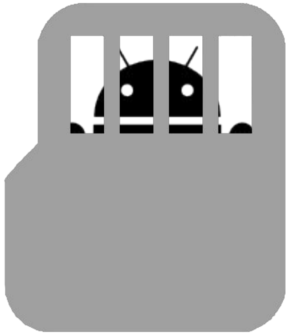
Man-In-The-Disk



Slava Makkaveev

Check Point SOFTWARE TECHNOLOGIES LTD.

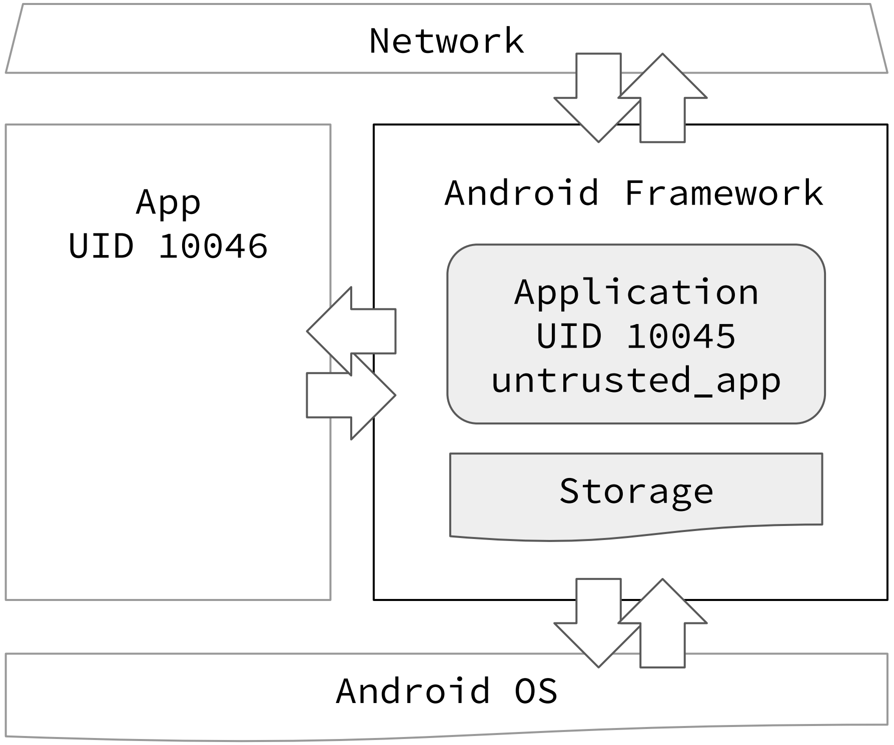
DEF CON 2018

Me

- Slava Makkaveev
    - Security Researcher
    - Check Point Software Technologies Ltd.
    - PhD in Computer Science
    - Reverse engineering and vulnerability research

Android Application Security Basics

App’s Sandbox Model



App’s Permissions

Normal/ Dangerous

- SMS
- CONTACTS
- STORAGE
- ...

Preinstalled/ Privileged

- WRITE_SETTINGS
- INSTALL_PACKAGES
- ...

ROM Signature/ SharedUserId

- ACCOUNT_MANAGER
- OEM_UNLOCK_STATE
- ...

What about Application’s Storage?

App’s Storage

Internal

- Built-in non-volatile memory
- Always available
- Private

External

- Partition in permanent memory
- Public

Removable

- Not always available
- World-readable

Why use External Storage?

