IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

FRAppE: Identifying Malicious Facebook Application

Urvashi Yadav[1], Khushi Kumari[2], Rohini Lokare[3], Suhasini Borge[4], Prof. D.V. Shinkar[5]

UG, Information Technology, JSPM's BSIOTR, Pune, India[1, 2, 3, 4]

Asst. Prof, Information Technology, JSPM's BSIOTR, Pune, India[5]

Abstract: Nowadays use of social networking site like Facebook, Twitter, Google+ for communication and maintaining relationship among various user is increased due to its popularity on network. Each user that uses the social networking sites are making profiles and uploading their private information. These social networks users are not aware of numerous security risk included in this networks like privacy, identity theft and sexual harassment and so on. The third party apps on social sites have main role to make the site more attractive and incredible. The hackers are using these third party apps to get the private information and get unauthorized access to their accounts. As we aware that not most but least of the applications on sites are malicious. As research goes on the research community has focused on detecting malicious wall-posts and campaigns. In this paper, we are going to find that applications are malicious or not? In earlier system, It is important to note that mypagekeeper that is our base data, cannot detect malicious apps; it only detects malicious posts on Facebook. Those malicious apps contain the bunch of malicious posts. In contrast, frappe Lite and frappe are designed to detect malicious apps. Therefore the frappe or frappe Lite that is being developed is more powerful than mypagekeeper to develop frappe, we use information gathered by observing the posting behavior of basic Facebook apps that are running on it. So, first we try to find out the features of malicious apps and other characteristics of malicious apps that are harmful to users.

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

I. INTRODUCTION

In this paper, we are discussing about FRAppE, a suite of efficient classification techniques for identifying whether an app is malicious or not. To build FRAppE, we use data from MyPageKeeper, a security app in Facebook that monitors the Facebook profiles. 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. Here we are generating the OTP (one time password) whenever user want to send a message to another user, or want to upload the picture. This guarantees the safety of user data from other person and doesn't allow third party to do changes to the account of any user as OTP is required whenever we are doing any activity in our account. In our model, it also generates the graph according to the attack of malicious app.

II. BACKGROUND

A. MyPageKeeper

MyPageKeeper is a Facebook app designed for detecting malicious posts on Facebook. Once a Facebook user installs MyPageKeeper, it periodically crawls posts from the user's wall and news feed. MyPageKeeper then applies URL blacklists as well as custom classification techniques to identify malicious posts. The key thing to note here is that MyPageKeeper identifies social malware at the granularity of individual posts, without grouping together posts made by any given application. In other words, for every post that it crawls from the wall or news feed of a subscribed user.

III. PROPOSED SYSTEM

A. Exiting System:

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.

Disadvantages of Existing System:

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 email address, hometown, and gender, and
3. The app can re-produce" by making other malicious apps popular.

B. 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, a security app in Facebook that monitors the Facebook profiles. This is arguably the first comprehensive study focusing on malicious Facebook apps that focuses on quantifying, profiling, and understanding malicious apps, and synthesizes this

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

information into an effective detection approach. Here we are generating the OTP (one time password) whenever user want to send a message to another user, or want to upload the picture. This guarantees the safety of user data from other person and doesn't allow third party to do changes to the account of any user as OTP is required whenever we are doing any activity in our account. In our model, it also generates the graph according to the attack of malicious app.

C. Architecture of Proposed System :

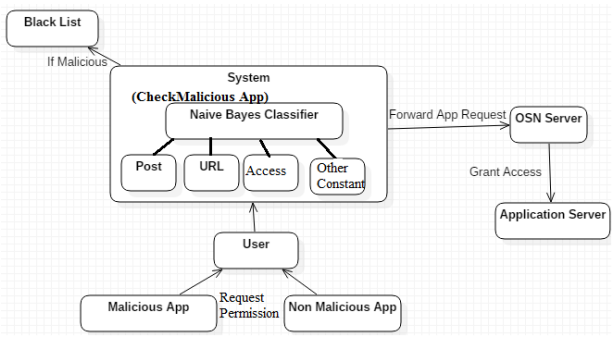


Fig 1 Architecture of Proposed System

The architectural design elaborate about what the actual system is. As shown in fig 1. Our system will detect weather the application is malicious or not by using Naïve Bayes classifier algorithm .As shown in fig App is popped to user and user gives request to server to use this app but before this request is going to proceed we will check whether the application is malicious or not by applying constraints on app(constraints such as is that app have suspicious redirecting url, app post contents, app close functions etc.). Otherwise it will pass that app request to server. Then server gives permission to user to access that app.

