ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Facebook Applications Novel Approach for Identifying Malicious

Ch.Sahithya [1], M. Kiran Kumar [2], M. RamyaSree [3], G.Madhurya [4], K.Swapna[5]

U.G. Student, Department of CSE, Sree Venkateswara College of Engineering, Nellore, Andhra Pradesh, India[1,3,4,5]

Associate Professor, Department of CSE, Sree Venkateswara College of Engineering, Nellore, Andhra Pradesh, India[2]

ABSTRACT*:* Facebook applications are very attractive now a days with 20 million installs a calendar day, third-party apps are a main reason for the reputation and addictiveness of Facebook. Unfortunately, hackers have take in the potential of using apps for spreading malware and spam. The problem is before now considerable, as we find that at least 13% of apps in datasets are malicious. So the research community has focused on detecting malicious posts and campaigns. In a Facebook application, can we determine if it is malicious? Our explanation involvement is in developing FRAppE Facebook Rigorous Application Evaluator possibly the first tool focused on detecting malicious apps on Facebook. To extend FRAppE, we use information grouped by scrutinizing the posting behaviour of 112K Facebook apps seen transversely 2.3 million users on Facebook. First, we recognize a set of types that help us distinguish malicious apps from benign ones. We find that malicious apps often distribute names with other apps, and they normally request less permission than benign apps. Second, leveraging these distinguishing features, we show that FRAppE can identify malicious apps with 99.6% accuracy, with no false positives and a high true positive rate (95.8%). Finally, we walk around the ecosystem of malicious Facebook apps and recognize mechanisms that these apps use to propagate.

KEYWORDS*:* Facebook apps, malicious, online social networks, spam**.**

I. INTRODUCTION

Online social networks (OSNs) facilitate and encourage third-party applications (apps) to improve the user experience on these platforms. Such improvements comprise interesting or entertaining ways of communicating along with online friends and miscellaneous activities such as playing games or listening to songs. Facebook offers developers an API that facilitates app addition into the Facebook user experience. There are 500K apps accessible on Facebook and on average, 20M apps are installed each day and many apps have purchased and preserve a really huge user base. For instance, Farmville and Cityville apps have 26.5M and 42.8M users to date.

In recent times, hackers have started taking benefit of the fame of this third-party apps platform and installing malicious applications. Malicious apps can make available a profitable business for hackers, given popularity of OSNs, with Facebook leading the way with 900M dynamic users. Hackers can benefit from a malicious app many habits

1) The app can achieve large information of users and their friends to spread spam;
2) The app can obtain users personal information such as e-mail address, gender, hometown
3) The app can “reproduce” by making other malicious apps popular.

In other words, there is motive and opportunity there are a lot of malicious apps spreading on Facebook daily, Nowadays a user has very limited information at the time of installing an app on Facebook. The problem is given an app’s identity number. Can we detect if the app is malicious? Currently, there is no business-related service, openly available information, or research-based tool to give advice a user regarding the risks of an app.The research community has remunerated small concentration to OSN apps specially. Most research related to spam and malware on Facebook has focused on identify malicious posts and social spam movements. At the same time, in a apparently backwards step, Facebook has take apart its app rating functionality recently. An up to date work studies how app permissions and community ratings correlate to privacy risks of Facebook apps FRAppE is a suite of proficient classification techniques for identifying whether an app is malicious or not. To construct FRAppE, we use data from MyPage- Keeper, a protection app in Facebook that supervises the Facebook profiles of 2.2 million users. We analyze

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

112K apps that made 92 million posts over 10 months. This is questionably the first wide ranging study focusing on malicious Facebook apps that focuses on measuring profiling and understanding malicious apps and produces this information into an efficient detection approach.

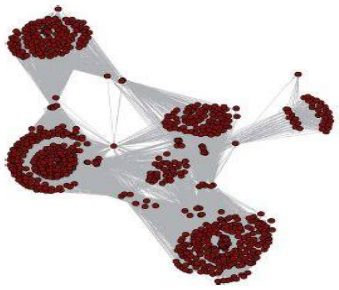

Figure 1: The emergence of AppNets on Facebook. Real snapshot of 770 highly collaborating apps

II.RELATED WORK

A. FACEBOOK APPS

Facebook enables third-party developers to offer services to its users by resources of Facebook applications, Typical desktop and Smartphone applications, installation of a Facebook application by a user does not engage the user downloading and carry out an application binary. As an alternative, when a user adds a Facebook application to her profile, the user grants the application server:
1) permission to access a subset of the information listed on the user's Facebook profile (e.g., the user's e-mail address)
2) permission to perform definite actions on behalf of the user (e.g., the ability to post on the user's wall).
Facebook grants these permissions to any application by handing an OAuth 2.0 token to the application server for every user who installs the application after that; the application can access the data and carry out the explicitly permitted actions on behalf of the user.

