International Journal of Research
Available at https://pen2print.org/index.php/ijr/

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 06 Issue 1
January 2019

Malicious Face Book Application Using FRAppE Algorithm

Gorantla Nagarjuna & M.Kiran Kumar

[1] PG Scholar Department of CSE, GVR & S College Of Engineering & Technology, Guntur (D.T),Andhra Pradesh

[2]Associate Professor,Department of CSE , GVR & S College Of Engineering & Technology, Guntur (D.T),Andhra Pradesh

Abstract:-

With 20 million introduces multi day, outsider applications are a noteworthy purpose behind the notoriety and addictiveness of Facebook. Lamentably, programmers have understood the capability of utilizing applications for spreading malware and spam. The issue is as of now noteworthy, as we find that at any rate 13% of applications in our dataset are vindictive. Up until this point, the exploration network has concentrated on identifying malignant posts and battles. In this paper, we make the inquiry: Given a Facebook application, would we be able to decide whether it is malignant? Our key commitment is in creating FRAppE—Facebook's Rigorous Application Evaluator—ostensibly the primary device concentrated on distinguishing malignant applications on Facebook. To create FRAppE, we utilize data accumulated by watching the posting conduct of 111K Facebook applications seen crosswise over 2.2 million clients on Facebook. To begin with, we recognize an arrangement of highlights that assistance us recognize malevolent applications from amiable ones. For instance, we locate that pernicious applications frequently share names with different applications, and they ordinarily ask for less consents than generous applications. Second, utilizing these distinctive highlights, we demonstrate that FRAppE can identify vindictive applications with 99.5% exactness, with no false positives and a high obvious positive rate (95.9%). At last, we investigate the environment of vindictive Facebook applications and distinguish components that these applications use to proliferate. Curiously, we locate that numerous applications conspire and bolster one another; in our dataset, we find 1584 applications empowering the viral proliferation of 3723 different applications through their posts. Long haul, we consider FRAppE to be a stage toward making an autonomous guard dog for application appraisal and positioning, in order to caution Facebook clients before introducing applications.

Keywords:- Facebook Apps, Malicious Apps, Profiling Apps, Online Social Networks.

I.INTRODUCTION

ONLINE social networks (OSNs) 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 that facilitates app integration into the Facebook user-experience. There are 500K apps available on Facebook , and on average, 20M apps are installed every day. Furthermore, many apps have acquired and maintain a really large user base. For instance, FarmVille and CityVille 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. Malicious apps can provide a lucrative business for hackers, given the popularity of OSNs, with Facebook leading the way with 900M active users. There are many ways that hackers can benefit from a malicious app: 1) the app can reach large numbers of users and their friends to spread spam; 2) the app can obtain users' personal information such as e-mail address, home town, and gender; and 3) the app can "reproduce" 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. In other words, there is motive and opportunity, and as a result, there are many malicious apps spreading on Facebook every day.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 the following: 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. As we show in Section III, 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. At the same time, in a seemingly backwards step, Facebook has dismantled iapp rating functionality recently. In this paper, we develop FRAppE, a suite of efficient classification techniques for identifying whether an app is malicious or not. To build FRAppE, we use data from MyPage- Keeper, a security app in Facebook that monitors the Facebook profiles of 2.2 million users. We analyze 111K apps that made 91 million posts over 9 months. This is arguably the first comprehensive study focusing on malicious Facebook apps that focuses on quantifying, profiling, and understanding malicious apps and synthesizes this information into an effective detection approach. Our work makes the following key contributions. *13% of observed apps are malicious*. We show that malicious apps are prevalent in Facebook and reach a large number of users.We find that 13% of apps in our dataset of 111K distinct apps are malicious. Also, 60% of malicious apps endanger more than 100K users each by convincing them to follow the links on the posts made by these apps, and 40% of malicious apps have over 1000 monthly active users each. *Malicious and benign app profiles significantly differ*. We systematically profile apps and show that malicious app profiles are significantly different than those of benign apps. A striking observation is the "laziness" of hackers; many malicious apps have the same name, as 8%of unique names of malicious apps are each used by more than 10 different apps (as defined by their app IDs). Overall, we profile apps based on two classes of features: 1) those that can be obtained on-demand given an application's identifier (e.g., the permissions required by the app and the posts in the application's profile page), and 2) others that require a cross-user view to aggregate information across time and across apps (e.g., the posting behavior of the app and the similarity of its name to other apps).

