International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

A Survey Paper on FRAPPE – Facebook Rigorous Application Evaluator

Nitya Sree P[1], Sajitha R Swathi[2], Vishwanatha[3]

[1]Student, Department of CSE, Sri Krishna Institute of Technology, Bengaluru
[2]Student, Department of CSE, Sri Krishna Institute of Technology, Bengaluru
[3]Assistant Professor, Department of CSE, Sri Krishna Institute of Technology, Karnataka, India

---

Abstract - *Facebook applications are one of the reasons for Facebook attractiveness. Unfortunately, numeroususers are not aware of the fact that many malicious Facebook applications exist. Hackers have realized the potential of using apps for spreading malware and spam. The problem is already significant, as in our dataset, we find that at least 20% of apps are malicious. So far, the research community has focused on detecting malicious posts. In this paper, we ask the question: Given a Facebook application, can we determine if it is malicious or not? Our key contribution is in developing FRAppE—Facebook's Rigorous Application Evaluator-the first tool focused on detecting malicious apps on Facebook. To develop FRAppE, we use information gathered by observing the posting behavior of 989 million Facebook apps are seen across 1.86 billion users on Facebook. First, we identify a set of features that help us distinguish malicious apps from benign ones. Second, leveraging these distinguishing features, we show that FRAppE can detect malicious apps with 99.5% accuracy, with no false positives and a high true positive rate (4.1%). Finally, we explore the ecosystem of malicious Facebook apps and identify mechanisms that these apps use to propagate. Long term, we see FRAppE as a step toward creating an independent watchdog for app assessment and ranking, so as to warn Facebook users before installing apps.*

Key Words: **Facebook apps, malicious, online social networks, spams,verification.**

1.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.

2. BACKGROUND

MYPAGEKEEPER

MyPageKeeper is created for protecting personal profile on the web by helping them to identify malicious websites. MyPageKeeper scans and monitors all content posted on your facebook wall and news feed. It uses advanced techniques to identify whether a piece of content is malicious, spam or related to phishing. Continous monitoring and the latest web based malware detection technologies allows MyPageKeeper to protect your online persona on Facebook.

MyPageKeeper also scans our social networking pages periodically to make sure that there is no malicious content, eg., malicious links or advertisements, on your Facebook walls and Twitter timelines. In addition to protecting us from visiting malicious websites accidentally, we also protect our friends who follow our updates.

Researchers from University of California , Riverside and security experts from StockTheHacker.com have joined forces to provide this as a service to the community.
The moment a Fb user installations My-PageKeeper, that routinely crawls content on the user's retaining wall along with reports given. MyPageKeeper blacklists websites, in addition to custom classification techniques to determine malevolent content. MyPageKeeper finds malevolent content along with high accuracy—97% associated with content flagged because of it indeed point to help malevolent sites also, it incorrectly flags just 0.005% associated with cancerous content. The key thing to note here's which MyPageKeeper determines cultural spyware and adware for the granularity associated with specific content, without having group together content of almost any given software. MyPage- Keeper aren't published simply by almost any software; a lot of content are made physically by way of user as well as published using a cultural plugin (e.g., by way of user simply clicking 'Like' as well as 'Share' when using outside website).

3. PROPOSED SOLUTION

3.1 Existing System

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.

Gao et al. 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.

Yang et al. and *Benevenuto et al.* developed techniques to identify accounts of spammers on Twitter. Others have proposed a honey-pot-based approach to detect spam accounts on OSNs.

Yardi et al. analyzed behavioral patterns among spam accounts in Twitter.

Chia et al. investigate risk signaling on the privacy intrusiveness of Facebook apps and conclude that current forms of community ratings are not reliable indicators of the privacy risks associated with an app.

Disadvantages:

1) Existing system works concentrated only on classifying individual URLs or posts as spam, but not focused on identifying malicious applications that are the main source of spam on Facebook.

2) Existing system provided only a high-level overview about threats to the Facebook graph and do not provide any analysis of the system.

3) Existing system works focused on accounts created by spammers instead of malicious application.

4) The app can obtain users' personal information such as email address, home town, and gender.The app can "re-produce" by making other malicious apps popular.

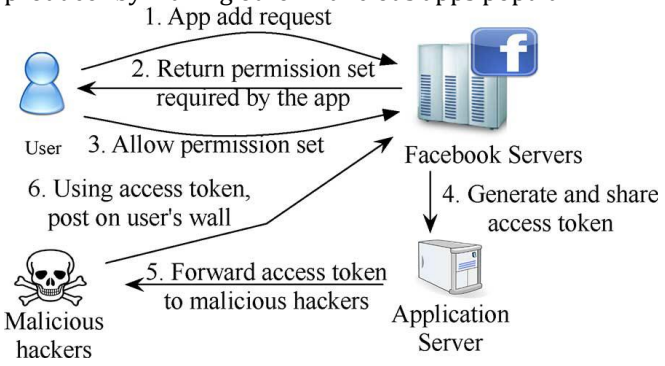

Fig 1: Architecture of Existing System.

