International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 4, April 2016

Detecting Malicious Apps on OSN Face-book Wall

[1]Ashutosh Lande, [2]Akshay Patil, [3]Pankaj Bondre , [4]Akshata Phawde, [5]Pradip Laturkar
[1][2][3][4] UG student, [5] Asst.Professor
[1][2][3][4] [5] DYPSOET
[1]landeashutosh85@gmail.com,[2] akkypatil646@gmail.com, [3]pankajbondre786@gmail.com,
[4]akshata1831@gmail.com, [5]aplaturkar@gmail.com

***Abstract*: In Online Social Networking (OSN), unfortunately, hackers have realized the potential of using apps for spreading malware and spam which are harmful to Face-book users. The problem is already significant, as we find that at least 13% of apps in our dataset are malicious. So far, the research community has focused on detecting malicious posts and campaigns. In this project, we ask the question to the Face-book user that, given a Face-book application, can you determine whether that application is malicious? Of course that user couldn't identify that. So, our key contribution is in developing "FRAppE—Face-book's Rigorous Application Evaluator", arguably the first tool focused on detecting malicious apps on Facebook. To develop FRAppE, we use information gathered by observing the posting behavior of 111K Face-book apps seen across 2.2 million users on Face-book. First, we identify a set of features that help us distinguish between malicious apps and benign apps. For example, we find that malicious apps often share names with other apps, and they typically request little permission than benign apps. Second, leveraging these distinguishing features, we show that FRAppE can detect malicious apps with 99.5% accuracy, with no false positives and a low false negative rate (4.1%). Finally, we explore the ecosystem of malicious Face-book apps and identify mechanisms that these apps use to propagate. Interestingly, we find that many apps collude and support each other; in our dataset, we find 1,584 apps enabling the viral propagation of 3,723 other apps through their posts. Long-term, we see FRAppE as a step towards creating an independent watchdog for app assessment and ranking, so as to warn Face-book users before installing apps.**

***Index Terms*— Face-book Apps, Malicious Apps, Profiling Apps, Online Social Networks**

I. INTRODUCTION

The social networking sites are making our social lives better but nevertheless there are a lot of issues with using these social networking sites. The issues are privacy, online bullying, potential for misuse, trolling, etc. These are done mostly by using fake applications or malicious applications spread by hacker or untrusted server.

Recently, hackers have started taking advantage of the popularity of this third-party apps platform and deploying malicious applications which can provide a lucrative business for hackers, given by the popularity of OSNs, with Face-book leading the way with 900M active users. There are many ways that hackers can benefit from a malicious app. To make matters worse, the deployment of malicious apps is simplified by ready-to-use toolkits. In other words, there is motive and opportunity, and as a result, there are many malicious apps spreading on Face-book every day. Online social networks (OSN) enable and encourage third party applications to enhance the user experience on these platforms like FACE-BOOK. 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, Face-book provides developers an API that facilitates app integration into the Face-book user-experience. There are 500K apps available on Face-book, and on average, 20M apps are installed every day. Furthermore, many apps have acquired and maintain a large user base.

II. LITERATURE REVIEW

1. Detecting and Characterizing Social Spam Campaigns (2010).

AUTHORS: Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, Ben Y. Zhao.

Description:

In this paper, authors presented an initial study to quantify and characterize spam campaigns launched using accounts on online social networks. They studied a large anonymized dataset of asynchronous "wall" messages between Facebook users. We analyze all wall messages received by roughly 3.5 million Facebook users (more than 187 million messages in all), and use a set of automated techniques to detect and characterize coordinated spam campaigns. System detected roughly 200,000 malicious wall posts with embedded URLs, originating from more than 57,000 user accounts. Authors

found that more than 70% of all malicious wall posts advertise phishing sites.

They study the characteristics of malicious accounts, and see that more than 97% are compromised accounts, rather than "fake" accounts created solely for the purpose of spamming. Finally, when adjusted to the local time of the sender, spamming dominates actual wall post activity in the early morning hours, when normal users are asleep.