- Share media files between apps
- Transfer files between smartphone and PC
- Compatibility with limited inner storage devices
- Hide the current size of the application

External Storage Protection

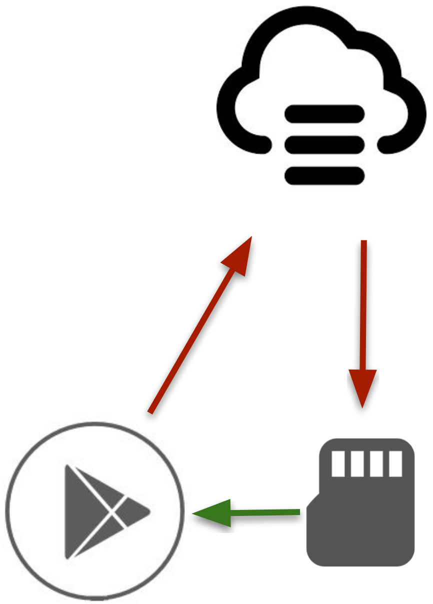
- Global storage access
  - READ_EXTERNAL_STORAGE permission
  - WRITE_EXTERNAL_STORAGE permission

- “Private” directory per application
  - Files are not accessible by MediaStore content provider
  - Observing prevention

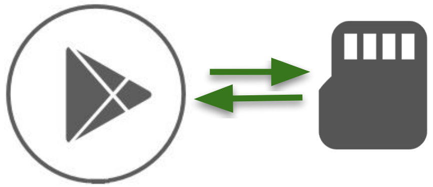
Man-In-The-Disk Attack

External Storage Usage Scenario

Downloading to
external storage

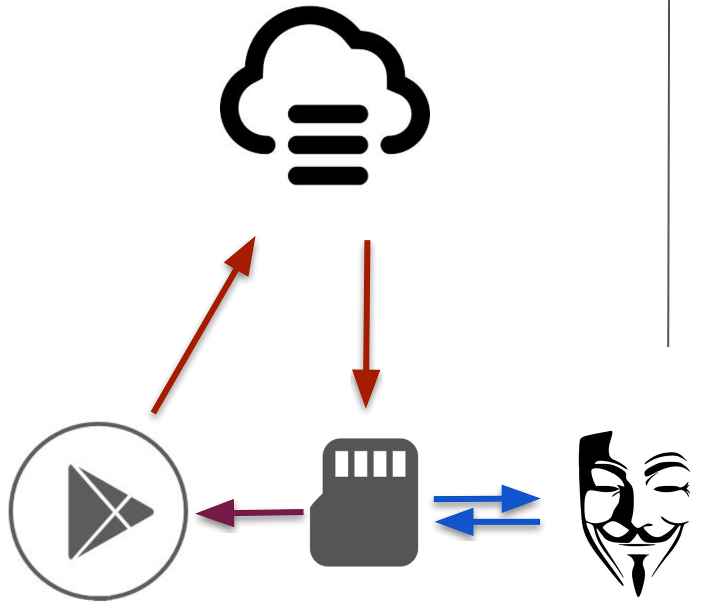


Maintaining working data
on external storage

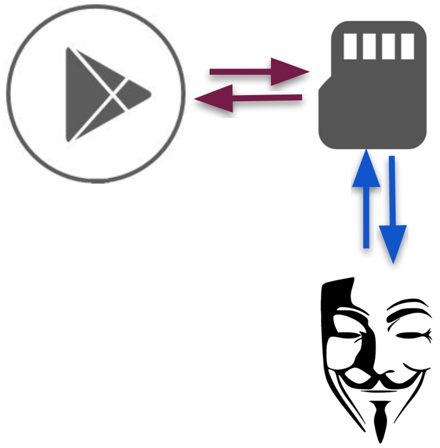


MITD Attack Vectors

Downloading to
external storage



Maintaining working data
on external storage



External Storage Observing

Java

```
FileObserver observer;

observer = new FileObserver(
  "/storage/emulated/0/path/to/folder") {

  @Override
  public void onEvent(int event,
    String file) {
    // ...
  }
};

observer.startWatching();
```

Native

```
int length, i = 0;
int fd, wd;
char buffer[EVENT_BUF_LEN];

fd = inotify_init();
wd = inotify_add_watch(fd, "/tmp", IN_CREATE);
length = read(fd, buffer, EVENT_BUF_LEN);
while (i < length) {
  struct inotify_event *event =
    (struct inotify_event *) &buffer[i];
  // ...
  i += EVENT_SIZE + event->len;
}
inotify_rm_watch(fd, wd);
close(fd);
```