• *The emergence of app-nets: Apps collude at massive scale*. We conduct a forensics investigation on the malicious app ecosystem to identify and quantify the techniques used to promote malicious apps.We find that apps collude and collaborate at a massive scale. Apps promote other apps via posts that point to the "promoted" apps. If we describe the collusion relationship of promoting–promoted apps as a graph, we find 1584 promoter apps that promote 3723 other apps. Furthermore, these apps form large and highly dense connected components, as shown in this system.

Furthermore, hackers use fast-changing indirection: Applications posts have URLs that point to a Web site, and the Web site dynamically redirects to many different apps; we find 103 such URLs that point to 4676 different malicious apps over the course of a month. These observed behaviors indicate well-organized crime: One hacker controls many malicious apps, which we will call an app-net, since they seem a parallel concept to botnets. *Malicious hackers impersonate applications*. We were surprised to find popular good apps, such as FarmVille and Facebook for iPhone, posting malicious posts. On further investigation, we found a lax authentication rule in Facebook that enabled hackers to make malicious posts appear as though they came from these apps. *FRAppE can detect malicious apps with 99% accuracy*. We develop FRAppE (Facebook's Rigorous Application Evaluator) to identify malicious apps using either using only features that can be obtained on-demand or using both on-demand and aggregation-based app information. FRAppE Lite, which only uses information available on-demand, can identify malicious apps with 99.0% accuracy, with low false positives (0.1%) and high true positives (95.6%). By adding aggregation-based information, FRAppE can detect malicious apps with 99.5% accuracy, with no false positives and higher true positives (95.9%). *Our recommendations to Facebook*. The most important message of the work is that there seems to be a parasitic eco-system of malicious apps within Facebook that needs to be understood and stopped. However, even this initial work leads to the following recommendations for Facebook that could potentially also be useful to other social platforms 1) *Breaking the cycle of app propagation*. We recommend that apps should not be allowed to promote other apps. This is the reason that malicious apps seem to gain strength by self-propagation. Note that we only suggested against a special kind of app promotion where the user clicks the app A installation icon, app A redirects the user to the intermediate installation page of app B, and the user cannot see the difference unless she examines the landing URL very carefully where client ID is different. At the end, the user ends up installing app B although she intended to install app A. Moreover, cross promotion among apps is forbidden as per Facebook's platform policy. *Enforcing stricter app authentication before posting*. We recommend a stronger authentication of the identity of an app before a post by that app is accepted. As we saw, hackers fake the true identify of an app in order to evade detection and appear more credible to the end user.

II. LITERATURE SURVEY

1) A technique for computer detection and correction of spelling errors

AUTHORS: F. J. Damerau

The method described assumes that a word which cannot be found in a dictionary has at most one error, which might be a wrong, missing or extra letter or a single transposition. The unidentified input word is compared to the dictionary again, testing each time to see if the words match—assuming one of these errors occurred. During a test run on garbled text, correct identifications were made for over 95 percent of these error types.

2) LIBSVM: A library for support vector machines

AUTHORS: C.-C. Chang and C.-J. Lin

LIBSVM is a library for Support Vector Machines (SVMs). We have been actively developing this package since the year 2000. The goal is to help users to easily apply SVM to their applications. LIBSVM has gained wide popularity in machine learning and many other areas. In this article, we present all implementation details of LIBSVM. Issues such as solving SVM optimization problems theoretical convergence multiclass classification probability estimates and parameter selection are discussed in detail.

3) Beyond blacklists: Learning to detect malicious Web sites from suspicious URLs

AUTHORS: J. Ma, L. K. Saul, S. Savage, and G. M. Voelker

Malicious Web sites are a cornerstone of Internet criminal activities. As a result, there has been broad interest in developing systems to prevent the end user from visiting such sites. In this paper, we describe an approach to this problem based on automated URL classification, using statistical methods to discover the tell-tale lexical and host-based properties of malicious Web site URLs. These methods are able to learn highly predictive models by extracting and automatically analyzing tens of thousands of features potentially indicative of suspicious URLs. The resulting classifiers obtain 95-99% accuracy, detecting large numbers of malicious Web sites from their URLs, with only modest false positives.

