Electronic Communications of the EASST
Volume X (2009)

Proceedings of the
Third International Workshop on
Formal Methods for Interactive Systems
(FMIS 2009)

Poporo: A Formal Framework for Social Networking

Nestor Catano, Vassilis Kostakos, Ian Oakley

4 pages

Guest Editors: Michael Harrison, Mieke Massink
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Poporo: A Formal Framework for Social Networking

Nestor Catano[1], **Vassilis Kostakos**[1], **Ian Oakley**[1]

[1] (ncatano,vk,ian)@uma.pt
Madeira-ITI, Portugal

Abstract: This position paper presents a novel approach for ensuring privacy in online social network applications through the combination of formal methods so as to reason in logic about privacy policies, graph theory and simulation to establish the potential threats of revealing information to users, and Human Computer Interaction to ensure that policies are usable and configurable.

Keywords: Social Networking, Security and Privacy Policies, Formal Methods, Refinement Calculus, Proof Carrying Code

1 Introduction

In recent years, online social network services in the form of websites such as Facebook, MySpace, LinkedIn and Hi5 have become popular tools to allow users to publish content, share common interests and keep up with their friends, family and business connections. A typical social network user profile features personal information (e.g. gender, birthday, family situation), a continuous stream of activity logged from actions taken on the site (such as messages sent, status updated, games played) and media content (e.g. personal photos and videos). The privacy and security of this information is a significant concern [GA05]. For example, users may upload media they wish to share with specific friends, but do not wish to be widely distributed to their network as a whole. Control of the access to the content on social network profiles is therefore an important issue. However, numerous tensions exist. For example, users find stipulating detailed privacy settings to be challenging and often fail to achieve their goals [BAC09]. Furthermore, social network services have conflicting goals. Although respecting the privacy of their client base is important, they must also grow and expand the connections between their users in order to be successful. This is typically achieved by exposing content to users through links such as "friends-of-friends", in which content relating to individuals known to a user's friends (but not the user) is revealed. Examples of this behaviour include gaining access to a photo album of an unknown user simply because a friend is tagged in one of the images.

This position paper argues that users need mechanisms to reliably restrict access to content in online social network services and suggests that formal methods [RV01] can provide a logical foundation with which to achieve this goal: to express and enforce privacy and security policies unambiguously. It outlines a vision in which social networking websites are used as a living test-bed for novel systems which combine formal methods, graph theory and Human-Computer Interaction (HCI) techniques to develop privacy and security systems which are secure, dependable, trustworthy and usable. These areas are discussed in the reminder of this paper.

2 Proposed Approach

The main components of Poporo are summarised as follows. We first construct Matelas[1], a predicate calculus abstract specification layer definition for social networking, modelling social-network content, privacy policies, social-networks friendship relations, and how these effect the policies with regards to content and other users in the network. Using refinement calculus techniques [HHS86], Matelas is ported into a social network core application that adheres to the stipulated policies. Using Proof Carrying Code (PCC) [Nec97], the functionality of this core is then extended by the development of plug-ins that adhere to the policies. These plug-ins are then automatically categorised in terms of their threat to privacy by analysing the APIs and graph algorithms utilised by each plug-in. Finally, HCI techniques are used to develop interfaces that effectively represent the policies of the core application to users, as well as enabling them to modify and adapt them to suit their preferences.

2.1 Matelas and Predicate Calculus

The basis of our work is Matelas, a specification layer that builds on predicate calculus and focuses on human centred privacy and security policies. Using refinement calculus techniques [HHS86], Matelas is used to construct a sound social network core application that adheres to stipulated policies. That is, from a predicate calculus specification of social networks, a code level specification model is attained while applying successive refinement steps.

We will deliver a social network core application that verifies and implements social network privacy policies considered in Matelas. The core application will serve as a common layer to which social network functionalities will be plugged-in. Matelas will distinguish four rather independent aspects of social networks, namely, user content and privacy issues, user content and how it is affected by friendship relations, the user interface, and the user content and its hierarchy.

2.2 Using Proof Carrying Code to Extend the Core Application

While the social network core application described in Section 2.1 is minimal in functionality, it will be considerably extended by incorporating plug-ins . This can be achieved by developing a framework where the plug-ins, written in popular programming languages such as Java or C, can demonstrate their adherence to the policies stipulated by Matelas. This will be achieved by using PCC [Nec97], which is a technique in which a code consumer (the social network core application) establishes a set of rules (privacy and security policies) that guarantee that externally produced programs (the plug-ins) can safely be run by the consumer. In addition to the code to be executed, the code producer must provide a proof of adherence to set of rules defined by the code consumer. This proof is run by the code consumer once, using a proof validator, to check whether the proof is valid and therefore the external program is safe to execute. It is imperative that the proof validator is automatic and fast. Hence, Matelas, while expressive enough for modelling general privacy and policy properties, must allow the (semi-) automatic checking of proofs.

[1] Matelas is the French word for the English word mattress.

