International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

Facebook's Rigorous Application Evaluator for Malicious Applications

[1]**Srinu.B** (guide)-srinub1307@gmail.com
[2]**Prathibha.N**-prathibhanallamasa@gmail.com
[3]**Kiran Sai Kumar.A**-kiran1990sai@gmail.com
[4]**Ayesha Samreen**-asamreen7866@gmail.com
[1, 2, 3,4]Dept.of Computer Science and Engineering, Vignan Institute of Technology and Science, Deshmukhi, Hyderabad. 508284

Abstract:

With 20 million installs a day, third-party apps are a major reason for the popularity and addictiveness of Facebook. Unfortunately, hackers have realized the potential of using apps for spreading malware and spam. 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 paper, we ask the question: Given a Facebook application, can we determine if it is malicious? Our key contribution is in developing Facebook'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 Facebook apps seen across 2.2 million users on Facebook. First, we identify a set of features that help us distinguish malicious apps from benign ones. For example, we find that malicious apps often share names with other apps, and they typically request fewer permissions 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 high true positive rate (95.9%). Finally, we explore the ecosystem of malicious Facebook 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 1584 apps enabling the viral propagation of 3723 other apps through their posts. 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...

Keywords

Network, Spam, Social Media, behavioral, user-linguistic...

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 [2] that facilitates app integration into the Facebook User Experience. There are 500K apps available on Facebook [3], and on average, 20M apps are installed every day [1]. 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 [4]–[6]. Malicious apps can provide a lucrative business for hackers, given the popularity of OSNs, with Facebook leading the way with 900M active users [7]. 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 [8]. In other words, there is motive and opportunity, and as a result, there are many malicious apps spreading on Facebook every day [9].

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 [10]–[12]. At the same time, in a seemingly backwards step, Facebook has dismantled its app rating functionality recently. A recent work studies how app permissions and community ratings correlate to privacy risks of Facebook apps [13]. Finally, there are some community- based feedback-driven efforts to rank applications, such as WhatsApp? [14]; though these could be very powerful in the future, so far they have received little adoption.

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 [15] 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.

II. Literature review

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.

III. System Design and Architecture

A. *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 Of Existing System:

- 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.
- Existing system works focused on accounts created by spammers instead of malicious application.
- Existing system provided only a high-level overview about threats to the Facebook graph and do not provide any analysis of the system.

Proposed System:

- 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.
- We find that malicious applications significantly differ from benign applications with respect to two classes of features: On-Demand Features and Aggregation-Based Features.
- We present two variants of our malicious app classifier— FRAppE Lite and FRAppE.
- 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.
- FRAppE—a malicious app detector that utilizes our aggregation-based features in addition to the on-demand features.

Advantages Of Proposed System:

- 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.
- 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.

IV. Design and Implementation

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

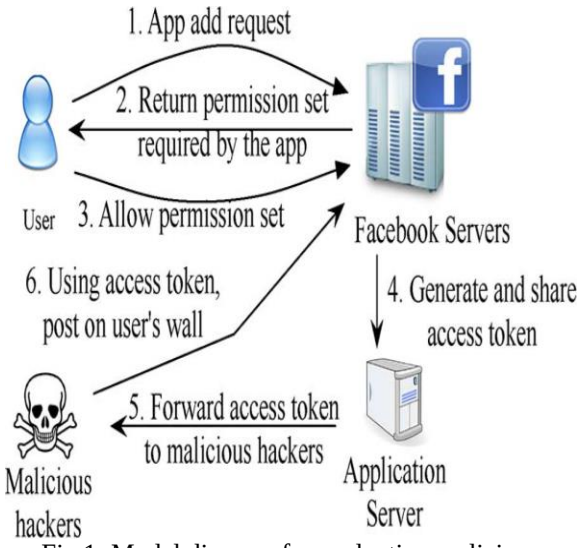

Fig 1: Model diagram for evaluating malicious applications

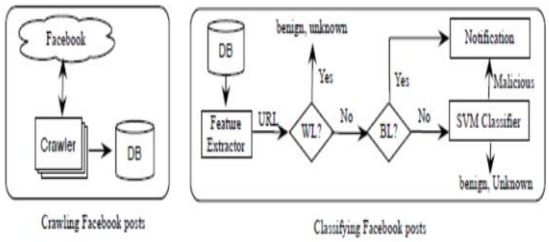

Fig.2 Architecture of FRAppE

A. *Modules:*

- Data collection
- Feature extraction
- Training
- Classification
- Detecting Suspicious

B. *Modules Description:*

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.

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’.

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.

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.

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

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

V. Results

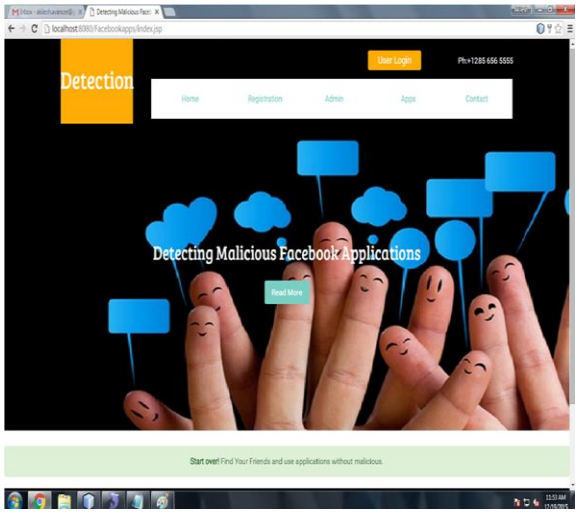

Fig 3: Home page for web application

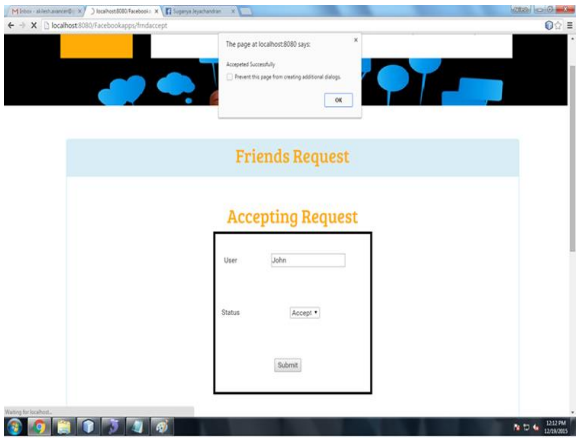

Fig 4: Accepting a friend request

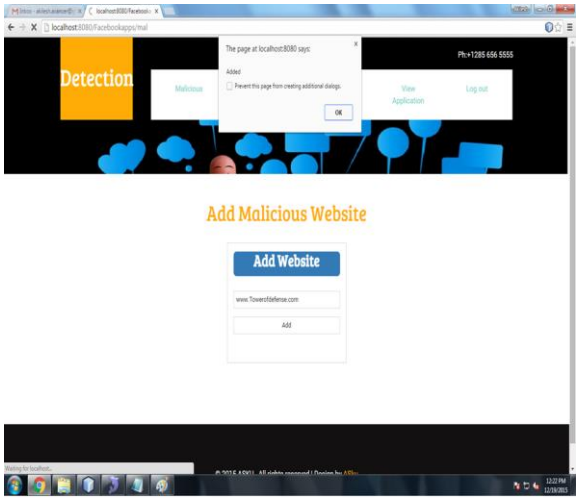

Fig 5: Adding a malicious website

VI. 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.

VII. References

[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.