4) Design and evaluation of a real-time URL spam filtering service

AUTHORS: K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song

On the heels of the widespread adoption of web services such as social networks and URL shorteners, scams, phishing, and malware have become regular threats. Despite extensive research, email-based spam filtering techniques generally fall short for protecting other web services. To better address this need, we present Monarch, a real-time system that crawls URLs as they are submitted to web services and determines whether the URLs direct to spam. We evaluate the viability of Monarch and the fundamental challenges that arise due to the diversity of web service spam. We show that Monarch can provide accurate, real-time protection, but that the underlying characteristics of spam do not generalize across web services. In particular, we find that spam targeting email qualitatively differs in significant ways from spam campaigns targeting Twitter. We explore the distinctions between email and Twitter spam, including the abuse of public web hosting and redirector services. Finally, we demonstrate Monarch's scalability, showing our system could protect a service such as Twitter--which needs to process 15 million URLs/day--for a bit under $800/day.

5) Detecting spammers on social networks

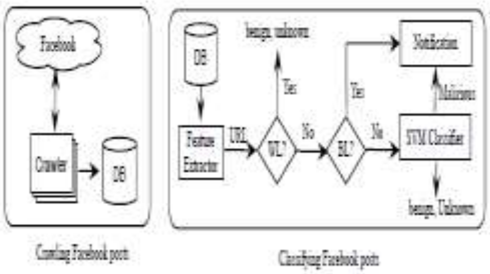
AUTHORS: G. Stringhini, C. Kruegel, and G.

Social networking has become a popular way for users to meet and interact online.

Users spend a significant amount of time on popular social network platforms (such as Facebook, MySpace, or Twitter), storing and sharing a wealth of personal information. This information, as well as the possibility of contacting thousands of users, also attracts the interest of cybercriminals. For example, cybercriminals might exploit the implicit trust relationships between users in order to lure victims to malicious websites. As another example, cybercriminals might find personal information valuable for identity theft or to drive targeted spam campaigns.

In this paper, we analyze to which extent spam has entered social networks. More precisely, we analyze how spammers who target social networking sites operate. To collect the data about spamming activity, we created a large and diverse set of "honey-profiles" on three large social networking sites, and logged the kind of contacts and messages that they received. We then analyzed the collected data and identified anomalous behavior of users who contacted our profiles. Based on the analysis of this behavior, we developed techniques to detect spammers in social networks, and we aggregated their messages in large spam campaigns. Our results show that it is possible to automatically identify the accounts used by spammers, and our analysis was used for take-down efforts in a real-world social network. More precisely, during this study, we collaborated with Twitter and correctly detected and deleted 15,857 spam profiles.

SYSTEM ARCHITECTURE:



III. Related Work:- Detecting spam on OSNs. Gao et al. [32] analyzed posts on the walls of 3.5 million Facebook users and showed that 10% of links posted on Facebook walls are spam. They also presented techniques to identify compromised accounts and spam campaigns. In other work, Gao et al. [31] and Rahman et al. [41] develop efficient techniques for online spam filtering on OSNs such as Facebook. While Gao et al. [31] rely on having the whole social graph as input,

and so, is usable only by the OSN provider, Rahman et al. [41] develop a third-party application for spam detection on Facebook. Others [37,44] present mechanisms for detection of spam URLs on Twitter. In contrast to all of these efforts, rather than classifying individual URLs or posts as spam, we focus on identifying malicious applications that are the main source of spam on Facebook.

Detecting spam accounts. Yang et al. [46] and Benevenuto et al. [26] developed techniques to identify accounts of spammers on Twitter. Others have proposed a honey-pot based approach [36,43] to detect spam accounts on OSNs. Yardi et al. [47] analyzed behavioral patterns among spam accounts in Twitter. Instead of focusing on accounts created by spammers, our work enables detection of malicious apps that propagate spam and malware by luring normal users to install them. App permission exploitation. Chia et al. [29] investigated the privacy intrusiveness of Facebook apps and concluded that currently available signals such as community ratings, popularity, and external ratings such as Web of Trust (WOT) as well as signals from app developers are not reliable indicators of the privacy risks associated with an app. Also, in keeping with our observation, they found that popular Facebook apps tend to request more permissions. They also found that 'Lookalike' applications that have names similar to popular applications request more permissions than is typical. Based on a measurement study across 200 Facebook Facebook users, Liu et al. [38] showed that privacy settings in Facebook rarely match users' expectations. To address the privacy risks associated with the use of Facebook apps, some studies [27, 45] propose a new application policy and authentication dialog. Makridakis et al. [40] use a real application named 'Photo of the Day' to demonstrate how malicious apps on Facebook can launch DDoS attacks using the Facebook platform. King et al. [34] conducted a survey to understand users' interaction with Facebook apps. Similarly, Gjoka et al. [33] study the user reach of popular Facebook applications. On the contrary, we quantify the prevalence of malicious apps, and develop tools to identify malicious apps that use several features beyond the required permission set. App rating efforts. Stein et al. [42] describe Facebook's Immune System (FIS), a scalable real-time adversarial learning system deployed in Facebook to protect users from malicious activities. However, Stein et al. provide only a high-level overview about threats to the Facebook graph and do not provide any analysis of the system. Furthermore, in an attempt to balance accuracy of detection with low false positives, it appears that Facebook has recently softened their controls for handling spam apps [11]. Other Facebook applications [5,7,15] that defend users against spam and malware do not provide ratings for apps on Facebook. Whatapp [23] collects community reviews about apps for security, privacy and openness. However, it has not attracted much reviews (47 reviews available) to date. To the best of our knowledge, we are the first to provide a classification of Facebook apps into malicious and benign categories.

