Detecting Malicious Applications in Facebook

Mitali R. Gurav¹, Akanksha A. Patil², Nagma Y. Dawdani³, Shivraj P. Bendugade³, Ragvendar O. Singh⁵
¹,²,³,⁴Department of Information Technology, INDIA
⁵Department of Computer Engineering & Information Technology, INDIA

ABSTRACT
Facebook’s Rigorous Application Evaluator—arguably the first tool centered on detective work malicious apps on Facebook. To develop FRAppE, we have a tendency to use info gathered by observant the posting behavior of 111K Facebook apps seen across a pair of 2 million users on Facebook. First, we have a tendency to establish a group of options that facilitate North American country distinguish malicious apps from benign ones. as an example, we have a tendency to find that malicious apps usually share names with alternative apps, and that they usually request fewer permissions than benign apps. Second, investing these characteristic options, we have a tendency to show that FRAppE can detect malicious apps with 99.5% accuracy, with no false positives and an occasional false negative rate (4.1%). Finally, we have a tendency to explore the scheme of malicious Facebook apps and establish mechanisms that these apps use to propagate. curiously, we have a tendency to find that a lot of apps interact and support every other; in our dataset, we find 1,584 apps facilitative the infectious agent propagation of three, 723 alternative apps through their posts. Long-term, we have a tendency to see FRAppE as a step towards making associate freelance watchdog for app assessment and ranking, thus on warn Facebook users before putting in apps.

Keywords---- Facebook Apps, Malicious Apps, Plying Apps, on-line Social Networks.

I. INTRODUCTION
Online social networks (OSN) enable and encourage third party applications (apps) to enhance the user experience on these platforms. Such enhancements include interesting or entertaining ways of communicating among online friends, and diverse activities such as playing games or listening to songs. For example, Facebook provides developers an API [10] that facilitate app integration into the Facebook user-experience. There are 500K apps available on Facebook [25], and on average, 20M apps are installed every day [1]. Furthermore, many apps have acquired and maintain a large user base. For instance, Farmville and City Ville apps have 26.5M and 42.8M users to date. Recently, hackers have started taking advantage of the popularity of this third-party apps platform and deploying malicious applications [17, 21, and 24]. Malicious apps can provide a lucrative business for hackers, given the popularity of OSNs, with Facebook leading the way with 900M active users [12]. There are many ways that hackers can benefit from a malicious app: (a) the App can reach large numbers of users and their friends to spread spam, (b) the app can obtain users ’personal information such email address, home town, and gender, and (c) the app can “re-produce” by making other malicious apps popular. To make matters worse, the deployment of malicious apps is simplified by ready-to-use toolkits starting at $25 [13]. In other words, there is motive and opportunity, and as a result, there are many malicious apps spreading on Facebook every day [20]. Despite the above worrisome trends, today, a user has very limited information at the time of installing an app on Facebook. In other words, the problem is: given an app’s identity number (the unique identifier assigned to the app by Facebook), can we detect if the app is malicious? Currently, there is no commercial service, publicly-available information, or research-based tool to advise a user about the risks of an app. Malicious apps are widespread and they easily spread, as an infected user jeopardizes the safety of all its friends. So far, the research community has paid little attention to OSN apps specifically. Most research related to spam and malware on
Facebook has focused on detecting malicious posts and social spam campaigns. A recent work studies how app permissions and community ratings correlate to privacy risks of Facebook apps. Finally, there are some community-based feedback driven efforts to rank applications, such as Whattapp; though these could be very powerful in the future, so far they have received little adoption. Over the past few months Facebook Security has been focused on helping our users stay free from malware. Malware, short for malicious software, are programs installed on your device to disrupt normal operations, collect personal information or gain access to a system. For the past couple years, we have developed systems to proactively identify spam and other malicious content posted on Facebook by user devices infected with malware. After we identify a possible infection, we will notify the user and provide, free of charge, an anti-virus product capable of cleaning the user's device. Now, we are pleased to offer these tools directly to the people who use our service. Facebook already uses internal scanners to detect spam and malicious messages that might have been sent from user accounts hijacked by malware. When found, such accounts are temporarily locked down and their owners are asked to go through a multi-step account recovery process that involves downloading and running a malware scanner called McAfee Scan and Repair. The new "malware checkpoints" feature will allow users who believe their computers might be infected with malware to initiate the account lockdown procedure themselves and perform an antivirus scan for free. Users will be able to choose to scan their computers with McAfee Scan and Repair, a run-once anti-malware scanner, or with Microsoft Security Essentials, a full-featured antivirus product that must be downloaded and installed.

II. METHODOLOGY

Module 1
Detecting malicious apps

Having analyzed the differentiating characteristics of malicious and benign apps, we next use these features to develop efficient classification techniques to identify malicious Facebook applications. We present two variants of our malicious app classifier FRAppE Lite and FRAppE. A. FRAppE Lite is a lightweight version that makes use of only the application features available on demand. Given a specific app ID, FRAppE Lite crawls the on-demand features for that application and evaluates the application based on these features in real time. We envision that FRAppE Lite can be incorporated, for example, into a browser extension that can evaluate any Facebook application at the time when a user is considering installing it to her profile. All of these features can be collected on demand at the time of classification and do not require prior knowledge about the app being evaluated. We use the Support Vector Machine (SVM classifier for classifying malicious apps. SVM is widely used for binary classification in security and other disciplines. We use the D-Complete dataset for training and testing the classifier. We use 5-fold cross validation on the D-Complete dataset for training and testing FRAppE Lite’s classifier. In 5-fold cross validation, the dataset is randomly divided into five segments, and we test on each segment independently using the other four segments for training. We use accuracy, false positive (FP) rate, and true positive (TP) rate as the three metrics to measure the classifier’s performance. Accuracy is defined as the ratio of correctly identified apps (i.e., a benign/malicious app is appropriately identified as benign/malicious) to the total number of apps. False positive rate is the fraction of benign apps incorrectly classified as malicious, and true positive rate is the fraction of benign and malicious apps correctly classified (i.e., as benign and malicious, respectively).

