Runtime Verification For Android Security

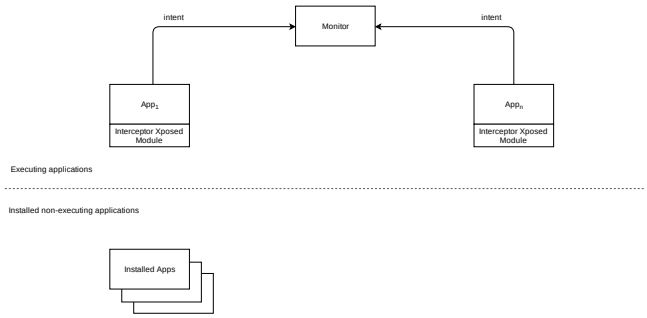
Richard Allen[1] and Markus Roggenbach[1][1111−2222−3333−4444]

Swansea University lncs@springer.com
http://www.springer.com/gp/computer-science/lncs

The Android operating system has found many applications in devices such as phones, tablets, watches and smart TVs. These are all applications of computing systems where the primary use of the system is not computation. However they have a computing system as an integral component. Furthermore they are open systems, i.e., they can communicate with other external computing systems. As such, these systems are vulnerable to attacks such as information theft, service abuse, ransomwear. In the words of the McAfee 2020 Threat Report: "Mobile Malware Is Playing Hide and Steal." It is not known to the user what data each app collects, what apps are doing with the data, or even who they share it with.

In this context we use techniques from runtime verification in order to alert the user of an Android device if there is a potential security breach. The advantages of this approach are:

- The user is informed immediately when there device is under attack.
- It is independent of the application code.
- It is extendable to further security properties.



Our system architecture (see above) makes use of the Xposed framework [**?**] in order to intercept calls to security sensitive Android O/S functions. It adds a monitor app to the user's device which logs these calls and, at arrival of a new event, checks for each security property of interest if it is fulfilled or not. The security properties are encoded in linear temporal logic (LTL). In a first attempt we utilised an algorithm by Rosu and Havelund [**?**] which appeared to be interesting thanks to it's low complexity: for the cost of one evaluation of a property φ is $O(|t| * |\varphi|)$. However upon investigation we discovered the dynamic complexity to be $O(|t|^2 * |\varphi|)$. This revealed the algorithm to be unsuitable for our requirements. We made changes that resulted in a new algorithm with a dynamic complexity of $O(|\varphi|)$.