Accepted Manuscript

Detection of app collusion potential using logic programming

Jorge Blascoa, Thomas M. Chen, Igor Muttik, Markus Roggenbach

PII: S1084-8045(17)30420-4

DOI: 10.1016/j.jnca.2017.12.008

Reference: YJNCA 2028

To appear in: *Journal of Network and Computer Applications*

Received Date: 28 August 2017

Revised Date: 16 November 2017

Accepted Date: 17 December 2017

Please cite this article as: Blascoa, J., Chen, T.M., Muttik, I., Roggenbach, M., Detection of app collusion potential using logic programming, *Journal of Network and Computer Applications* (2018), doi: 10.1016/j.jnca.2017.12.008.

Detection of App Collusion Potential Using Logic Programming

Jorge Blasco[a], Thomas M. Chen[b], Igor Muttik[c], Markus Roggenbach[d]

[a] *Jorge.BlascoAlis@rhul.ac.uk, Information Security Group. Royal Holloway, University of London*
[b] *Tom.Chen.1@city.ac.uk, Electronic and Electrical Engineering Department. City, University of London*
[c] *igor.muttik@cybercurio.com, Cyber Curio*
[d] *M.Roggenbach@swansea.ac.uk, Department of Computer Science, Swansea University*

Abstract

Mobile devices pose a particular security risk because they hold personal details (accounts, locations, contacts, photos) and have capabilities potentially exploitable for eavesdropping (cameras/microphone, wireless connections). The Android operating system is designed with a number of built-in security features such as application sandboxing and permission-based access control. Unfortunately, these restrictions can be bypassed, without the user noticing, by colluding apps whose combined permissions allow them to carry out attacks that neither app is able to execute by itself. While the possibility of app collusion was first warned in 2011, it has been unclear if collusion is used by malware in the wild due to a lack of suitable detection methods and tools. This paper describes how we found the first collusion in the wild. We also present a strategy for detecting collusions and its implementation in Prolog that allowed us to make this discovery. Our detection strategy is grounded in concise definitions of collusion and the concept of ASR (Access-Send-Receive) signatures. The methodology is supported by statistical evidence. Our approach scales and is applicable to inclusion into professional malware detection systems: we applied it to a set of more than 50,000 apps collected in the wild. Code samples of our tool as well as the detected malware are available.

Keywords: Android, Collusion, Malware, MoPlus