***2.Is this App Safe? A Large Scale Study on Application Permissions and Risk Signals (2012*).**

AUTHORS: Pern Hui Chia, Yusuke Yamamoto, N. Asokan

Description:

Third-party applications (apps) drive the attractiveness of web and mobile application platforms. Many of these platforms adopt a decentralized control strategy, relying on explicit user consent for granting permissions that the apps request. Users have to rely primarily on community ratings as the signals to identify the potentially harmful and inappropriate apps even though community ratings typically reflect opinions about perceived functionality or performance rather than about risks. With the arrival of HTML5 web apps, such user-consent permission systems will become more widespread. We study the effectiveness of user-consent permission systems through a large scale data collection of Face-book apps, Chrome extensions and Android apps. The analysis confirms that the current forms of community ratings used in app markets today are not reliable indicators of privacy risks of an app. We find some evidence indicating attempts to mislead or entice users into granting permissions: free applications and applications with mature content request more permissions than is typical; "lookalike" applications which have names similar to popular application.

3. LIBSVM: A Library for Support Vector Machines (2011).

AUTHORS: Chih-Chung Chang and Chih-Jen Lin

Description:

LIBSVM is a library for Support Vector Machines (SVMs). Authors 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, authors presented all implementation details of LIBSVM. Issues such as solving SVM optimization problems, theoretical convergence, multi-class classification, probability estimates, and parameter selection are discussed in detail. Support Vector Machines (SVMs) are a popular machine learning method for classification, regression, and other learning tasks. LIBSVM is currently one of the most widely used SVM software.

4. Social Applications: Exploring A More Secure Framework (2009).

AUTHORS:

Description:

Online social network sites, such as MySpace, Face-book and others have grown rapidly, with hundreds of millions of active users. A new feature on many sites is social applications applications and services written by third party developers that provide additional functionality linked to a user's profile. However, current application platforms put users at risk by permitting the disclosure of large amounts of personal information to these applications and their developers. This paper formally abstracts and defines the current access control model applied to these applications, and builds on it to create a more secure framework. We do so in the interest of preserving as much of the current architecture as possible, while seeking to provide a practical balance between security and privacy needs of the users, and the needs of the applications to access users' information. We present a user study of our interface design for setting a user-to-application policy. Our results indicate that the model and interface work for users who are more concerned with their privacy, but we still need to explore alternate means of creating policies for those who are less concerned.

5. Trust evaluation on Facebook using multiple contexts

***AUTHORS*:** Tomá, Jan Samek

***Description*:**

This paper applies the term trust from the point of view of artificial intelligence to social network analysis methods. It evaluates current available interactions for a model of trust considering various social networks. A mathematical model of trust for Facebook is designed. This model is implemented in Python programming language. Experiments are conducted on a sample amount of Face-book users and furthermore analyzed from the perspective of both artificial intelligence and social psychology.

III. SURVEY OF PROPOSED SYSTEM

In this work, 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 MyPageKeeper. To build FRAppE, we use data from MyPageKeeper, 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 nine months. This is arguably the first comprehensive study focusing on malicious Face-book apps that focuses on quantifying, profiling, and understanding malicious apps, and synthesizes this information into an effective detection approach.

We have introduced two features i.e. classifiers to detect the malicious apps FRAppE Lite and FRAppE . In first classifier it detect the initial level detection e.g. apps identity number , name and source etc and in second level detection the actual detection of malicious app has been done..

IV. MODULES

1. User

The user firstly regeister himself with the system after that he will sign in to his account & send request to system for adding new application to his profile & wait for response.