IV. ALGORITHM

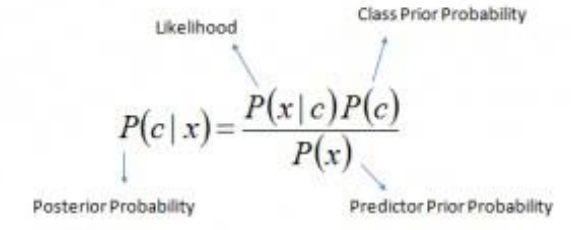
A. Naive Bayes Algorithm For Checking App Is Malicious Or Not

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors. In simple terms, a Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

In our paper Naïve Bayes is used for checking app is malicious or not. According to permission set, it contains the number of permissions required to access any app, but as we know that malicious app need fewer permissions to access anyone's info as compared to benign apps that needs to satisfy all the criteria to access the same things. In our app, we are using naïve Bayes algorithm which fills the information in the feature table like 1 or 0 according to whether the app asked for permission or not.

Bayes theorem provides a way of calculating posterior probability P(c—x) from P(c), P(x) and P(x—c). Look at the equation below:

1. P(c—x) is the posterior probability of class (c, target) given predictor (x, attributes).
2. P(c) is the prior probability of class.
3. P(x—c) is the likelihood which is the probability of predictor given class.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

B. How Naïve Bayes is used in our project:

Now, we need to classify whether Application is malicious or not based on permission set condition. Let's follow the below steps to perform it.

1. Convert the data set into a frequency tables.Here we are considering permission set. We are assuming 1 means permission is required and 0 means permission is not required.

Dataset Table according to our Project as follows:

Permission Set	Condition Values
Publish Stream	0
Offline Access	1
User Birth Date	1
Email	1
Publish Access	1
Publish Stream	1
Offline Access	1
User Birth Date	0
Email	0
Publish Access	0

Table 1.Dataset Table

Permission Set	0(permission not required)	1(permission required)
Publish Stream	1	1
Offline Access	0	2
User Birth Date	1	1
Email	1	1
Publish Access	1	1

Table 2.Frequency Table

DOI 10.17148/IJARCCE.2016.54182

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

2. Create Likelihood table by finding the probabilities like Publish Stream probability = 0.20 and probability of Non malicious app is 0.60.

Permission Set	0	1
Publish Stream	1	1
Offline Access	0	2
User Birth Date	1	1
Email	1	1
Publish Access	1	1
	All=4	All=6
	4/10=0.40	6/10=0.60

Table3.Likehood Table

3. Now, use Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

4. **Problem:** App is non malicious if permission set is 1 for Publish Stream .Is this statement is correct?

Consider Yes=1 and 0=No

We can solve it using above discussed method of posterior probability.

P(Yes | Publish Stream) = P(Publish Stream | Yes) * P(Yes) / P (Publish Stream)

Here we have

P (Publish Stream |Yes) = 1/6= 0.16,

P(Publish Stream) = 2/10= 0.36,

P(Yes)= 6/10 = 0.60

Now,

P (Yes | Publish Stream) = 0.16*0.6/0.2=0.48, which has lower probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

V. IMPLEMENTATION

A. OTP Generation Process :

It is the process of authentication where the OTP is generated at the time of message sending to other user or uploading photo. This helps to give confirmation to the system that the correct user is using the system as the OTP is generated at your email account whose password is not known to the hacker or the third party. So it totally safe as user is making the changes to account and OTP is helping them to do secure changes.

Basically, a onetime password (OTP) is an automatic- ally generated numeric or alphanumeric string of characters that authenticates the user for a single transaction or session.

Here we are using the numeric string in our OTP as shown in fig 2. An OTP is more secure than a static password, especially a user-created password, which is typically weak. OTPs may replace authentication login information or may be used in addition to it, to add another layer of security.