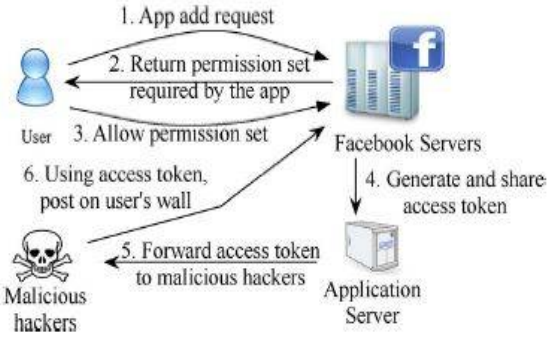

Fig.2 depicts the steps involved in the installation and operation of a Facebook application

Operation of Malicious Applications*:* Malicious Facebook applications normally operate as follows.
• Step 1: Hackers induce users to install the app, usually with a few fake promises (e.g., free iPads).
• Step 2: Just the once a user installs the app, it redirects the user to a Web page where the consumer is requested to perform tasks, such as completing a survey, another time with the lure of fake rewards.
• Step 3: The app thereafter accesses personal information (e.g., birth date) from the user's profile, which the hackers can potentially use to earnings.
• Step 4: The app makes malicious posts on behalf of the user to attract the user's friends to install the identical app
This way the cycle carries on with the app or colluding apps reaching additional users. Personal information or surveys can be selling to third parties to eventually profit the hackers.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

B. OUR DATASETS

The basis of our study is a dataset obtained from 2.3M Facebook users, who are supervised by MyPageKeeper, our security application for Facebook. MyPageKeeper evaluates every URL that it observes on any user's wall or news feed to determine if that URL points to social spam. MyPageKeeper classifies a URL as public spam if it points to a Web page that:

1) Spreads malware;
2) Attempts to "phish" for personal information;
3) Requests the user to carry out tasks that profit the owner of the Web site;
4) Promises false rewards; or
5) Attempts to entice the user to artificially inflate the reputation of the page

Dataset Name	# of apps	
	benign	malicious
D-Total	111,167	
D-Sample	6,273	6,273
D-summary	6,067	2,528
D-Inst	2,257	491
D-Profile Feed	6,063	3,227
D-Complete	2,255	487

Table.1: Summary of the dataset collected by My Page Keeper from June 2011 to March 2012

App ID	App Name	Post Count
235597333185870	What Does Your Name Mean?	1006
159474410806928	Free Phone Calls	793
233344430035859	The Apps	564
296128667112382	WhosStalking?	434
142293182524011	Farmville	210

Table 2: Top malicious apps in D-Sample dataset

III. PREVALENCE OF MALICIOUS APPS

In 60% of malicious apps get a minimum of 100 thousand clicks on the URLs they post. We have a tendency to quantify the reach of malicious apps by decisive a bound on the quantity of clicks on the links enclosed in malicious posts. for every malicious app in our D-Sample dataset, we have a tendency to establish all bit.ly URLs in posts created by that application. Across the posts created by the 6273 malicious apps within the D-Sample dataset, we have a tendency to found that 3805 of those apps had posted5700 bit.ly URLs in total. We have a tendency to queried bit.ly for the clicking count of every computer address. The distribution across malicious apps of the whole variety of clicks received by bit.ly links that they'd announce. We have a tendency to see that hr of malicious apps were able to accumulate over 100K clicks every, with 2 hundredth receiving quite 1M clicks every. The applying with the very best variety of bit.ly clicks during this experiment the "What is that the sexiest factor regarding you?" apperceived one 742 359 clicks. Though it'd be attention-grabbing to search out the bit.ly click-through rate per user and per post, we have a tendency to don't have knowledge for the quantity of users UN agency saw these links. we will question bit.ly's API just for the quantity of clicks received by a link.