Module 2
Identifying New Malicious Apps

We next train Frappe’s classifier on the entire D-Sample dataset (for which we have all the features and the ground truth classification) and use this classifier to identify new malicious apps. To do so, we apply FRAppE to all the apps in our-Total dataset that are not in the D-Sample dataset; for these apps, we lack information as to whether they are malicious or benign. Of the 98 609 apps that we test in this experiment, 8144 apps were flagged as malicious by FRAppE. Validation: Since we lack ground truth information for these apps flagged as malicious, we apply a host of complementary techniques to validate FRAppE’s classification. We next describe these validation techniques; we were able to validate 98.5% of the apps flagged by FRAppE.

Deleted From Facebook Graph: Facebook itself monitors its platform for malicious activities, and it disables and deletes from the Facebook graph malicious apps that it identifies. If the Facebook API (https://graph.facebook.com/appID) returns false for a particular app ID, this indicates that the app no longer exists on Facebook; we consider this to be indicative of blacklisting by Facebook. This technique validates 81% of the malicious apps identified by FRAppE. Note that Facebook’s measures for detecting malicious apps are however not sufficient; of the 1464 malicious apps identified by FRAppE (that were validated by other techniques below) but are still active on Facebook, 35% have been active on Facebook since over 4 months with 10% dating back to over 8 months.

App Name Similarity: If an application’s name exactly matches that of multiple malicious apps in the D-Sample dataset, that app too is likely to be part of the same campaign and therefore malicious. On the other hand, we found several malicious apps using version numbers in their name (e.g., “Profile Watchers v4.32,” “How long have you spent logged in? v8”). Therefore, in addition, if
an app name contains a version number at the end and the rest of its name is identical to multiple known malicious apps that similarly use version numbers, this too is indicative of the app likely being malicious.

Posted Link Similarity: If a URL posted by an app matches the URL posted by a previously known malicious app, then these apps are likely part of the same spam campaign, thus validating the former as malicious.

Typo squatting of Popular App: If an app’s name is “typo squatting” that of a popular app, we consider it malicious. For example, we found five apps named “Farmville,” which are seeking to leverage the popularity of “Farmville.” Note that we used “typo squatting” criteria only to validate apps that were already classified as malicious by Frappe. We did not use this feature as standalone criteria for classifying malicious apps in general. Moreover, it could only validate 0.5% of apps in our experiment as shown in Table VIII. Manual Verification: For the remaining 232 apps unverified by the above techniques, we cluster them based on name similarity among themselves and verify one app from each cluster with cluster size greater than 4. For example, we find 83 apps named “Past Life.” This enabled us to validate an additional 147 apps marked as malicious by FRAppE.

Promotion Graph Characteristics From the app promotion dataset we collected above, we construct a graph that has an undirected edge between any two apps that promote each other via direct or indirect promotion, i.e., an edge between and if the former promotes the latter. We refer to this graph as the “Promotion graph.”

1) Different Roles in Promotion Graph: Apps act in different roles for promotion.
2) Connectivity: Promotion graph forms large and densely connected groups.
3) Longest Chain in Promotion: App-nets often exhibit long chains of promotion.
4) Participating App Names in Promotion Graph: Apps with the same name often are part of the same app-net.

Module 4
App Collaboration

Next, we attempt to identify the major hacker groups involved in malicious app collusion. For this, we consider different variants of the “Campaign graph” as follows.

• Posted URL campaign: Two apps are part of a campaign if they post a common URL.
• Hosted domain campaign: Two apps are part of a campaign if they redirect to the same domain once they are installed by a user. We exclude apps that redirect to apps.facebook.com.
• Promoted URL campaign: Two apps are part of a campaign if they are promoted by the same redirection URL. It is important to note that, in all versions of the Campaign graph, the nodes in the same campaign form a clique. Finally, we construct the “Collaboration graph” by considering the union of the “Promotion graph” and all variants of the “Campaign graph.” We find that the Collaboration graph has 41 connected components, with
the GCC containing 56% of nodes in the graph. This potentially indicates that 56% of malicious apps in our corpus are controlled by a single malicious hacker group. The largest five component sizes are 3617, 781, 645, 296, and 247.

Module 5
Hosting Domains

We investigate the hosting domains that enables redirection Web sites. First, we find that most of the links in the posts are shortened URLs, and 80% of them use the bitly shortening service. We consider all the bit.ly URLs among our dataset of indirection links (84 out of 103) and resolve them to the full URL. We find that one-third of these URLs are hosted on amazons.com.

III. CONCLUSION

Thus we have detected external malicious from entering to user’s account. Applications present a convenient means for hackers to spread malicious content on Facebook. However, little is understood about the characteristics of malicious apps and how they operate. In this work, using a large corpus of malicious Facebook apps observed over a nine month period, we showed that malicious apps differ significantly from benign apps with respect to several features. For example, malicious apps are much more likely to share names with other apps, and they typically request fewer permissions than benign apps. Leveraging our observations, we developed this project, an accurate classifier for detecting malicious Facebook applications. Most interestingly, we highlighted the emergence of App Nets—large groups of tightly connected applications that promote each other. We will continue to dig deeper into this ecosystem of malicious apps on Facebook, and we hope that Facebook will benefit from our recommendations for reducing the menace of hackers on their platform.

REFERENCES