**What is CopperDroid? How to use it?**

CopperDroid is a web interface, where you can upload APK file which is then analysed and results of Dynamic analysis will be shown. It covers low-level analysis, which is OS specific, such as writing to a file and high-level analysis, that are Android specific, such as making a phone call or sending an SMS.

It can tell from where the malicious application will initiate its behaviour either from java, JNI, or native code execution.

The way CopperDroid is built, is that the Android system will be on top of CopperDroid’s emulator, which is built on top of QEMU, and that part communicate to the other part where all the analysis happen, is by **A GDB stub is an implementation of RSP, which enables the target machine to communicate with the host machine on which a remote GDB session with a client is established.** The analysis takes place on the other side, of CooperDroid’s and it distinguishes Binder analysis from system call tracking.

**Info about Android**

However, Android applications are written in java language, and then deployed to Android Packages archives, known as APK’s. Each APK contains multiple components that can be described as a program in whole, each one of the components is designed to accomplish a certain task, for example, user-interface actions.

Each android application run in a separate userspace process, which in fact is assigned a unique user and group ID. Thus, each application is in an isolated environment.

**A number of components can make up an application. In particular, Android defines activities, services, content providers, and broadcast receivers. Activities, services, and broadcast receivers are activated by intents, i.e., asynchronous messages exchanged between individual components to request an action. Activity and service intents specify actions to be performed. Conversely, broadcast receiver intents define the received event and are delivered to the interested broadcast receivers**

But since the components of an application can be treated as one, the components within an application can interact …

In Each APK file there must be a manifest included, which contains list of components, permissions, software and hardware features an application has and use. Interestingly, the applications manifest can reveal few information about whether the app is malicious or not.

CopperDroid use two mechanisms that helps in analysing applications, one way is by tracking system calls invocations. Unlike other systems, it focuses on intercepting transitions registered in the **cpsr** register, between supervisor and user modes. Thus, allow it to recover all system calls made whether it returned values or not (e.g., **exit**, **execve**).

Dissecting Inter-process communication (IPC) and remote procedure calls (RPC), is the second mechanism CopperDroid use. The Binder kernel driver, handles **ioctl** system calls, which allows two components of an application to communicate(lorenzo2paper^). By examining the communication that occurs over these channels, it can distinguish the characteristics of a malicious app. **A System Call-Centric Analysis and Stimulation Technique to Automatically Reconstruct Android Malware Behaviors**

**CopperDroid: Automatic Reconstruction of Android Malware Behaviors**

By combining both system call and, and its parameters, obtained from Binder unmarshalled data. CopperDroid is capable of reconstructing the behaviours accurately. Introducing a unmarshalling Oracle, that runs on top unmodified Android Emulator. It purpose is to deserialize Binder Communication, and make the IPC communication more readable.

**How accurate it is? Why is it used in the research paper?**

CopperDroid has been evaluated over 2900 samples, that are taken from public datasets, Android Malware Genome Project, Contagio and McAfee. It indicates that it is capable of behavioural reconstruction at a significant level. However, more than 60% of the analysed samples had approximately 25% additional behaviours, which prove the system’s capabilities.

To assess CopperDroid’s effectiveness, analysing all the samples without external stimulation, then applied it on the same malware dataset...

Fdffdaf direct quote.

Hgyhgf change later

Ghaghg to be completed.