IV. PROFILING APPLICATIONS

Given the significant collision that malicious apps have on Facebook, we next seek to develop a tool that can identify malicious applications. Toward developing an understanding of how to build such a tool, in this section, we compare malicious and benign apps with respect to various features.

A. ON-DEMAND FEATURES

The on-demand features associated with an application refer to the features that one can obtain on demand given the application's ID. Such metrics include app name, description, category, company, and required permission set.

1) Application Summary: Malicious apps typically have incomplete application summaries. First, we compare malicious and kindly apps with respect to attributes present in the application's summary app description, company

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

name, and category. Description and company are free-text attributes, either of which can be at most 140 characters. On the other hand, category can be selected from a predefined (by Facebook) list such as "Games," "News," etc., that matches the app functionality best. Application developers can also specify the company name at the time of app creation. For example, the "Mafia Wars" app is configured with description as "Mafia Wars: Leave a legacy behind," company as "Zynga," and category as "Games."

2) Required Permission Set: 97% of malicious apps require only permission from users. Every Facebook app requires authorization by a user before the user can use it. At the time of installation, every app requests the user to grant it a set of permissions that it requires. These permissions are chosen from a pool of 64 permissions predefined by Facebook. Example permissions include access to information in the user's profile (e.g., gender, e-mail, birthday, and friend list), and permission to post on the user's wall.

3) Redirect URI: Malicious apps redirect users to domains with poor reputation. In an application's installation URL, the "redirect URI" parameter refers to the URL where the user is redirected to once she installs the app. We extracted the redirect URI parameter from the installation URL for apps in the D-Inst dataset and queried the trust reputation scores for these URIs from WOT shows the corresponding score for both benign and malicious apps. WOT assigns a score between 0 and 100 for every URI, and we assign a score of 1 to the domains for which the WOT score is not available. We see that 80% of malicious apps point to domains for which WIT does not have any reputation score, and a further 8% of malicious apps have a score less than 5. In contrast, we find that 80% of benign apps have redirect URIs pointing to the apps.facebook.com domain and therefore have higher WOT scores. We speculate that malicious apps redirect users to Web pages hosted outside of Facebook so that the same spam/malicious content, e.g., survey scams, can also be propagated by other means such as e-mail and Twitter spam.

Domains	Hosting # of malicious apps
thenamemeans3.com	34
fasfreeupdates.com	53
wikiworldmedia.com	82
technicalyard.com	96
thenamemeans2.com	138

Table3: Top five domains hosting malicious apps in D-Inst dataset

4) Client ID in App Installation URL: 78% of malicious apps trick users into putting in alternative apps by employing a totally different consumer ID in their app installation computer address. For a Facebook application with ID, once any user visits this computer address, Facebook queries the appliance server registered for app A to fetch many parameters, like the set of permissions needed by the app. Facebook then redirects the user to a computer address that encodes these parameters within the computer address. One amongst the parameters during this computer address is that the "client ID" parameter. If the user accepts to put in the appliance, the ID of the appliance that she's going to find you putting in is that the price of the consumer ID parameter. Ideally, as delineate within the Facebook app developer tutorial, this consumer ID ought to be the image of the app ID, whose installation computer address the user originally visited. However, in our D-Inst dataset, we discover that seventy eight of malicious apps use a consumer ID that differs from the ID of the initial app, whereas only one of benign apps does thus. A doable reason for this can be to extend the survivability of apps.

5) Posts in App Profile: 97% of malicious apps do not have posts in their profiles. An application's profile page presents a forum for users to communicate with the app's developers (e.g., to post comments or questions about the app), or vice versa (e.g., for the app's developers to post updates about the application). Typically, an app's profile page thus accumulates posts over time. We examine the number of such posts on the profile pages of applications in our dataset.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

B. AGGREGATION-BASED FEATURES

Next, we analyze applications with respect to aggregation based features. Unlike the features we considered so far, aggregation based features for an app cannot be obtained on demand. Instead, we envision that aggregation-based features are gathered by entities that monitor the posting behaviour of several applications across users and across time. Entities that can do so include Facebook security applications installed by a large population of users, such as MyPageKeeper, or Facebook itself. Here, we consider two aggregation-based features: similarity of app names, and the URLs posted by an application over time. We compare these features across malicious and benign apps.