When the users click the link which leads to the malicious app installation page, they redirect users to different pages for collecting victims personal information and make her complete surveys so that they can earn money. Once the app is installed, hackers get permission to post anytime on the victims wall. So, they make the same post and appears victims friends news feed and thus the cycle repeats and the app spreads in Facebook as shown in figure 2.

Figure 2: Cycle how malicious app works on Facebook.

3.2 Proposed System

These problems are overcomes in the proposed system. In the proposed system , we develop FRAppE tool, to detect and block the malicious applications in the Face book. When user is trying to post the offensive words or posts to the user's Face book wall, those words or posts are detected using the dictionary and it gets filtered. When we found any installation of the malicious app, user wall gives a warning notification that the app found is malicious, whether to install it or not. Offensive words or posts which are not related are detected and blocked using the FRAppE tool. These words or posts will not display in the public wall. Instead of that such post will be migrated to the blocked post list. User can view those things secretly and also a warning mail is send to the user. It is safe and secure. Unnecessary information will not be added in our wall.

FRAppE, a tool stands for Face book's Rigorous Application Evaluator which is helpful in monitoring the entire system. In Authentication and Authorization module, the user will register the data and login into the pages to view their profile to see all the contacts, the user will do all the works here. They can easily access the data from the database. If any malicious app is found in the profile, it will be detected using FRAppE in warn malicious app module and after detecting it will send the warning notification message. If there is any post of offensive words or posts in the user wall, those offensive words will be detected and blocked using the dictionary and these overall details will be stored in the database. The next work to be done by the database is to send a warning mail to the user. The blocked words will be send to the private wall and it can be viewed in blocked post list by the user alone.

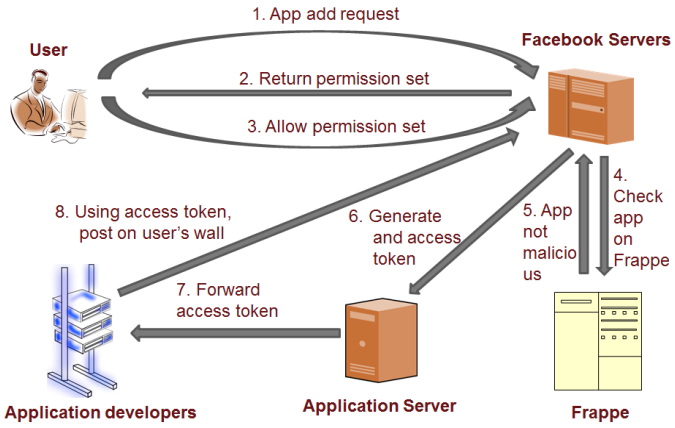

Fig 3 : Proposed system to detect malicious apps

Advantages:

1) The proposed work 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.
2) Several features used by FRAppE, such as the reputation of redirect URIs, the number of required permissions, and the use of different client IDs in app installation URLs, are robust to the evolution of hackers.
3) Not using different client IDs in app installation URLs would limit the ability of hackers to instrument their applications to propagate each other.
4) We 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 applications.
5) Background on app cross promotion among apps.
6) App collaboration to identify the major hacker groups.
7) Investigate hosting domains that enables redirection websites.

4. DETECTING MALICIOUS APPS

4.1 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 that can evaluate any Facebook application at the time when a user is considering installing it to her profile.

4.2 FRAppE

Next, we consider FRAppE—a malicious app detector that utilizes our aggregation-based features in addition to the on-demand features. 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.

5. SYSTEM ARCHITECTURE

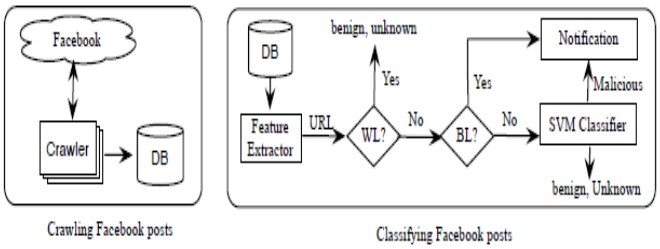

Fig 4 : System architecture showing crawling and classifying Facebook posts

The key novelty of MyPageKeeper lies in the classification module (summarized in Fig 4). As described earlier, the input to the classification module is a URL and the related social context features extracted from the posts that contain the URL. Our classification algorithm operates in two phases, with the expectation that URLs and related posts that make it through either phase without a match are likely benign and are treated as such.

