Android O

Security enforcements

Hello DroidCon!

Javier Cuesta Gómez

Android Engineer manager @Grab

Grab

68 Million

#1 e-hailing in SEA

Largest in SEA

>930,000

drivers in network

142 CITIES

7 COUNTRIES

6 SERVICES

95% share

third-party taxi-hailing apps

>70% share

private cars & growing

Android 2017 security

450 reports
$1.1 payout

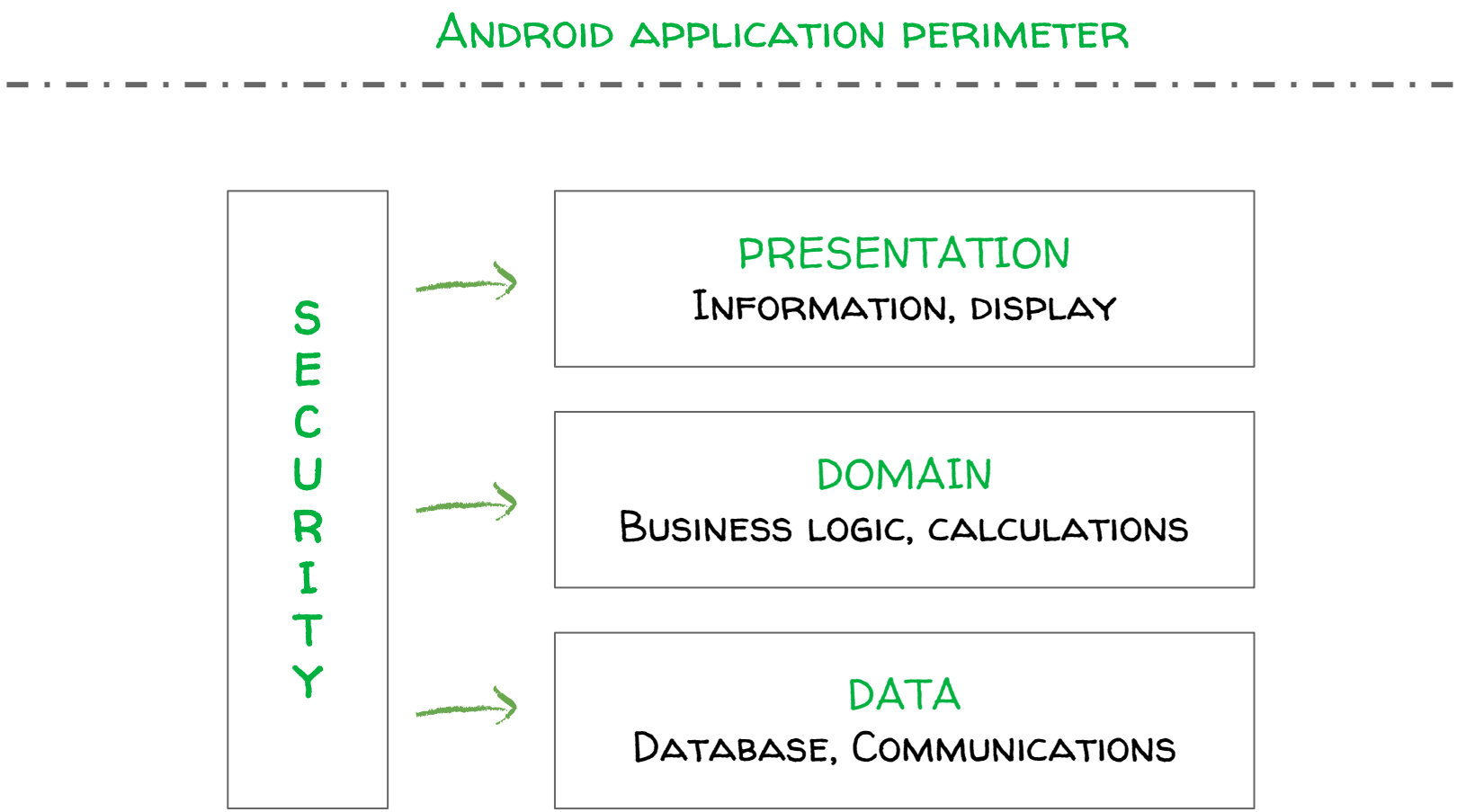
- Unintended data leakage
- Weak server side controls
- Client side injections
- Poor Authorization and Authentication
- Insufficient transport layer protection