1) App Name: *87% of malicious apps have an app name identical to that of at least one other malicious app.* An application's name is configured by the app's developer at the time of the app's creation on Facebook. Since the app ID is the unique identifier for every application on Facebook, Facebook does not impose any restrictions on app names. Therefore, although Facebook does warn app developers not to violate the trademark or other rights of third parties during app configuration, it is possible to create multiple apps with the same app name.

We examine the similarity of names across applications. To measure the similarity between two app names, we compute the Damerau–Levenshtein edit distance between the two names and normalize this distance with the maximum of the lengths of the two names. We then apply different thresholds on the similarity scores to cluster apps in the D-Sample dataset based on their name; we perform this clustering separately among malicious and benign apps.

We see that malicious apps tend to cluster to a significantly larger extent than benign apps. For example, even when only clustering apps with identical names the number of clusters for malicious apps is less than one fifth that of the number of malicious apps, i.e., on average, five malicious apps have the same name. Fig. 12 shows that close to 10% of clusters based on identical names have over 10 malicious apps in each cluster. For example, 627 different malicious apps have the same name "The App." On the contrary, even with a similarity threshold of 0.7, the number of clusters for benign apps is only 20% lesser than the number of apps. Moreover, while most of the clustering of app names for malicious apps occurs even with a similarity threshold of 1, there is some reduction in the number of clusters with lower thresholds. This is due to hackers attempting to "typo-squat" on the names of popular benign applications. For example, the malicious application "FarmVile" attempts to take advantage of the popular "FarmVille" app name, whereas the "Fortune Cookie" malicious application exactly copies the popular "Fortune Cookie" app name. However, we find that a large majority of malicious apps in our D-Sample dataset show very little similarity with the 100 most popular benign apps in our dataset. Our data therefore seems to indicate that hackers creating several apps with the same name to conduct a campaign is more common than malicious apps typo-squatting on the names of popular apps.

2) External Link to Post Ratio: *Malicious apps often postlinks pointing to domains outside Facebook, whereas benign apps rarely do so.* Any post on Facebook can optionally include an URL. Here, we analyze the URLs included in posts made by malicious and benign apps. For every app in our D-Sample dataset, we aggregate the posts seen by MyPageKeeper over our 9-month data-gathering period and the URLs seen across these posts. We consider every URL pointing to a domain outside of facebook.com as an external link. We then define an "external-link-to-post ratio" measure for every app as the ratio of the number of external links posted by the app to the total number of posts made by it.

V. DETECTING MALICIOUS APPS

Having analysed 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

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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

that can evaluate any Facebook application at the time when a user is considering installing it to her profile lists the features used as input to FRAppE Lite and the source of each feature. 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.

B. *FRAppE*

Next consider FRAppE—a malicious app detector that utilizes our aggregation-based features in addition to the on-demand features. Table VII shows the two features that FRAppE uses in addition to those used in FRAppE Lite. Since the aggregation based features for an app require a cross-user and cross-app view over time, in contrast to FRAppE Lite, we envision that FRAppE can be used by Facebook or by third-party security applications that protect a large population of users.

C. IDENTIFYING NEW MALICIOUS APPS

FRAppE's classifier on the entire D-Sample dataset (for which we have all the character 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 D-Total dataset that are not in the D-Sample dataset; for these apps, we lack information as to whether they are malicious or benign.

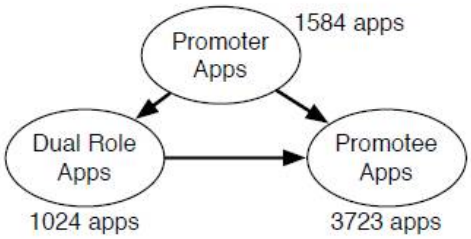

Fig 1: Relationship between collaborating applications

Deleted From Facebook Graph: Facebook itself monitors its platform for malicious activities disables and deletes from the Facebook graph malicious apps that it identified, 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.

VI. MALICIOUS APPS ECOSYSTEM

Analysis shows that malicious apps are out of control on Facebook and indicates that they do not operate in isolation. Indeed, we discover that malicious apps collude at scale lots of malicious apps share the same name, several of them redirect to the same domain upon installation, etc. These observed behaviours indicate well organized crime, with a few prolific hacker groups controlling many malicious apps.

