Usable Logging as a Security Response to Physical Attacks on Mobile Devices

José Franco, Ana Cristina Pires, Luís Carriço, Tiago Guerreiro
LASIGE, Faculdade de Ciências, Universidade de Lisboa
fc44914@alunos.fc.ul.pt,acdpires@fc.ul.pt,lmc@di.fc.ul.pt,tjvg@di.fc.ul.pt

ABSTRACT

Users are susceptible to privacy breaches when people close to them gain physical access to their phones. We present logging as a security response to this threat, one that is able to accommodate for the particularities of social relationships. To this end, and explore the feasibility of the logging approach, we present a prototype developed for Android that continuously gathers user interactions and translates them into human-readable units. Our future work will focus on understanding the amount and richness of information required for users to distinguish intrusions from ordinary usage.

1 INTRODUCTION

Currently, users are susceptible to the threat of physical intrusion to their personal mobile devices, perpetrated by people known to them. This threat is neither remote nor improbable[4, 5].

To protect against physical intrusion, mobile devices typically employ authentication, both at the start of operation, and, in some cases, to access specific functionality. In practice, however, authentication is often not adequate to user needs, as is revealed by its underwhelming adoption [1, 3]. Additionally, with the most popular authentication methods, adversaries in close proximity can easily observe the owner entering the code [8]. Finally, the security of unlock authentication is dependent on the owner not losing control over the device within a session, an assumption that is increasingly not reasonable in the case of known non-owners [2, 4].

For end-users the choice then seems to be between either to absolutely trust, or to absolutely distrust, people they know – a proposition that falls in stark contrast with people's desire for granular regulation of privacy [4, 6]. This research aims to bridge that gap, by offering security capabilities that accommodate the social relationship between self and others. In this paper, we explore feasibility of providing security logging capabilities in a way that is suitable for end-users. To do so, we developed an Android prototype that: 1) gathers user interactions and screen views, without breaking the "app model"; and, 2) parses the raw stream of usage data into human-readable units of information (e.g., Scrolled through Messages for 1 minute).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Woodstock '18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

2 BUILDING A USABLE ACTIVITY LOGGER

To explore the feasibility of the activity logging approach, we developed a tool for Android, designed to be accessible to end-users.

2.1 Concept

This approach is grounded on security technologies in non-consumer computing assets. Large computer networks have for long maintained intrusion prevention systems (IPS) as a second-line capability, aimed at detecting security threats that have found a way into the organizational perimeter despite primary security barriers, and mitigate them, either through counter-actions, or by gathering and keeping information that can help in incident recovery [7].

Activity logging is a type of response that host-based IPS can provide to security threats. For the purposes of this research, we limit the concept to a service that runs on the background, as to not interfere with regular operation, and collects time-stamped information about which applications were interacted with, what actions were performed in those applications. Such security mechanism can run concurrently to other deterrents to physical intrusion, like unlock authentication, if users so choose.

Unlike the typical IPS, a major design consideration for a personal activity logger is the ease of operation. In an organization, when a security incident occurs, specialists conduct analysis of security logs using complex tools and techniques. For end-users, the logs must be made available in usable fashion. The amount and granularity of information collected is not usable as a way to distinguish behaviors. As such, we parse this data, user actions and screen contents, into human-readable actions that are likely to be informative for future logging interfaces.

2.2 Implementation

We developed a service that runs in the background, gathering activity logs and abstracting them to higher level descriptions. This logging prototype is then divided into two major components: data gathering and parsing.

2.2.1 Logging. The data gathering component is responsible for listening to accessibility events, using the Android Accessibility Services API, occurring in a user smartphone. Using the Accessibility Services API in security applications is not unheard of. Many password managers use this functionality to access fields inside running apps. Every time an event (e.g., a focus, a scroll) is captured, the respective view tree is also gathered and associated to the event, depending if the view tree is gathered within a valid period (we call this association an action). It is through the collection of these actions, within the context of an application, that a sequence is generated. This data is then fed to the parsing system for processing.

2.2.2 Generating Human-Readable Logs. The parsing system is constantly listening to newly generated sequences and is responsible for transforming the raw data stream into its respective human-readable log. First, the system transforms the gathered data into two respective domain-specific languages (DSL): an event log; and their respective view tree content log. As an example, the event log DSL is represented in the system as follows:

```
STARTSEQ(package, seq_start_time):
  SCROLL(time, view_id, view_class, ...);
  SCROLL(time, view_id, view_class, ...);
  FOCUS(time, view_id, view_class, ...);
ENDSEQ
```

Second, these DSLs are fed into a sequence of different grammars, processed by ANTLR4 [1] technology. Two branches of parsing are created: the event log DSL parsing; and the content view log DSL parsing.