Main vulnerable code reasons

1. Rush to release
2. Accidental coding errors
3. Lack of policies requirements

ANDROID

SOFTWARE ARCHITECTURE



SECURITY ENHANCED ARCHITECTURE

ANDROID APPLICATION PERIMETER

SECURITY

THREAT PREVENTION, AUTHENTICATION, AUTHORISATION, SLA

PRESENTATION

INFORMATION, DISPLAY

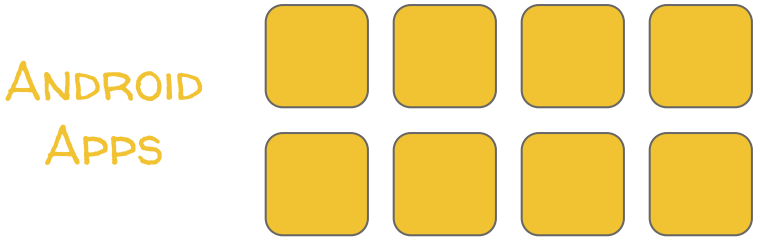
DOMAIN

BUSINESS LOGIC, CALCULATIONS

DATA

DATABASE, COMMUNICATIONS

ANDROID O – PROJECT TREBLE

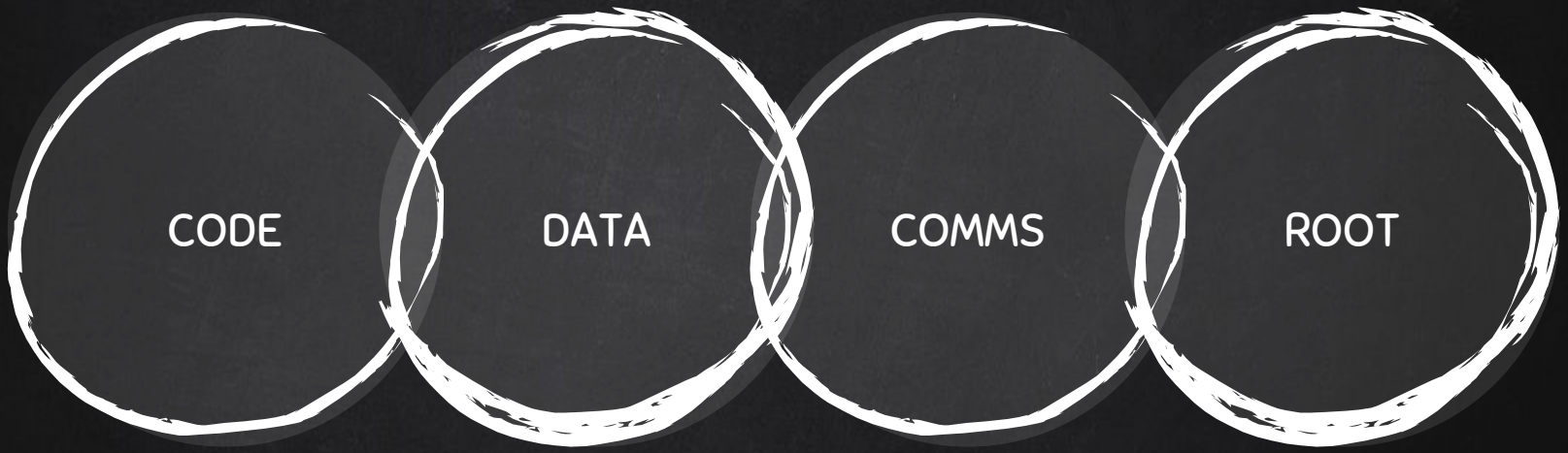


CTS
DEVELOPER API

ANDROID OS FRAMEWORK

VTS
VENDOR INTERFACE

VENDOR IMPLEMENTATION


CODE
DATA
COMMS
ROOT

Enforce security...

In your code

REVERSE ENGINEERING

EXTRACTING KNOWLEDGE OR DESIGN INFORMATION
FROM ANYTHING MAN-MADE.

