# Mobile Application Security

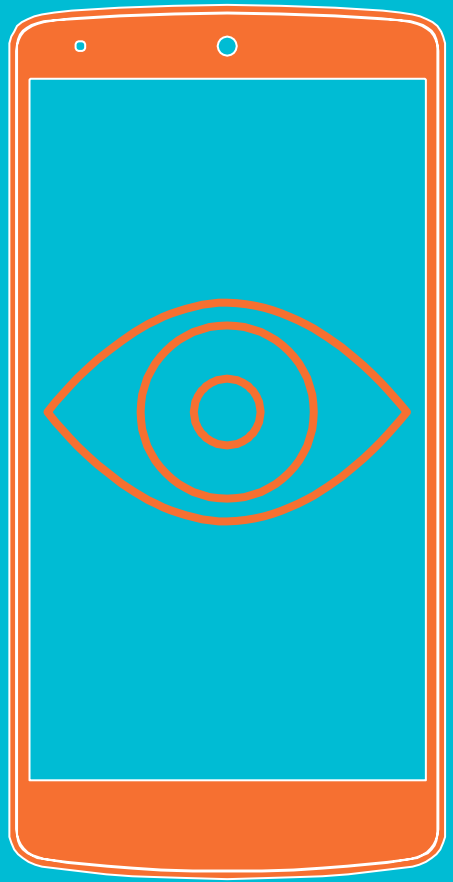
Ethan Tuning
Gavin Rouse
Collin Nolen

# Table of contents

# 1.

## Vulnerabilities

*What vulnerabilities do mobile apps have?*



# Vulnerabilities

*Reports have stated that nearly 95% of mobile applications are vulnerable to attacks.*

**IBM Study Shows:**

40% of enterprises never undergo proper testing of the apps that they create.

Despite huge budgets going into mobile app development, companies spend, next to nothing, on security features.

# Vulnerabilities

## Attacking Techniques

- Reverse Engineering
- Application Data Theft
  - Request Manipulation
  - DoS
  - Injection
  - etc.

# Vulnerabilities

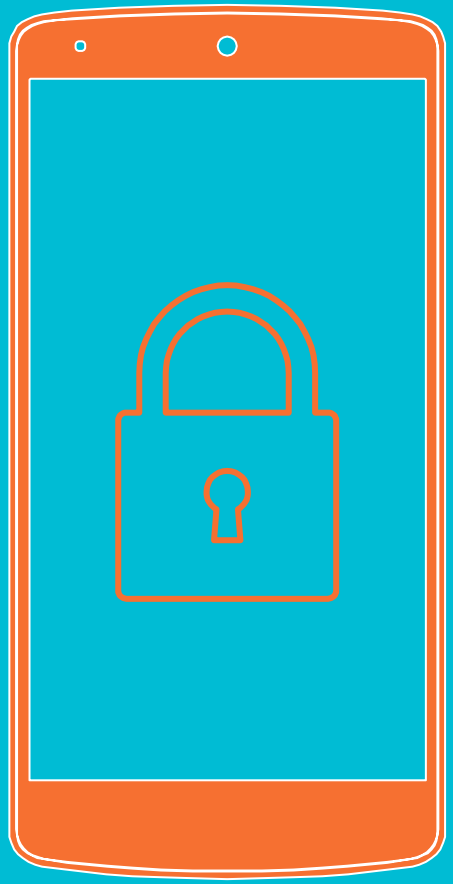
## Most Common Vulnerabilities

- File Permissions
- Client Data Storage
- Transport Layer Security
- Authentication
- Inter Process Communication

# 2.

## Mitigation

*What can be done to ensure security in mobile apps?*



# Mitigation

Three important steps to ensuring the security of a mobile app:

1. Information Gathering
2. Static Analysis
3. Dynamic Analysis

# Mitigation

## Information Gathering

- Often, developers dive headfirst into a project without fully understanding the app or the supporting infrastructure
- Leads to increased chance of insecure code and practices
- Research and identify things such as:
  - Does the app support 3G, 4G, wifi connection
  - Does the app support commerce transactions
  - What hardware components will the app interact with
  - Will the app interact with other apps on the phone
  - What frameworks will the app use

# Mitigation

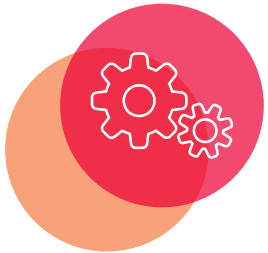
## Static Analysis

- Analyze the app's source code for insecurities
- Analyze and review things such as:
  - Permissions the app requests
  - Resources the app requests
  - Libraries being used are up to date and secure
  - User authentication is secure
  - Sensitive data is properly and securely encrypted
  - Does the app log data, and if so, is any sensitive data logged

# Mitigation

## Dynamic Analysis

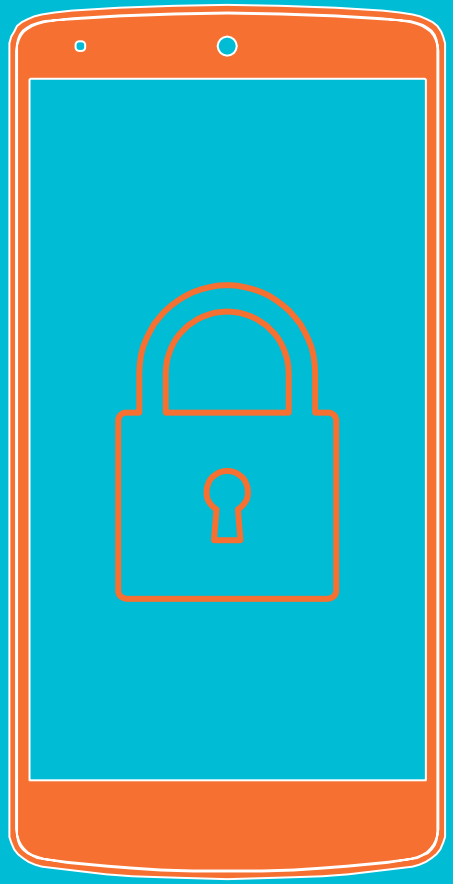
- Using data collected during first two steps, an informed vulnerability assessment can be ran
- Essentially, run the app and attack it
- Try things such as:
  - Fuzz testing
  - Brute force attacks against keys, pins, and hashes
  - Assessing authentication methods
  - Look for unencrypted data storage
  - Dumping device/application memory in order to obtain sensitive information