2. System Server

Verify users & his request. The app request will forward to application server send token request for application to user which contains user's personal information

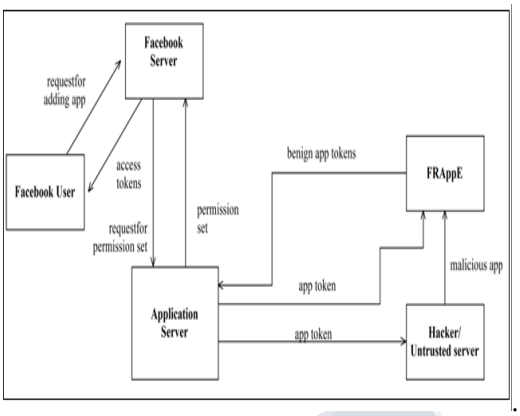
3. Application server

Saves all data about application such as ID of apps with respect to location of app(URL)

4. FRAppE

FRAppELite:-It contains basic information of application like name, Id, location etc like the MYPAGEKEEPER of face book which only crawls post on the walls of application. FRAppE checks whether the application is malicious or benign. If app is malicious it alerts the user with respect to application server.

V. SYSTEM ARCHITECTURE



VI. EXPECTED RESULTS

A.Comparison of proposed FRAppE system with existing facebook for detecting malicious apps.

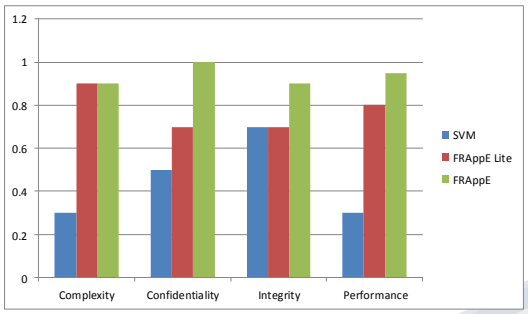
Paremeters	Existing (mypagekeeper in OSN)	Proposed (FRAppE)
Complexicity	O(n)	O(n2)
confidentiality	Low	high
Integrity	Low	high

B. Analysis graph of different classifiers.

Parameters	SVM	FRAppE Lite	FRAppE
Complexicity	O(n)	O(n2)	O(n2)
confidentiality	Low	Average	high
Integrity	average	Average	high
Performance	High	Good	better

Parameters	SVM	FRAppE Lite	FRAppE

Complexity	0.3	0.9	0.9
confidentiality	0.5	0.7	1
Integrity	0.7	0.7	0.9
Performance	0.3	0.8	0.95



Algorithm

Step 1: Start

Step 2: At first user sends request to server for adding an application to his profile like some game app etc.
At initial user don't aware that whether requested application is benign or malicious as user has knows name of that application so he blindly request for adding use that app

Step 3: When user request comes at server side it returns the one access request which contains the permissions required to app which user want to install on his profile , permissions like , Application wants to access user information from profile like name, date of birth etc. and this token are send to application server.

Step 4: In this step user returns the permission set to allow the access the information from his profile to that particular app and the user request is forwarded to application server. Here user doesn't aware that whether that app is benign or malicious so, here our FRAppE comes in picture.

FRAppE checks whether that app is malicious or benign by applying some classifications such as FRAppE Lite and FRAppE.

FRAppE Lite: This is the initial level for detection or classifier i.e. FRAppE Lite checks the application ID no, name and location of application and verifies with the available benign application in the application server. The FRAppE takes the help of mypagekeeper which is our primary dataset in facebook for maintaining the record of authenticated apps

Step 6: End

VII. CONCLUSION

An application presents a convenient means for hackers to spread malicious content on Face-book. However, little is understood about the characteristics of malicious apps and how they operate. In this project, using a large corpus of malicious Face-book 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 little permission than benign apps. Leveraging our observations, we developed FRAppE, an accurate classifier for detecting malicious Face-book applications. Most interestingly, we highlighted the emergence of AppNets large groups of tightly connected applications that promote each other. The application which are malicious their review, ranking and reporting will be done.

REFERENCES

[1] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In IMC, 2010.

[2]P. Chia, Y. Yamamoto, and N. Asokan. Is this app safe? A large scale study on application permissions and risk signals.In WWW, 2012.

[3]C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2, 2011.

[4]A. Besmer, H. R. Lipford, M. Shehab, and G. Cheek. Social applications: exploring a more secure framework. In SOUPS,2009.

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

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

[7] "Wiki: Facebook platform," 2014 [Online]. Available: http://en. wikipedia.org/wiki/Facebook_Platform

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

[9] "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

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

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

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

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

[14] 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.

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

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