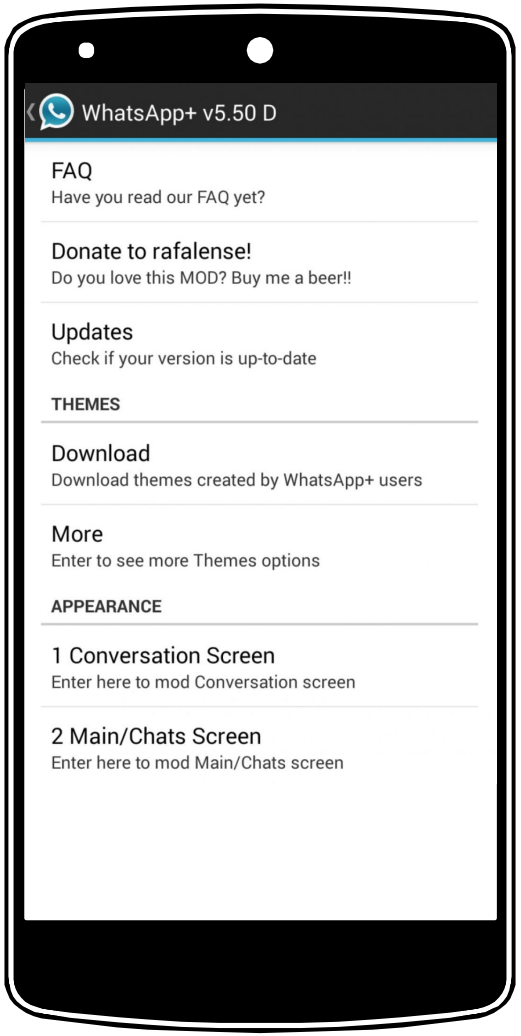
- DOWNLOAD APK FROM BLACK MARKETS — APK MIRROR
- USE REVERSE ENGINEERING TOOLS — APK TOOL
- KNOWLEDGE TAKING:
  - CONSUMER BASIS: ANALYSING AND UNDERSTANDING BEHAVIOUR
  - WHITE HAT: SECURITY ANALYSIS, PENETRATION TESTS, BUG DETECTION, REPORTING
  - BLACK HAT: UPDATING FEATURES, MALWARE, EXPLOITS, VIRUS

CODE EXAMPLE

DECODE RESOURCES TO NEARLY ORIGINAL FORM
REBUILD THEM AFTER MAKING SOME MODIFICATIONS

APKTOOL

- UNZIP APKS ARE NOTHING MORE THAN A ZIP FILE CONTAINING RESOURCES AND ASSEMBLED JAVA CODE. CLASSES.DEX AND RESOURCES.ARSC.
- DECODING: `apktool d testapp.apk` `apktool d foo.jar`
- BEHAVIOURAL MODIFICATION. CHECK BYPASS
- BUILDING: `apktool b foo.jar.out`
- SIGNING
- PUBLISHING



WhatsApp PLUS became one of the best and most used unofficial modes for WhatsApp, allowing users to customize many aspects of the popular instant messaging service with features that the official client doesn't include by default.

WhatsApp Plus has been forced to shut down by WhatsApp in January 2015 due to a cease and desist order.

ProGuard & DexGuard

Shrinked .apk file that is more difficult to Reverse engineer

- Build.gradle MinifyEnabled true
- proguard-rules.txt create proguard-rule
  - -keepattributes *Annotation*
  - -keep public class * extends java.lang.Exception
  - -dontwarn com.crashlytics.**
  - -printmapping mapping.txt

NEW IN ANDROID O

Better App Management and Controls – Malware

setting permissions on a per-app basis, instead of globally allowing all applications to install if the checkbox is enabled, will forced to decide whether they want to download it and what it's permissions should be.

Lot of verification at Play Store to ensure no malware is present, but users can side-load an application from a third-party app store.

Enforce security...

In your data

CRYPTOGRAPHY

Techniques for secure data in the presence of third parties called intruders

Keychain or Android Keystore Provider?

- KeyChain API, **system-wide credentials**
- Android Keystore provider, **individual** app store its own **credentials**. only the app itself can access

Common usages

- Android lock up screen methods. Pattern, Pin, **FingerPrint**
- Android pay

CRYPTOGRAPHY

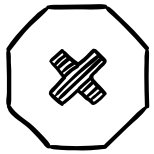
- Algorithms
  - Symmetric: same key (secret), to encrypt and decrypt. AES, DES, Blowfish
  - Asymmetric: different keys. Public and Private. RSA

- Passphrase & seeds
  - Preprocessed hashed (gradle scripts)
  - App specific Os information

- Where to store the keys
  - KeyChain, KeyStore (API 18 4.3 Jelly Bean)
  - SP, DB (being keys encrypted symmetric)

RECOMMENDED

Do not save data on the device



- Android EXTERNAL data storage
  - Use a binary serialized format
  - Secure sensitive data that does not need to be displayed (such as passwords) as a hash. A hash is one-way, it cannot be un-hashed or decrypted.
  - encrypt all sensitive information. Facebook conceal library.