# 3.

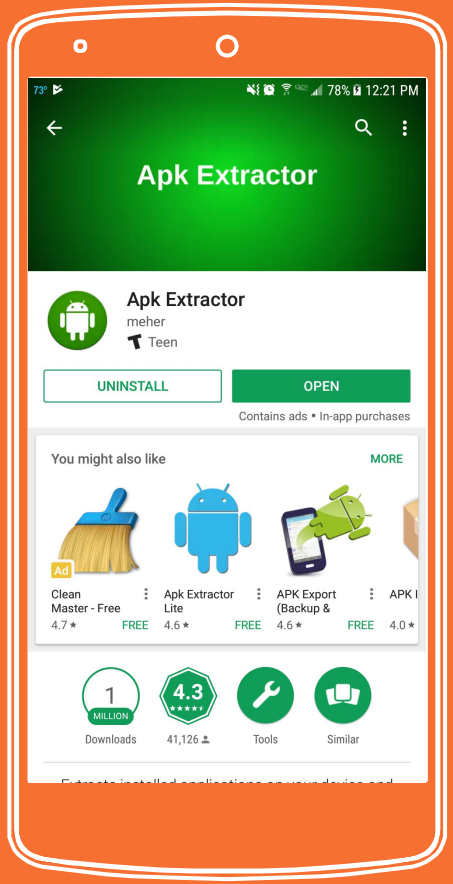
## Example

*Testing applications.*



# Example

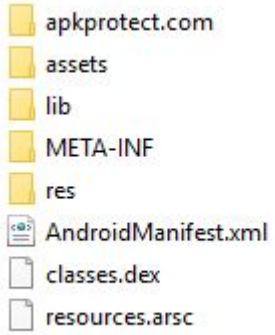
- Install APK Extractor.
- Install application.
- Extract APK
- Move APK file to Computer



# Example

## Unpacking.

- Change the APK's file extension from '.akp' to '.zip'
- Extract files
- Source code lives under 'classes.dex'



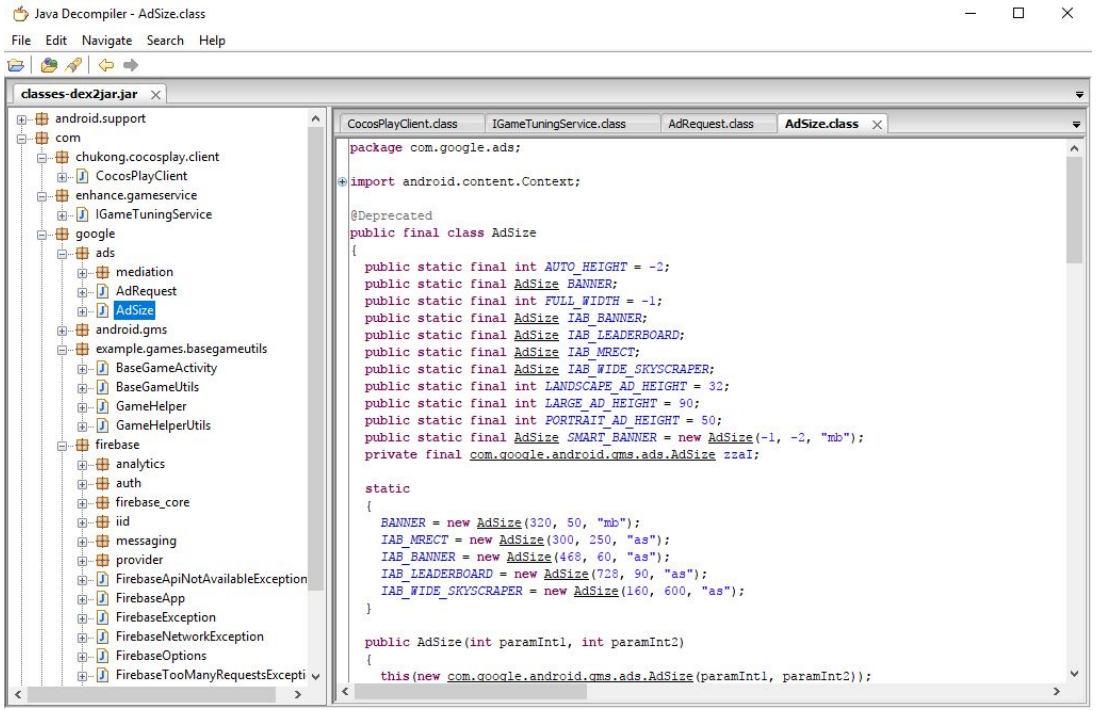

## Converting.

- Download d2j-dex2jar
- Execute code in commandline `d2j-dex2jar classes.dex`
- Converts 'classes.dex' to 'classes.jar'

# Example

## Viewing.

- Download program jd-gui
- Run executable
- Select newly created '.jar' file.




Demo