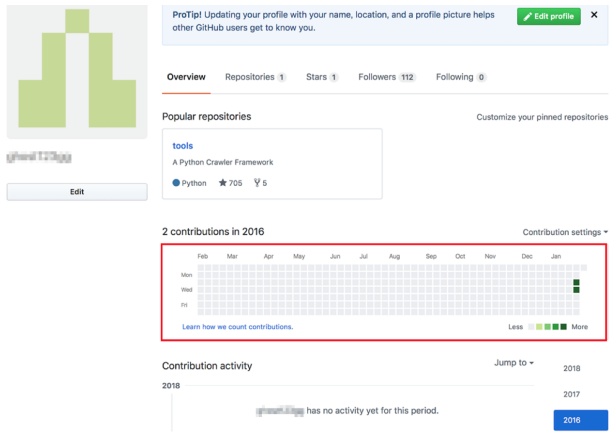
[51] Haizhong Zheng, Minhui Xue, Hao Lu, Shuang Hao, Haojin Zhu, Xiaohui Liang, and Keith Ross. 2017. Smoke Screener or Straight Shooter: Detecting Elite Sybil Attacks in User-Review Social Networks. *arXiv preprint arXiv:1709.06916* (2017).

[52] zhihu.com. 2018. China's mainland GitHub fraud has grown exponentially, behind it... (Translated from Chinese). https://zhuanlan.zhihu.com/p/38791657.

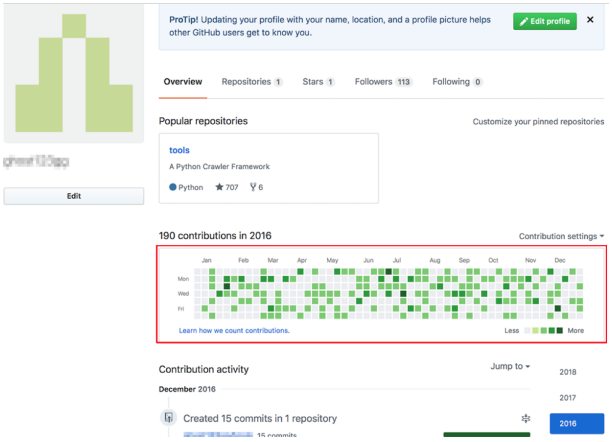
APPENDIX

A Forging Retroactive Commits

Figure 11 shows the abuse of forging retroactive commits. Adversaries can pretend to be an active developer through this abuse.



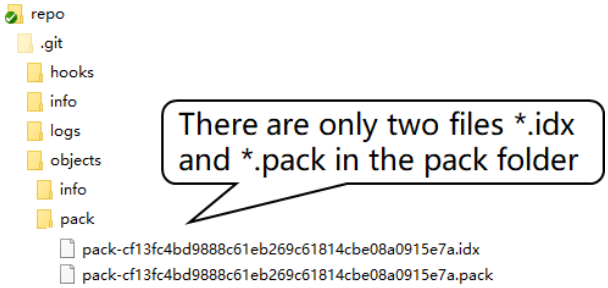
(a) Before the promotion, the honeypot repository had only two contributions in 2016.



(b) After the promotion, the honeypot repository had 190 contributions in 2016.

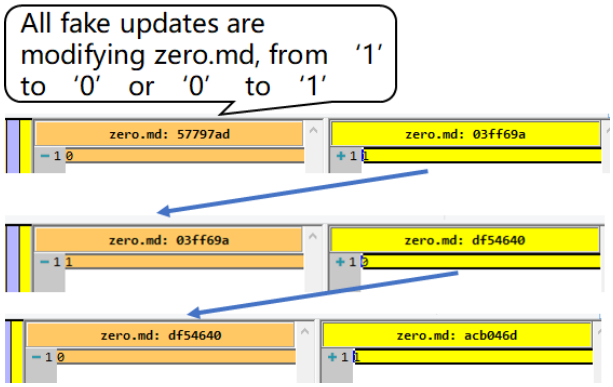
Figure 11: Forging historical commits

Figure 12 shows how adversaries forge retroactive commits and pretend to be a skillful and hardworking developer. In this process, they must first change historical records stored separately in two local files "pack-*.idx" and "pack-*.pack." Then, they forge a temporary file and tamper to add, modify, or delete characters. By uploading these files back to GitHub, adversaries can change historical records at will.



(a) There are two important files in the pack folder.

(b) Adversaries forge retroactive commits all at 17:2 0.



(c) The content of retroactive commits is simple.

Figure 12: The process of forging historical commits.