The policies for Java plug-ins can be written in JML [LBR06] (Java Modeling Language), which allows different formal methods tools to check program correctness [BCHJ05, BCC$^+$05]. JML specs have the advantage over predicate calculus based models in that they are close to Java, and thus are closer to average programmers. We therefore envisage to investigate on systematic ways JML specs can be translated into predicate calculus based models.

In summary, the main output of this work is a PCC based plug-in validator that checks plug-ins for compliance with Matelas defined policies, and a translation definition from Matelas to JML.

2.3 Formal Analysis of Social Networking Privacy

An important step in understanding the privacy threats of plug-ins is a rigorous assessment of the privacy threats posed by the various graph theory algorithms commonly used in social networking systems and services. For example, consider a plug-in that suggests new friends to a user, based on the user's existing friends and the relationships between those friends. One way to achieve this is to use a local clustering algorithm (e.g. transitivity or clustering coefficient). If such a plug-in has access to all of a user's friends, then it can statistically calculate the user's age by looking at his/her friends' age, location (similarly by looking at the locations of the user's friends), gender, work, etc. A first step in mapping the privacy threats of various graph algorithms provided by the social network core application is to run extensive simulations. Crucially, the underlying network structure may be important in determining the severity of the privacy threat. Therefore, a number of artificial social networks will be generated using a variety of parameters (such as size, density, degree distribution, average path length). These networks will be then populated with private and public information, such that they resemble a real-world social network. Our simulation environment will then use Monte-Carlo simulation of a number of graph algorithms (such as shortest path, calculation of betweenness, statistic features), to explore their privacy implications. Our formal specification engine will be used during these simulations to ensure that the algorithms adhere to privacy policies.

We will deliver a classification of graph theory algorithms based on the threat to privacy. This classification can be used to automatically assign a simple "Privacy" label (e.g. Green, Yellow, Red) to 3rd party software or plug-ins that will be used by our formal specification engine. This label is intended to communicate to users the threat to their privacy, in a simple and understandable fashion, much in the spirit of Section 2.4.

2.4 Human Considerations

While a social network application or plug-in may provably adhere to policies, these policies are typically sufficiently abstract to allow for human error. While a plug-in may not access users date of birth without explicit authorisation, it is still possible for users to inadvertently give such authorisation. This may happen either by accident or, most likely, due to the complexity of the settings and preferences interface that the user is asked to interact with. Hence it is imperative to augment the provably correct social network core and plug-ins with understandable human interfaces that enable end users to express their privacy policies and preferences, as well as to review and modify them. This can be achieved by a number of approaches. First, providing clear and understandable labels and metaphors that effectively present policies. Second, enabling users to interact with their policies, obtaining feed-forward about the potential effects of any changes they make to their policies. To this end, we will use iterative design and testing of initial sketches and prototypes. In addition, we will perform heuristic analyses and cognitive walk-throughs to identify potential problems with our human interfaces. Finally, we will carry out user observations to measure users' reaction to our designs as well as their subjective preferences.

We will deliver a set of prototype designs that will be integrated into our social networking system (possibly developed as a plug-in). These designs will be responsible for acting as a bridge between end users and our formal specification engine.

3 Conclusion

This position paper has presented a novel vision of maintaining privacy and securing data in online social network services through the combination of formal methods (to provide provable behaviours), simulation and graph theory (to provide meaningful generalisations of algorithmic specified activities) and HCI (to ensure that systems are usable and allow individuals to quickly and effectively configure them). We believe that this approach will lead to the development of new back-end systems, front-end prototypes and theoretical understanding relevant to the complex issues underlying the security and privacy of data stored in online social network services.

Bibliography

[BAC09] J. Bonneau, J. Anderson, L. Church. Privacy Suites: Shared Privacy for Social Networks. In *Symposium on Usable Privacy and Security (SOUPS)*. July 2009.

[BCC$^+$05] L. Burdy, Y. Cheon, D. Cok, M. Ernst, J. Kiniry, G. T. Leavens, K. R. M. Leino, E. Poll. An Overview of JML Tools and Applications. *International Journal on Software Tools for Technology Transfer (STTT)* 7(3):212–232, June 2005.

[BCHJ05] C. Breunesse, N. Catano, M. Huisman, B. Jacobs. Formal Methods for Smart Cards: An Experience Report. *Science of Computer Programming* 55(1-3):53–80, March 2005.

[GA05] R. Gross, A. Acquisti. Information Revelation and Privacy in Online Social Networks. In *Workshop on Privacy in the Electronic Society (WPES)*. Pp. 71–80. 2005.

[HHS86] J. He, C. A. R. Hoare, J. W. Sanders. Data Refinement Refined. In *European Symposium on Programming (ESOP)*. Pp. 187–196. 1986.

[LBR06] G. Leavens, A. Baker, C. Ruby. Preliminary Design of JML: A Behavioral Interface Specification Language for Java. *ACM SIGSOFT Software Engineering Notes* 31(3):1–38, 2006.

[Nec97] G. C. Necula. Proof-Carrying Code. In *Symposium on Principles of Programming Languages (POPL)*. P. 106119. Paris, January 1997.

[RV01] A. Robinson, A. Voronkov. *Handbook of Automated Reasoning*. MIT Press, 2001.