Oxidizing Robots: Advancements of Rust on Android

Daniel Hugenroth & Luis A. Saavedra
University of Cambridge
{*dh623,las91*}*@cl.cam.ac.uk*

Smartphones have become the primary computing devices for people worldwide. As such, they store both company secrets and our most intimate pictures and messages; they keep track of our whereabouts during our everyday lives; and they secure authentication as a second-factor; This trove of sensitive data and use-cases make them a high-value target for criminals and sophisticated state-level actors alike.

Targeted attacks, such as those performed by state-level adversaries, typically exploit software vulnerabilities. The most powerful ones use remote-code exploitation (RCE) of vulnerabilities that are not known or patched (0-days) and do not require any action from the user (Zero-Click). Historically, many such vulnerabilities have been found in media codecs and file format parsers written in C/C++. These components exist both in the operating system and in individual apps and their dependencies.

Over the past few years, Rust has gained popularity as a programming language that offers high performance comparable to existing native code, while simultaneously providing zero-overhead correctness guarantees that are verified at compile time. Large companies, including Google, have already announced plans to rewrite critical code paths using Rust. This trend lends credibility to Rust's potential as a candidate for reducing security and correctness bugs.

In our presentation we offer a snapshot of our ongoing research to measure and analyse the status-quo of Rust on Android. For this we examine both the Android Open-Source Project (AOSP) and apps distributed through the Android Play Store.

We use the commit history of the AOSP repository to capture the progress of integrating Rust into Android. Specifically, we examine where Rust is being deployed and whether there is a focus on components that have previously exposed vulnerabilities. In the second part of our presentation, we leverage an existing database of app files to investigate Rust's adoption in the wider ecosystem. We explore which apps are among the first to adopt Rust and whether these changes are part of the app's own code or imported through (transitive) dependencies.

Finally, we identify roadblocks that might impede more wide adoption and deployment of Rust on Android. We believe that the insights from the Android ecosystem provide valuable lessons for promoting safer languages on other mobile devices as well.