- Android INTERNAL data storage
  - secure because of Android sand boxing its apps. UID level control on the files.
  - Unprotected against rooting and ADB allowing developers to copy data off devices,

NEW IN ANDROID O

SANDBOXING STRATEGY

Separating general Android functionality from manufacturer-specific code has tangible security benefits.

Updatability is a big part of it, but Treble is also really good for helping us sandbox different parts of the operating system

There's now this contrast between the [pure Android] pieces and the device-dependent pieces. If you have an exploit in one side, it is now much harder for that to then exploit the other

Enforce security...

In your Communications

MAN-IN-THE-MIDDLE

BETWEEN CLIENT AND SERVER
EAVESDROPPING OR CHANGING THE DATA.

- WiFi Analyzer: HELPS YOU FIND A GOOD SPOT
- Wireless Tether: CREATE 'FREE_WIFI" HOTSPOT
- ConnectBot: FIGURE OUT WHAT THE WiFi INTERFACE IS ACTUALLY CALLED
- Shark for Root: LOGGING PACKETS

A STEP FURTHER

- Data Siphon: REDIRECTS ALL TRAFFIC FROM HIS ROGUE AP TO A NETWORK WITH HOUSED MACHINES. (Real Time)

CERTIFICATE PINNING

STORING THE INFORMATION FOR DIGITAL CERTIFICATES/PUBLIC KEYS

- Store server certificate within app.
  - What if server certificate gets updated/renewed?
- Replacing the system's TrustStore
  - with one that only contains specific white-listed certificates.
- ping against public certificate Hash

KOTLIN EXAMPLE

```
fun getHttpClientBuilder(addSessionId: Boolean): OkHttpClient.Builder {
    val certificatePinner = CertificatePinner.Builder()
            .add(getPassengerAPIHost(), BuildConfigHelper.CA_CERT)
            .build()

    val builder = OkHttpClient.Builder()
            .connectTimeout(CONNECTION_TIMEOUT, TimeUnit.MILLISECONDS)
            .addInterceptor(mLoggingInterceptor)
            .addInterceptor(HttpHeaderUtils.createRequestInterceptor(addSessionId))

    if (getCertPinSwitch().isCertPinOn()) {
        builder.certificatePinner(certificatePinner)
    }

    return builder
}
```

RECOMMENDED

1. Always use SSL connections if there is anything sensitive – apps data.
2. Never use self signed certificates in production.
3. Disable HTTP redirects in your networking library/code. Some libraries disable this by default. Having these enabled can make MITM attacks a lot easier.
4. If the user is inputting data, always escape it using URLEncoder.encode (userInput, "UTF-8"); if the data will be used as part of a URL, *DB queries as well as if you're saving the input to a JSON or XML file.*
5. Set a maximum length on every field that requires user input.
6. Validate the input.

NEW IN ANDROID O

Better, More Secure Protocols

Oreo's attention to deprecating older insecure protocols for network connections. "The use of SSLv3 for secure HTTPS connections is being discontinued, this prevents the device and its apps from using a known insecure protocol that could leak sensitive data,"

Google has also hardened certain network connection APIs from not falling back to older TLS versions that can leak sensitive data.

Enforce security...

In android os

ROOTING DEVICES

Unlocking the operating system

- **Custom ROM Flashing** Flash a ROM with a modified operating system
  - Advantages: root access is permanent
  - Disadvantages: updates must be shipped by the ROM provider
  - Risks: trust the ROM provider

- **Soft Flashing** keep the factory ROM provided by the manufacturer, modifying it
  - How: custom recovery image to the smartphone

- **Exploiting**
  - Advantages: normally gained through a special one click application. Unroot the device by simply update
  - Disadvantages: root access is just gained temporary

ROOT DETECTION

SPECIFIC PACKAGES AND FILES, DIRECTORY PERMISSIONS, RUNNING CERTAIN COMMANDS.