Private Directory Observing

No notification → Polling method

```
File watchDir;
Timer timer;

watchDir = new File(Environment.getExternalStorageDirectory().toString() + "/path/to/folder");
timer = new Timer();
final int FPS = 100;
timer.scheduleAtFixedRate(new ObserverTask(), 0, 1000/FPS);

class ObserverTask extends TimerTask {
  public void run() {
    File[] files = watchDir.listFiles();
    for (int i = 0; i < files.length; i++) {
      // ...
    }
  }
}
```

Security guide based on “Android developer training articles”

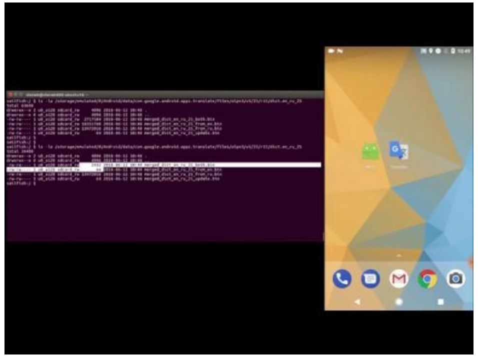
“You should perform input validation when handling data from external storage…”

Google Translate (com.google.android.apps.translate)

Holds offline mode translation packages on external storage

libtranslate.so is compromised

../Android/data/com.google.android.apps.translate/files/olpv3/v5/25/r11/



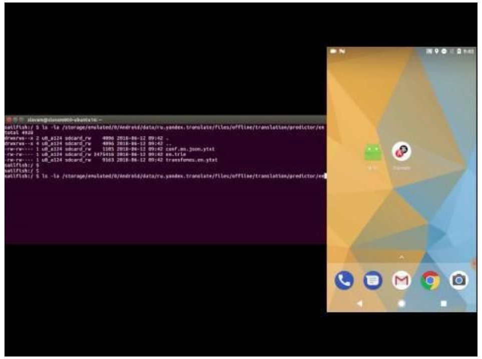
```
[illegible]
```

Yandex Translate (ru.yandex.translate)

Holds offline mode translation packages on external storage

libmobile-android.so is compromised

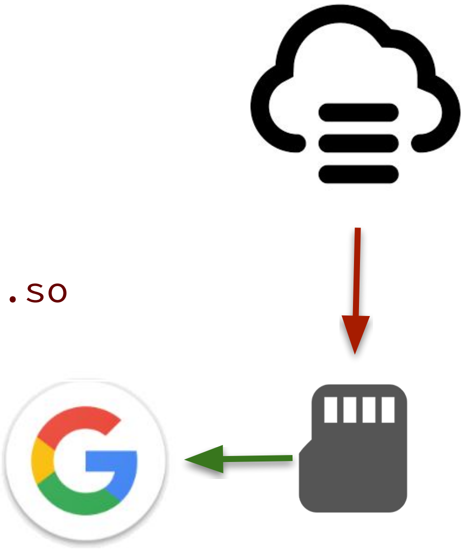
../Android/data/ru.yandex.translate/files/offline/translation/

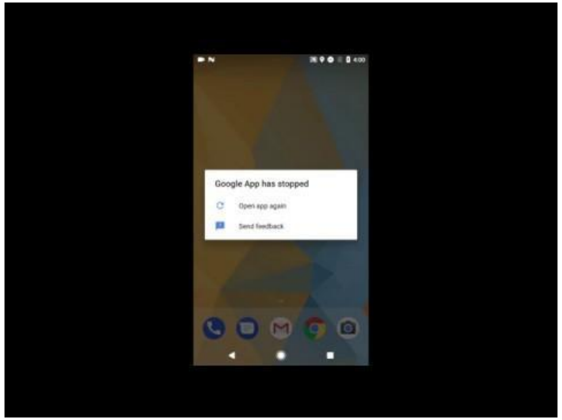


```
sailfish:/ $ ls -la /storage/emulated/0/Android/data/ru.yandex.translate/files/offline/translation/predictor/en
total 4920
drwxrws--x 2 u0_a124 sdcard_rw    4096 2018-06-12 09:42 .
drwxrws--x 4 u0_a124 sdcard_rw    4096 2018-06-12 09:42 ..
-rw-rw---- 1 u0_a124 sdcard_rw    1105 2018-06-12 09:42 conf.en.json.ytxt
-rw-rw---- 1 u0_a124 sdcard_rw 2475410 2018-06-12 09:42 en.trie
-rw-rw---- 1 u0_a124 sdcard_rw    9163 2018-06-12 09:42 transfomes.en.ytxt
sailfish:/ $
sailfish:/ $
sailfish:/ $ ls -la /storage/emulated/0/Android/data/ru.yandex.translate/files/offline/translation/predictor/en
```