IV.CONCLUSION

Applications present 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 paper, using a large corpus of malicious Facebook apps observed over a 9-month period, we showed thatmalicious apps differ significantly from benign apps with respect to several features. For example, malicious apps aremuchmore likely to share names with other apps, and they typically request fewer permissions than benign apps. Leveraging our observations, we developed FRAppE, 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.

V.BIBLIOGRAPHY

References Made From:-

[1] C. Pring, “100 social media statistics for 2012,” 2012 [Online]. Available: http://thesocialskinny.com/100-social-media-statistics-for-2012/

[2] Facebook, Palo Alto, CA, USA, “Facebook Opengraph API,” [Online]. Available: http://developers.facebook.com/docs/reference/api/

[3] “Wiki: Facebook platform,” 2014 [Online]. Available: http://en. wikipedia.org/wiki/Facebook_Platform

[4] “Pr0file stalker: Rogue Facebook application,” 2012 [Online]. Available: https://apps.facebook.com/mypagekeeper/?status=scam_report-_fb_survey_scam_pr0file_viewer_2012_4_4

[5] “Whiich cartoon character are you—Facebook survey scam,” 2012 [Online]. Available: https://apps.facebook.com/mypagekeeper/?status=scam_report_fb_survey_scam_whiich_cartoon_character_are_you_2012_03_30

[6] G. Cluley, “The Pink Facebook rogue application and survey scam,” 2012 [Online]. Available: http://nakedsecurity.sophos.com/2012/02/27/pink-facebook-survey-scam/

[7] D. Goldman, “Facebook tops 900 million users,” 2012 [Online]. Available: http://money.cnn.com/2012/04/23/technology/facebookq1/index.htm

[8] R. Naraine, “Hackers selling $25 toolkit to create malicious Facebook apps,” 2011 [Online]. Available: http://zd.net/g28HxI

[9] HackTrix, “Stay away from malicious Facebook apps,” 2013 [Online]. Available: http://bit.ly/b6gWn5

[10] 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.

[11] H. Gao *et al.*, “Detecting and characterizing social spam campaigns,” in *Proc. IMC*, 2010, pp. 35–47.

[12] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary, “Towards online spam filtering in social networks,” in *Proc. NDSS*, 2012.

[13] P. Chia, Y. Yamamoto, and N. Asokan, “Is this app safe? A large scale study on application permissions and risk signals,” in *Proc. WWW*, 2012, pp. 311–320.

[14] “WhatApp? (beta)—A Stanford Center for Internet and Society Website with support from the Rose Foundation,” [Online]. Available: https://whatapp.org/facebook/

[15] “MyPageKeeper,” [Online]. Available: https://www.facebook.com/ apps/application.php?id=167087893342260

[16] Facebook, Palo Alto, CA, USA, “Facebook platform policies,” [Online]. Available: https://developers.facebook.com/policy/

[17] Facebook, Palo Alto, CA, USA, “Application authentication flow using OAuth 2.0,” [Online]. Available: http://developers.facebook. com/docs/authentication/

[18] “11 million bulk email addresses for sale—Sale price $90,” [Online]. Available: http://www.allhomebased.com/BulkEmailAddresses.htm

[19] E. Protalinski, “Facebook kills app directory, wants users to search for apps,” 2011 [Online]. Available: http://zd.net/MkBY9k

[20] SocialBakers, “SocialBakers: The recipe for socialmarketing success,” [Online]. Available: http://www.socialbakers.com/