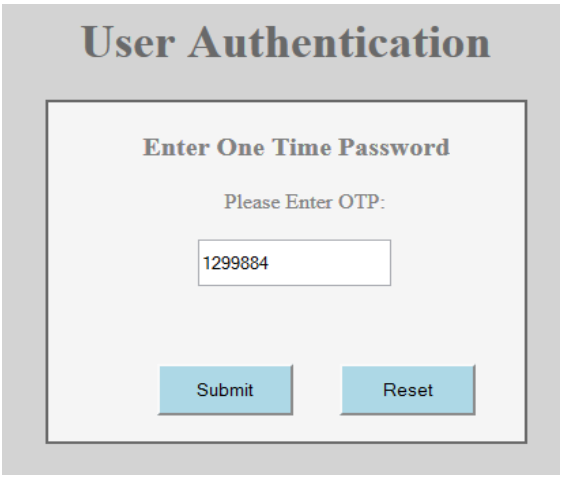


Fig 2. Authentication Process

B. Graph Generation Process

Attack graphs are important tools for analysing security vulnerabilities in enterprise networks. Attack graph increases as soon as user enters the wrong OTP which indicates the he is not the right user or malicious attack is gone to be happen as shown in fig 3.

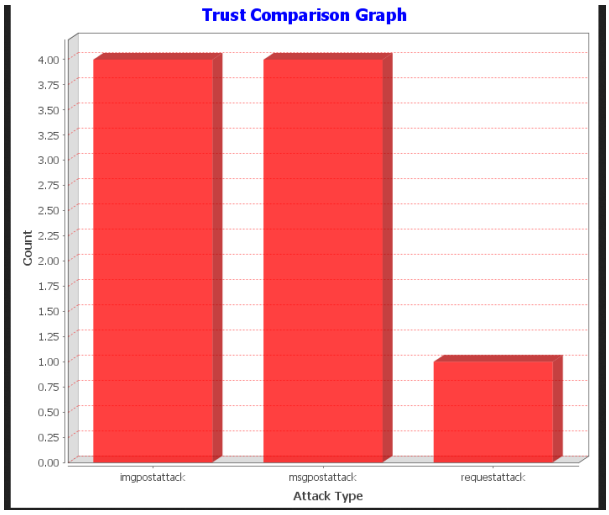


Fig3. Trust comparison graph

C. Malicious app features

We are adding two more features in our data base that helps to distinguish between malicious and begin apps, i.e. app name similarity & external link to post ratio as shown in fig 4.

Feature	Description
App name similarity	Is app's name identical to a known malicious app?
External link to post ratio	Fraction of app's posts that contain links to domains outside Facebook

Fig4. Additional features used in FRAppE.

VI. CONCLUSION

OSN Applications present a suitable means for spammers to spread harmful content on Social networks. In this paper, using a large amount of malicious social apps, we

DOI 10.17148/IJARCCE.2016.54182

showed that malicious apps differ drastically from normal apps with respect to a number of features. That's Why we are using Naïve Bayes classifier to classify the apps with respect to their feature for Example, post, URL, access permissions etc.

ACKNOWLEDGMENT

It gives us great pleasure in presenting the paper on 'FRAppE: Identifying Malicious Facebook Applications'. We would like to take this opportunity to thank my internal guide **Asst. Prof. D.V. Shinkar** for giving us all the help and guidance needed. We are really grateful to them for their kind support. Her valuable suggestions were very helpful. We are also grateful to **Asst. Prof. A.H. Tidake** Head of Information Technology Department, for her indispensable support, suggestions. In the end our special thanks to our principal **Dr. T.K. Nagaraj** for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Paper.

REFERENCES

[1] P.Chia,Y. Yamamoto, and N. Asokan. Is this app safe? a large scale study on application permissions and risk signals. In WWW, 2012.
[2] M.S. Rahman, T.-K. Huang, H. V. Madhyastha, and M. Faloutsos. Efficient and Scalable Socware Detection in Online Social Networks. In USENIX Security, 2012.
[3] N.Wang, H. Xu, and J. Grossklags. Third-party apps on facebook: privacy and the illusion of control. In CHIMIT, 2011.
[4] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary. Towards online spam ltering in social networks. In NDSS, 2012.
[5] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and Evaluation of a Real-Time URL Spam Filtering Service. In Proceedings of the IEEE Symposium on Security and Privacy, 2011.
[6] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao.
[7] Detecting and characterizing social spam campaigns. In IMC, 2010.