Google Voice Typing (com.google.android.googlequicksearchbox)

Downloads offline speech recognition languages through external storage





libgoogle_speech_jni.so
is compromised

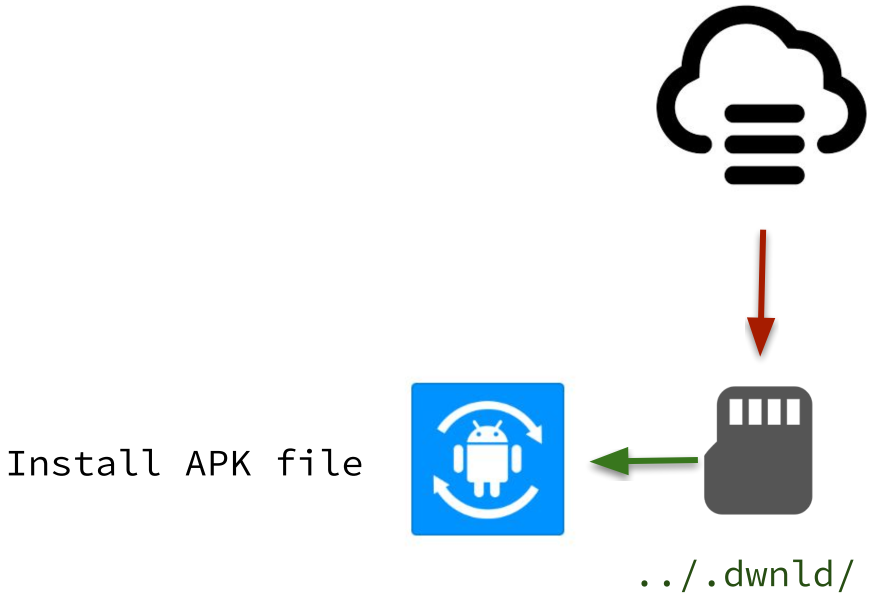
../app_g3_models/

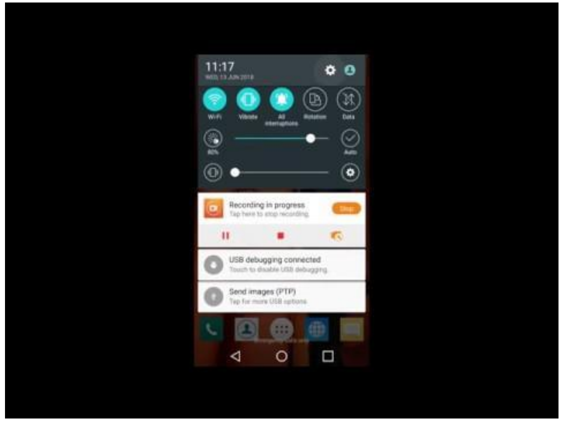
../Android/data/com.google.android.googlequicksearchbox/files/download_cache/

“You should not store executables or class files on external storage…”

LG Application Manager (com.lge.appbox.client)