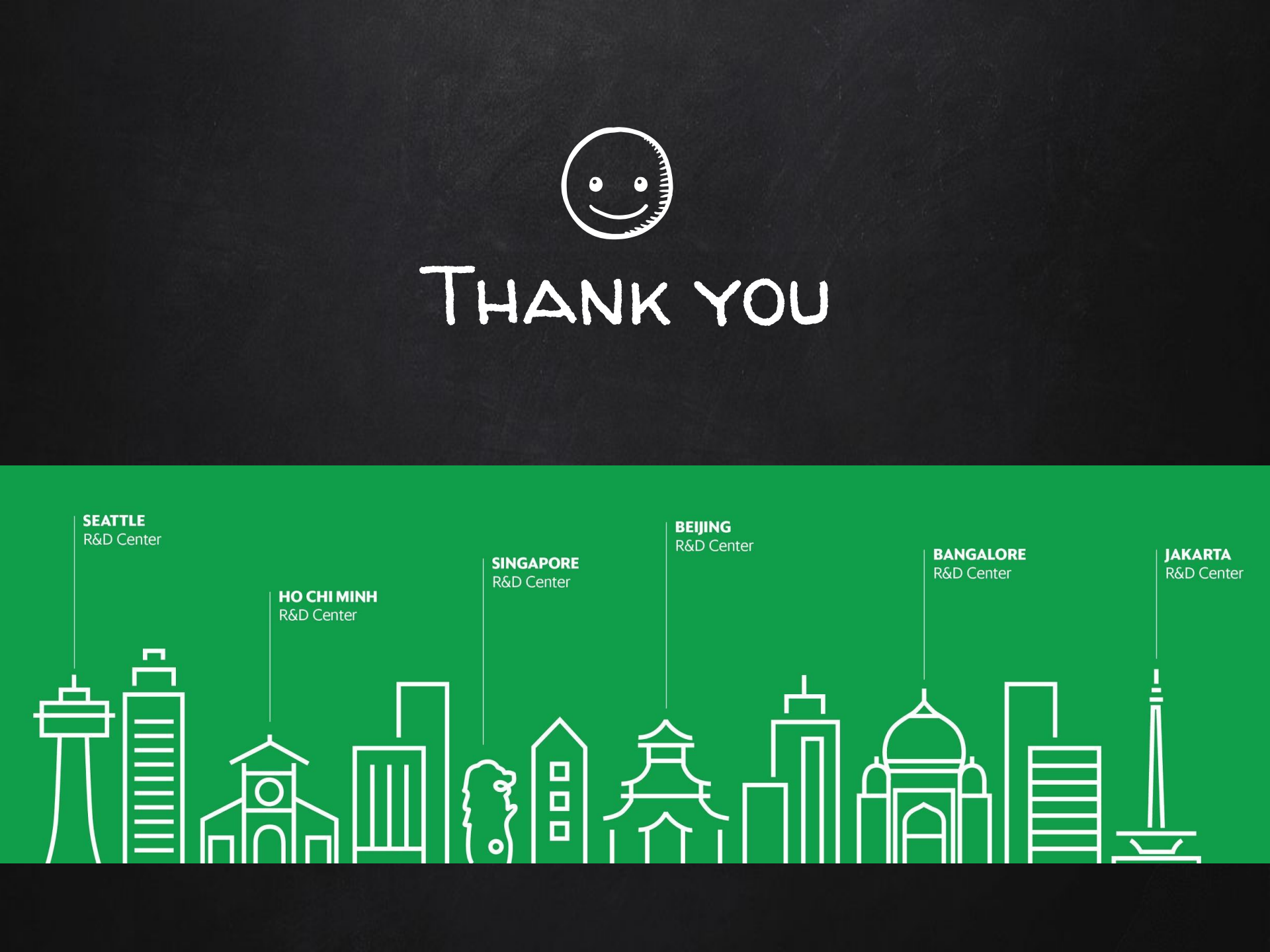
- Checking the BUILD tag for test-keys.
  - By default, stock Android ROMs from Google are built with release-keys tags
- Checking for Over The Air (OTA) certs.
- Existence of su in the path and some other hard-coded directories
  - Multiple libraries available in Github, most common RootTools
- Installed files and packages
  - Superuser.apk
  - Com.noshufou.android.su / com.thirdparty.superuser/ eu.chainfire.supersu ....

NEW IN ANDROID O

Verified Boot System

Verified Boot goes a step further and prevents users or hackers from booting to older more vulnerable versions of the OS an adversary may have rolled the system back to.

The feature also supports the ability for apps and mobile device management firms to secure hardware areas of an Android device upon boot.


THANK YOU
SEATTLE
R&D Center
HO CHI MINH
R&D Center
SINGAPORE
R&D Center
BEIJING
R&D Center
BANGALORE
R&D Center
JAKARTA
R&D Center