A common way in which malicious apps scheme is by having one app place links to the installation page of another malicious app.

A. BACKGROUND ON APP CROSS PROMOTION

Cross promotion among apps, which is forbidden as per Facebook platform policy, occurs in two different ways. The promoting app can post a link that points directly to another app, or it can post a link those points to a redirection URL, which points dynamically to any one of a set of apps.

Posting Direct Links to Other Apps: We found evidence that malicious apps often promote each other by making posts that redirect users to the promoter's app page; here, when app1 posts a link pointing to app2 , we refer to app1 as the promoter and app2 as the promoter. Promoter apps make such posts on the walls of users who have been tricked into installing these apps. These posts then appear in the news feed of the victim's friends. The post contains an appropriate message to lure users to install the promoted app, thereby facilitate the promoter to accumulate more

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

victims. To study such cross promotion, we crawled the URLs posted by all malicious apps in our dataset and identified those where the landing URL corresponds to an app installation page.

B. 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.* The "Promotion graph" contains 6331 malicious apps that engage in collaborative promotion. Among them, 25% are *promoters*, 58.8% are *promotees*, and the remaining 16.2% play both roles.

2) Connectivity: *Promotion graph forms large and densely connected groups*. We identified 44 connected components among the 6331 malicious apps. The top five connected components have large sizes: 3484, 770, 589, 296, and 247. Upon further analysis of these components, we find the following.

• ***High connectivity***: 70% of the apps collude with more than 10 other apps. The maximum number of collusions that an app is involved in is 417.

• ***High local density*****:** 25% of the apps have a local clustering coefficient2 larger than 0.74.

We use the term app-net to refer to each connected component in the Promotion graph. As an example of an app-net, Fig. 17 shows the local neighborhood of the "Death Predictor" app, which has 26 neighbors and has a local clustering coefficient of 0.87. Interestingly, 22 of the node's neighbors share the same name.

3) Degree Distribution: To understand the relationship between promoter and promotee apps, we create a directed graph, where each node is an app, and an edge from to indicates that promotes.

We can see that 20% of apps have in-degree or out-degree more than 50, which shows that 20% of the apps have been promoted by at least 50 other apps and another 20% of the apps have each promoted 50 other apps.

4) Longest Chain in Promotion: App-nets often exhibit long chains of promotion. We are interested in finding the longest path of promotion in this directed graph. However, finding a simple path of maximum length in a directed cyclic graph is an NP-complete problem. Therefore, we approximate this with a depth-first-based search that terminates after a threshold runtime.We see that the longest path of promotion is 193, and 40% of promoter-only apps have a longest path at least 20. In such paths, at most 17 distinct app names were used, and 40% of the longest paths use at least four different app names. For example, "Top Viewers v5" promotes "Secret Lookers v6," which in turn promotes "Top Lookers v6." "Top Lookers v6" then promotes "who Are They? v4," which in turn promotes "Secret Lurkers v5.73, " and so on.

C. 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 indirection URL. It is important to note that, in all versions of the Campaign graph, the nodes in the same campaign form a clique.

D. HOSTING DOMAINS

We investigate the hosting domains that enable redirection Web sites. First, we find that most of the links in the posts are shortened URLs, and 80% of them use the bit.ly shortening service.

E. APP PIGGYBACKING

From our dataset, we also discover that hackers have found ways to make malicious posts appear as if they had been posted by popular apps. To do so, they exploit weaknesses in Facebook's API. We call this phenomenon *app*

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

piggybacking. One of the ways in which hackers achieve this is by luring users to "Share" a malicious post to get.

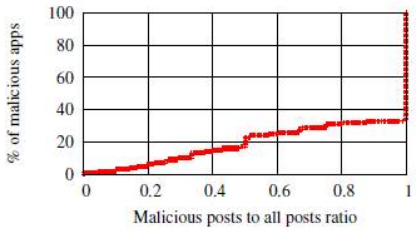

Fig.2: Distribution across apps of the fraction of an app's posts that are malicious.

F. CROSS PROMOTION AS A SIGN OF MALICIOUS INTENTIONS

Thus far, we studied cross promotion among malicious apps based on posts marked as malicious by My Page Keeper. However, My Page Keeper may have failed to flag the posts of many malicious apps. Therefore, here we study the prevalence of cross promotion simply by observing whether the post made by an app includes a URL that points to another app. This enables us to discover a new set of malicious apps that we have failed to identify so far.