Installs/Updates LG related apps through external storage





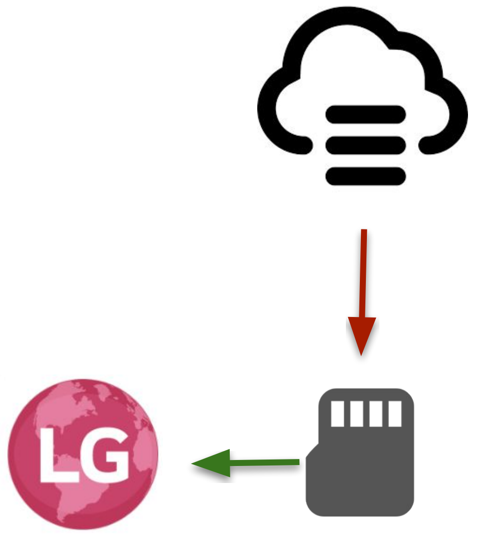
Recording in progress
Tap here to stop recording
Stop

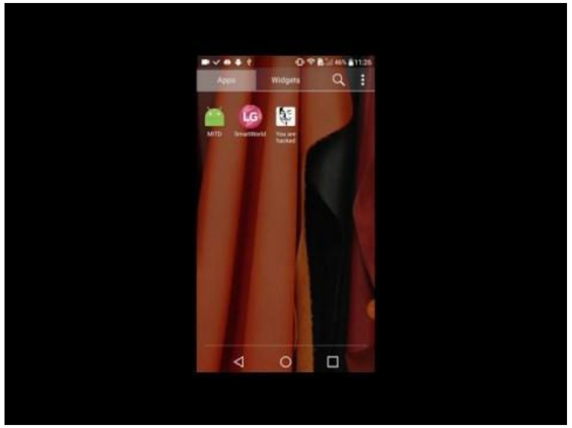
USB debugging connected
Touch to disable USB debugging

Send images (PTP)
Tap for more USB options

LG World (com.lge.lgworld)

Updates itself through external storage

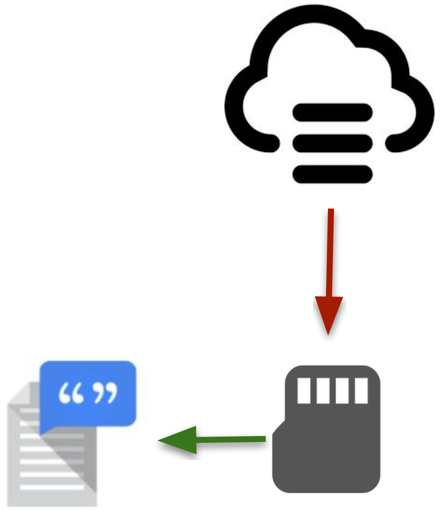




“... external storage files should be signed and cryptographically verified prior to dynamic loading...”

Google Text-to-speech (com.google.android.tts)

Downloads voice data through external storage



1. Downloading of a voice packet (zip file) to external storage

2. Verification of voice packet’s signature

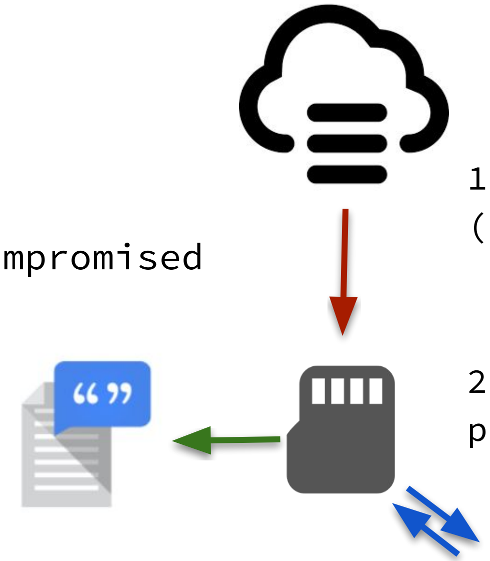
3. Decompression of the voice packet to inner storage

../app_voices_greco_v2/

../Android/data/com.google.android.tts/files/download_cache/

Google Text-to-speech (com.google.android.tts)