The event log parsing consists on finding higher-level patterns on the event sequence. For example, the system can group redundant events (such as scrolling multiple times in the same period), find navigation window changes (by looking for a specific pattern of FOCUS, CLICK and SCROLL events) and find text input. One of the intermediary event log DSL structure is represented internally as follows:

```
STARTSEQ(package, seq_start_time):
  SCROLLED(start_time, end_time);
  CHANGED_TAB(time, window_name);
  TYPED(time, text);
ENDSEQ
```

Each application may have its own content grammar, and each grammar is responsible for matching context-sensitive information, or may be subject to a generic grammar. For example, in Messenger, the system tries to find possible conversation messages (by looking for a specific pattern within the content log) and specific window contents (such as the conversation name on top of a message). Every intermediary DSL generated becomes smaller in size, but more descriptive overall. Thirdly, after every parsing branch is concluded, the system matches the final event log DSL with the respective final content log DSL and generates a final merged DSL, which can be represented as follows:

```
SEQUENCE(package, seq_start_time, duration):
  CHANGED_TAB(time, tab_name);
  SCROLLED(time, content_type, duration);
  OPEN_CONV(time, conv_name);
  TYPED(time, text);
ENDSEQ
```

At the end of these processes we provide a continuous description of human-readable and meaningful user actions, from which the following are examples: *opened Messenger, scrolled through the conversations for 7 seconds, opened conversation with Tiago, browsed through text messages for 4 seconds, typed a message; opened Photos, scrolled through photos for 13 seconds; opened Gmail, scrolled through 15 mails, opened email from Tiago Guerreiro.*

[1] http://www.antlr.org/

3 OUTLOOK

To deal with both the use case of unattended access and the one of social sharing abuse, while addressing users concerns over being perceived as overly protective, we are exploring activity logging as a security response. To test the technical feasibility of this concept, we built a logging tool with information gathering capabilities for Android. To make collected information usable, we abstracted low-level user and application information to human-readable meaningful bits of information. This prototype runs on mainstream Android devices without breaking their security model, by using the Accessibility services provided by the operating system.

Activity logging offers users the opportunity to know if the device was snooped on, without having to engage in any explicit action, such as changing users' accounts. It can also act as a deterrent: if potential snoops know that the device records activity, they may reasonably fear being detected. One additional positive aspect of the presented approach is that it runs locally with no sensitive information leaving the device perimeter.

The proposed approach does not come without challenges. One particular concern is that they could also afford kinds of practices which are themselves worrisome, like using logging to collect private information from others or using it as a "honeypot" to assess how others can be trusted. Another concern is the increasing value of the log itself as a target of snooping, which brings challenges on how access to this log should be secured.

Our future work will focus on designing and evaluating interfaces that accommodate the collected information and allow users to spot intrusions to their usage.

4 ACKNOWLEDGEMENTS

This work was supported by national funds through FCT project mIDR (AAC02/SAICT/-2017, project 30347, cofunded by COMPETE/FEDER/FNR) and funding to the LASIGE Research Unit, ref. UIDB/00408/2020 and ref. UIDP/00408/2020.

REFERENCES

[1] Serge Egelman, Sakshi Jain, Rebecca S Portnoff, Kerwell Liao, Sunny Consolvo, and David Wagner. 2014. Are you ready to lock?. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 750–761.

[2] Alina Hang, Emanuel Von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. 2012. Too much information!: user attitudes towards smartphone sharing. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*. ACM, 284–287.

[3] Marian Harbach, Emanuel Von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. 2014. It'sa hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *Symposium on usable privacy and security (SOUPS)*. 213–230.

[4] Diogo Marques, Tiago Guerreiro, Luís Carriço, Ivan Beschastnikh, and Konstantin Beznosov. 2019. Vulnerability & Blame: Making Sense of Unauthorized Access to Smartphones. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[5] Diogo Marques, Ildar Muslukhov, Tiago Guerreiro, Luís Carriço, and Konstantin Beznosov. 2016. Snooping on mobile phones: Prevalence and trends. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*.

[6] Leysia Palen and Paul Dourish. 2003. Unpacking privacy for a networked world. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 129–136.

[7] Karen Scarfone and Peter Mell. 2007. Guide to intrusion detection and prevention systems (idps). *NIST special publication* 800, 2007 (2007), 94.

[8] Emanuel Von Zezschwitz, Paul Dunphy, and Alexander De Luca. 2013. Patterns in the wild: a field study of the usability of pattern and pin-based authentication on mobile devices. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*. ACM, 261–270.