1) Data Collection: Cross promotion via posted apps. facebook. com URL: The simplest way to identify whether a post made by an app is pointing to an app's page is to examine if the URL in the post points to the apps.facebook.com domain. We collect 41M URLs monitored by My Page Keeper that point to the apps.facebook.com/namespace domain. The posts containing these URLs were posted by 13 698 distinct apps. We then identify the app ID corresponding to each *namespace*, and thus identify cross-promotion relationships between promoter and promote apps. After ignoring self-promotion where one app promotes itself, we identify 7700 cross-promoting relations involving 4782 distinct apps

Cross promotion via posted shortened URLs: The above method however does not suffice for identifying all instances of cross promotion since many apps post shortened URLs. To investigate app promotion via shortened URLs, we collect 5.8M shortened links monitored by My Page Keeper, out of which 65 448 URLs resolve to the apps.facebook.com domain.

2) Analyzing Cross-Promoting Apps: To identify malicious apps from the 5077 apps, we compare them to our corpus of 14K malicious apps identified by FRAppE in Section V. We consider both apps in a promoter–promotee relationship malicious if either of them appear in our malicious app corpus. This enables us to identify an additional 2052 malicious apps. However, the rest of the 3025 apps are not connected to FRAppE-detected malicious apps.

VII. CONCLUSION

In survey, we explored different research attempts towards exploring the Facebook network, analyzing malicious content on it, and analyzing events on online social media. In universal applications present opportune means for hackers to spread malicious content on Facebook. On the other hand, slight is understood about the characteristics of malicious apps and how they operate. By using a large corpus of malicious Facebook apps observed over a nine-month period, we explained that malicious apps differ significantly from benign apps with respect to several features. For instance, malicious apps are much more likely to share names with other apps and they typically request less permission than benign apps. Leveraging our observations, we developed FRAppE, an precise 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 excavate deeper into this ecosystem of malicious apps on Facebook, and we wish that Facebook will advantage from our recommendations for reducing the hazard of hackers on their platform.

REFERENCES

1. Facebook, Palo Alto, CA, USA, "Facebook Open graph API" [Online]. Available: http://developers.facebook.com/docs/reference/api/
2. "Wiki: Facebook platform," 2014 [Online]. Available: http://en. wikipedia.org/wiki/Facebook Platform

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

3. "Which cartoon character are you—Facebook survey scam," 2012 [Online]. Available: https://apps.facebook.com/mypage keeper/? Status = scam_ report _ fb _ survey _ scam _ which_ cartoon_character_are_you_2012_03_30
4. R.Naraine,"Hackers selling $25 tool kit to create malicious Facebook apps,"2011 [Online]. Available: http://zd.net/g28HxI
5. HackTrix,"Stay away from malicious Facebook apps,"2013[Online].Available: http://bit.ly/b6gWn5
6. M. S. Rahman, T.-K. Huang, H. V. Madhyastha, and M. Faloutsos, "Efficient and scalable socware detection in online social networks," in Proc. USENIX Security, 2012, p.32.
7. H. Gaoetal.,"Detecting and characterizing social spam campaigns," in Proc. IMC, 2010, pp. 35–47.
8. H.Gao,Y.Chen,K.Lee,D.Palsetia, and A.Choudhary,"Towards online spam altering in social networks," in Proc.NDSS,2012 .
9. "My Page Keeper," [Online]. Available: https://www.facebook.com/ apps/application.php?id=167087893342260
10. Facebook, Palo Alto, CA, USA, "Application authentication flow using OAuth 2.0," [Online]. Available: http://developers.facebook. com/docs/authentication/

BIOGRAPHY


Ch. Sahitya is a B.Tech scholar pursuing CSE in Sree Venkateswara College of Engineering, Nellore, AP.


M. RamyaSree is a B.Tech scholar pursuing CSE in Sree Venkateswara College of Engineering, Nellore, AP.


G. Madhurya is a B.Tech scholar pursuing CSE in Sree Venkateswara College of Engineering, Nellore, AP.


M. Kiran Kumar working as Associate Professor in CSE department in Sree Venkateswara College of Engineering, Nellore, AP. He completed his UG & PG in JNTU. His areas of interest are Cloud Computing, BigData and Wireless Sensor Networks.