Downloads voice data through external storage



1. Downloading of a voice packet (zip file) to external storage

libtts_android.so is compromised

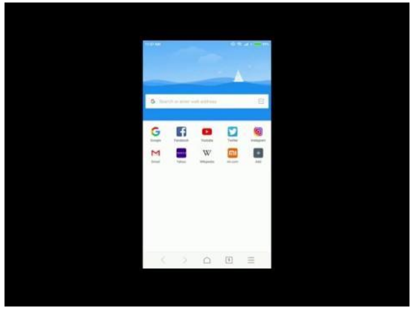
3. Decompression of the voice packet to inner storage

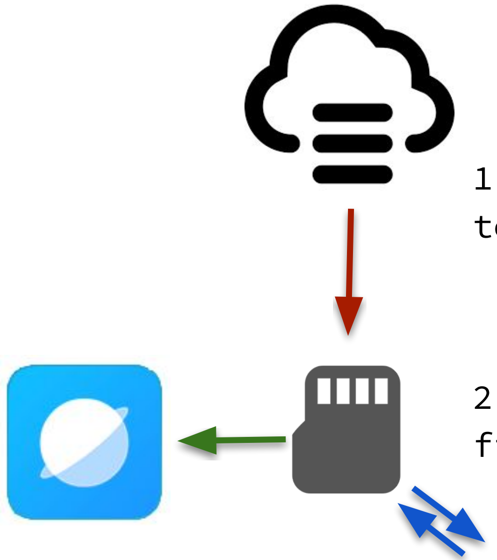
2. Verification of voice packet's signature

2.5 Overwrite the voice packet

Xiaomi Browser (com.android.browser package)

Updates itself through external storage





1. Downloading of APK file to external storage

2. Verification of APK file's SHA1 hash

2.5 Overwrite the APK

3. Installation of the APK

Summary

- Device external storage is a public area which can be observed/modified by a third-party application as well as the device user
- Android does not provide relevant protection for the data in the external storage
- Many ROM pre-installed and popular apps hold sensitive data in the external storage
- Storage-based Man-In-The-Disk attack can break fortified Android app’s sandbox protection

Hunting for Man-In-The-Disk

MITD Research

Target

- Application’s native library (*.so)
- A code flow that handles (parses/decodes/etc.) a controllable data file

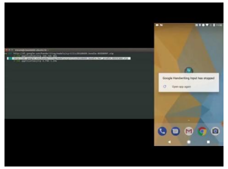
Research approach

- Implement simplest Java to Native adapter to reproduce the flow
- Fuzz the native lib by permutation of the data in the controllable file

Java to native adapter

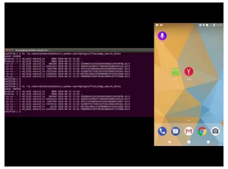
Google Handwriting

```
public static void load() {
  System.load("/path/to/libgnustl_shared.so");
  System.load("/path/to/libhwrword.so");
}

public static void main(String[] args) {
  FileInputStream st1 = new FileInputStream(
    args[0]);
  FileInputStream st2 = new FileInputStream(
    "/path/to/hwr_prodlm.4DE9C666");

  WordRecognizerJNI rec = new WordRecognizerJNI();
  rec.initJNIFromFileInputStream(
    st1, 0, st1.getChannel().size(),
    st2, 0, st2.getChannel().size(),
    null, 0, 0, null, 0);
}
```