[21] “Selenium—Web browser automation,” [Online]. Available:http://seleniumhq.org/

[22] “bit.ly API,” 2012 [Online]. Available: http://code.google.com/p/bitlyapi/ wiki/ApiDocumentation

[23] Facebook, Palo Alto, CA, USA, “Permissions reference,” [Online]. Available: https://developers.facebook.com/docs/authentication/permissions/

[24] Facebook, Palo Alto, CA, USA, “Facebook developers,” [Online]. Available: https://developers.facebook.com/docs/appsonfacebook/tutorial/

[25] “Web-of-Trust,” [Online]. Available: http://www.mywot.com/

[26] F. J. Damerau, “A technique for computer detection and correction of spelling errors,” *Commun. ACM*, vol. 7, no. 3, pp. 171–176,Mar. 1964.

[27] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[28] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Beyond blacklists: Learning to detect malicious Web sites from suspicious URLs,” in *Proc. KDD*, 2009, pp. 1245–1254.

[29] A. Le, A.Markopoulou, and M. Faloutsos, “PhishDef: URL names say it all,” in *Proc. IEEE INFOCOM*, 2011, pp. 191–195.

[30] C. Wueest, “Fast-flux Facebook application scams,” 2014 [Online]. Available: http://www.symantec.com/connect/blogs/fast-fluxfacebook-application-scams

[31] “Longest path problem,” 2014 [Online]. Available: http://en.wikipedia. org/wiki/Longest_path_problem

[32] “App piggybacking example,” [Online]. Available: https://apps.facebook.com/mypagekeeper/?status=scam_report_fb_survey_scam_ Converse_shoes_2012_05_17_boQ

[33] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, “Design and evaluation of a real-time URL spam filtering service,” in *Proc. IEEE Symp. Security Privacy*, 2011, pp. 447–462.

[34] S. Lee and J. Kim, “WarningBird: Detecting suspicious URLs in Twitter stream,” in *Proc. NDSS*, 2012.

[35] C. Yang, R. Harkreader, and G. Gu, “Die free or live hard? Empirical evaluation and new design for fighting evolving Twitter spammers,” in *Proc. RAID*, 2011, pp. 318–337.

[36] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, “Detecting spammers on Twitter,” in *Proc. CEAS*, 2010, pp. 1–9.

[37] G. Stringhini, C. Kruegel, and G. Vigna, “Detecting spammers on social networks,” in *Proc. ACSAC*, 2010, pp. 1–9.

[38] K. Lee, J. Caverlee, and S.Webb, “Uncovering social spammers: Social honeypots + machine learning,” in *Proc. SIGIR*, 2010, pp. 435–442.

[39] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd, “Detecting spam in a twitter network,” *First Monday*, vol. 15, no. 1, 2010 [Online]. Available: http://firstmonday.org/ojs/index.php/fm/article/view/2793/2431

[40] A. Besmer, H. R. Lipford, M. Shehab, and G. Cheek, “Social applications: Exploring a more secure framework,” in *Proc. SOUPS*, 2009,Art. no. 2.

[41] N. Wang, H. Xu, and J. Grossklags, “Third-party apps on Facebook: Privacy and

the illusion of control," in *Proc. CHIMIT*, 2011, Art. no.4.

[42] A. Makridakis *et al.*, "Understanding the behavior of malicious applications in social networks," *IEEE Netw.*, vol. 24, no. 5, pp. 14–19, Sep.–Oct. 2010.

[43] J. King, A. Lampinen, and A. Smolen, "Privacy: Is there an app for that?," in *Proc. SOUPS*, 2011, Art. no. 12.

[44] M. Gjoka, M. Sirivianos, A. Markopoulou, and X. Yang, "Poking Facebook: Characterization of OSN applications," in *Proc. 1st WOSN*, 2008, pp. 31–36.

[45] T. Stein, E. Chen, and K. Mangla, "Facebook immune system," in *Proc. 4th Workshop Social Netw. Syst.*, 2011, Art. no. 8.

[46] L. Parfeni, "Facebook softens its app spam controls, introduces better tools for developers," 2011 [Online]. Available: http://bit.ly/LLmZpM

[47] "Norton Safe Web," [Online]. Available: http://www.facebook.com/ apps/application.php?id=310877173418

[48] "Bitdefender Safego," [Online]. Available: http://www.facebook.com/ bitdefender.safego