# Android App Reverse Engineering 101

Learn to reverse engineer Android applications!

View the Project on GitHub maddiestone/AndroidAppRE

---

# 1. Introduction

Welcome to Android™ App Reverse Engineering 101! This workshop's goal is to give you the foundations to begin reverse engineering Android applications. While this workshop won't teach you the details of Android app development, Android malware analysis, Android vulnerability hunting, etc., I hope to give you all the necessary foundations through this workshop such that you can apply your new Android reversing skills to doing those things.

This workshop will be wholly based on reverse engineering through static analysis, or analyzing and understanding an application by examining its code. I won't be covering dynamic analysis where an analyst runs an application and understands the application by executing it, debugging it, etc. Why? Static analysis tends to be a less approachable skill for people to pick up on their own, so I want to help you do it! (And I really love static analysis)

## Environment

All of the exercises in this workshop can be done in the virtual machine (VM) that is available here.

- Virtual Box VM
- .ova for VMWare

The VM is an Ubuntu 18.04 64-bit image. The username is `AndroidAppRE` and the password is `android`. The VM has the following tools installed:

- jadx - Android decompiler. We load the APKs into jadx. And are able to analyze the DEX bytecode using jadx.

- Ghidra - Software reverse engineering tool. We use its ARM disassembler/decompiler functionality in the exercises to statically analyze the native libraries.

## Table of Contents



## Acknowledgements


This workshop was modeled after the format used by Amanda Rousseau's (@malwareunicorn) RE 101 workshop which she released using GitHub pages. Thanks, Amanda, for the inspiration!





**NEXT** > 2. Android Application Fundamentals


---