Using whitelists and blacklists: To improve the efficiency and accuracy of our classifier, we use lists of URLs and domains in the following two steps. First, MyPageKeeper matches every URL against a whitelist of popular reputable domains. We currently use a whitelist comprising the top 70 domains listed by Quantcast, excluding domains that host user-contributed content (e.g., OSNs and blogging sites). Any URL that matches this whitelist is deemed safe, and it is not processed further. Second, all the URLs that remain are then matched with several URL blacklists that list domains and URLs that have been identified as responsible for spam, phishing, or malware. Again, the need to minimize classification latency forces us to only use blacklists that we can download and match against locally. Such blacklists include those from Google's Safe Browsing API, Malware Patrol , PhishTank , APWG , SpamCop , joewe in , and Escrow Fraud. Querying blacklists that are hosted externally, such as SURBL, URIBL and WOT, will introduce significant latency and increase MyPageKeeper's latency in detecting socware, thus inflating the window of vulnerability. Any URL that

matches any of the blacklists that we use is classified as socware.

Using machine learning with social context features: All URLs that do not match the whitelist or any of the blacklists are evaluated using a Support Vector Machines (SVM) based classifier. SVM is widely and successfully used for binary classification in security and other disciplines. We train our system with a batch of manually labeled data, that we gathered over several months prior to the launch of MyPageKeeper. For every input URL and post, the classifier outputs a binary decision to indicate whether it is malicious or not. Our SVM classifier uses the following features.

6. IMPLEMENTATION

6.1 Data Collection

The data collection component has two subcomponents: the collection of facebook apps with URLs and crawling for URL redirections. Whenever this component obtains a facebook app with a URL, it executes a crawling thread that follows all redirections of the URL and looks up the corresponding IP addresses. The crawling thread appends these retrieved URL and IP chains to the tweet information and pushes it into a queue. As we have seen, our crawler cannot reach malicious landing URLs when they use conditional redirections to evade crawlers. However, because our detection system does not rely on the features of landing URLs, it works independently of such crawler evasions.

6.2 Feature Extraction

The feature extraction component has three subcomponents: grouping of identical domains, finding entry point URLs, and extracting feature vectors. To classify a post, MyPageKeeper evaluates every embedded URL in the post. Our key novelty lies in considering only the social context (e.g., the text message in the post, and the number of Likes on it) for the classification of the URL and the related post. Furthermore, we use the fact that we are observing more than one user, which can help us detect an epidemic spread.

It detects Presence of Spam keywords like 'FREE', 'DEAL' and 'HURRY'.

6.3 Training

The training component has two subcomponents: retrieval of account statuses and training of the classifier. Because we use an offline supervised learning algorithm, the feature vectors for training are relatively older than feature vectors for classification. To label the training vectors, we use the account status; URLs from suspended accounts are considered malicious whereas URLs from active accounts are considered benign. We periodically update our classifier using labeled training vectors.

6.4 Classification

The classification component executes our classifier using input feature vectors to classify suspicious URLs. When the classifier returns a number of malicious feature vectors, this component flags the corresponding URLs information as suspicious.

The classification module uses a Machine Learning classifier based on Support Vector Machines, but also utilizes several local and external white lists and blacklists that help speed up the process and increase the over-all accuracy. The classification module receives a URL and the related social context features extracted in the previous step.

These URLs, detected as suspicious, will be delivered to security experts or more sophisticated dynamic analysis environments for an in-depth investigation.

6.5 Detecting Suspicious

The Detecting Suspicious and notification module notifies all users who have social malware posts in their wall or news feed. The user can currently specify the notification mechanism, which can be a combination of emailing the user or posting a comment on the suspect posts.

7. 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 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 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.

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the paper on "FRAPPE – Facebook Rigorous Application Evaluator". We would like to take this opportunity to thank my internal guide **Prof. Vishwanatha, Assistant Professor,** for giving us all the help and guidance needed. We are really grateful to him for his kind support. His valuable suggestions were very helpful. We are also grateful to **Dr. K. S. Jagadeeshgowda, Head of The Department** of **Computer Science and**

Engineering, for his indispensable support, suggestions. In the end our special thanks to our principal **Dr. Manjunatha. A**, for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Paper.

REFERENCES

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

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

[3] 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.

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

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

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

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

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

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

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

[11] S. Lee and J. Kim. Warningbird: Detecting suspicious urls in twitter stream. In NDSS, 2012.

[12] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove.Analyzing facebook privacy settings: user expectations vs. reality. In IMC, 2011.

[13] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious urls. In KDD, 2009.

[14] Makridakis, E. Athanasopoulos, S. Antonatos, D. Antoniades, S. Ioannidis, and E. P. Markatos.Understanding the behavior of malicious applications in social networks. Netwrk. Mag. of Global Internetwkg., 2010.

[15] 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.

[16] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In Proceedings of the 4th Workshop on Social Network Systems, 2011.

[17] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In ACSAC, 2010.

[18] 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.

[19] N. Wang, H. Xu, and J. Grossklags. Third-party apps on facebook: privacy and the illusion of control. In CHIMIT, 2011.

[20] Yang, R. Harkreader, and G. Gu. Die free or live hard? empirical evaluation and new design for fighting evolvingntwitter spammers. In RAID, 2011.