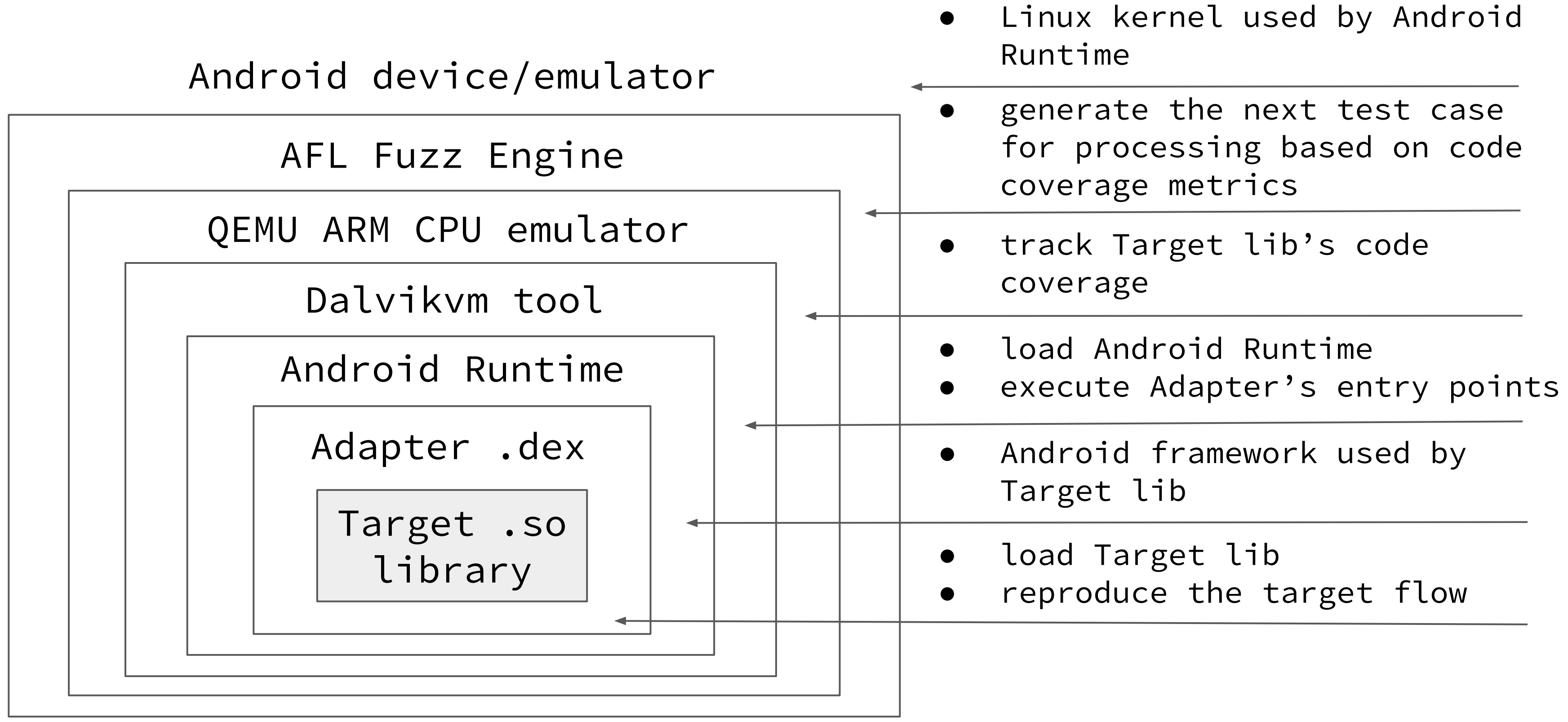


Yandex Search

```
public static void load() {
  System.load("/path/to/liboffline_search-data_reader.so");
  System.load("/path/to/liboffline_search.so");
}

public static void main(String[] args) {
  copyFile(args[0], "/path/to/edge_search_dicts/xxx.dict");

  long searchObj =
    OfflineSearchNativeC.JELOfflineSearchLibraryCreate(
      "/path/to/edge_search_dicts", 0, 2);

    OfflineSearchNativeC.JELOfflineSearchLibraryCreateSuggestions(
      searchObj, "a");
}
```



Fuzzing of Application native



- Linux kernel used by Android Runtime
- generate the next test case for processing based on code coverage metrics
- track Target lib's code coverage
- load Android Runtime
- execute Adapter's entry points
- Android framework used by Target lib
- load Target lib
- reproduce the target flow

Thank you!

github.com/CheckPointSW/android_appfuzz

slavam@checkpoint.com