CVE-2013-4787(Master Key Vulnerability) & Zip implementation of Android

Masata NISHIDA

AVTOKYO2013.5 (16th Feb. 2014)

English Ver.

Who am I?

- Masata Nishida (西田 雅太)
- SecureBrain
- I’ m not a malware researcher, I’ m just a software developer.
- Rubyist
- @masata_masata

Agenda

- Explanation about CVE-2013-4787 with source code
- ~~About Zip implementation of Android~~

CVE-2013-4787 Master Key Vulnerability

http://www.flickr.com/photos/plaisanter/5344873860/

CVE-2013-4787
Master Key Vulnerability

Uncovering Android Master Key That Makes 99% of Devices Vulnerable

Like Share

Written By Jeff Forristal, Bluebox CTO

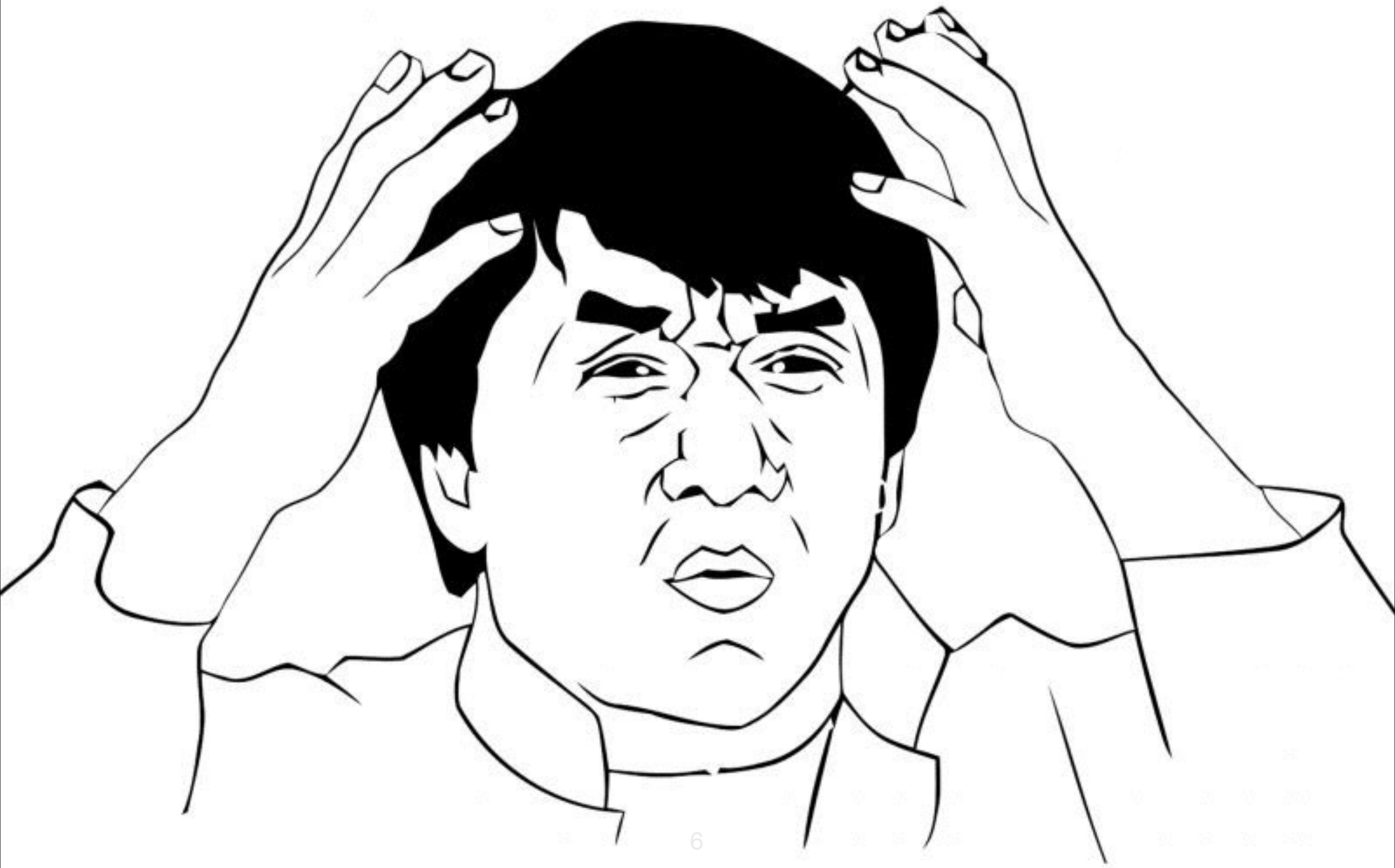
The Bluebox Security research team – Bluebox Labs – recently discovered a vulnerability in Android's security model that allows a hacker to modify APK code without breaking an application's cryptographic signature, to turn any legitimate application into a malicious Trojan, completely unnoticed by the app store, the phone, or the end user. The implications are huge! The vulnerability, which has existed since version 1.6 of Android (codenamed "Donut"), could affect any Android phone released in the last 4 years[1] – or nearly 900 million devices[2] – and depending on the type of application a hacker can exploit the vulnerability for anything from data theft to creation of a mobile botnet.

- Vulnerability in Android OS
- The Attacker can inject any code into the app without changing the signature of target application.

⇒ huge impact & user can't notice this attack.

http://bluebox.com/corporate-blog/bluebox-uncovers-android-master-key/

OH CRAP!!



Keywords

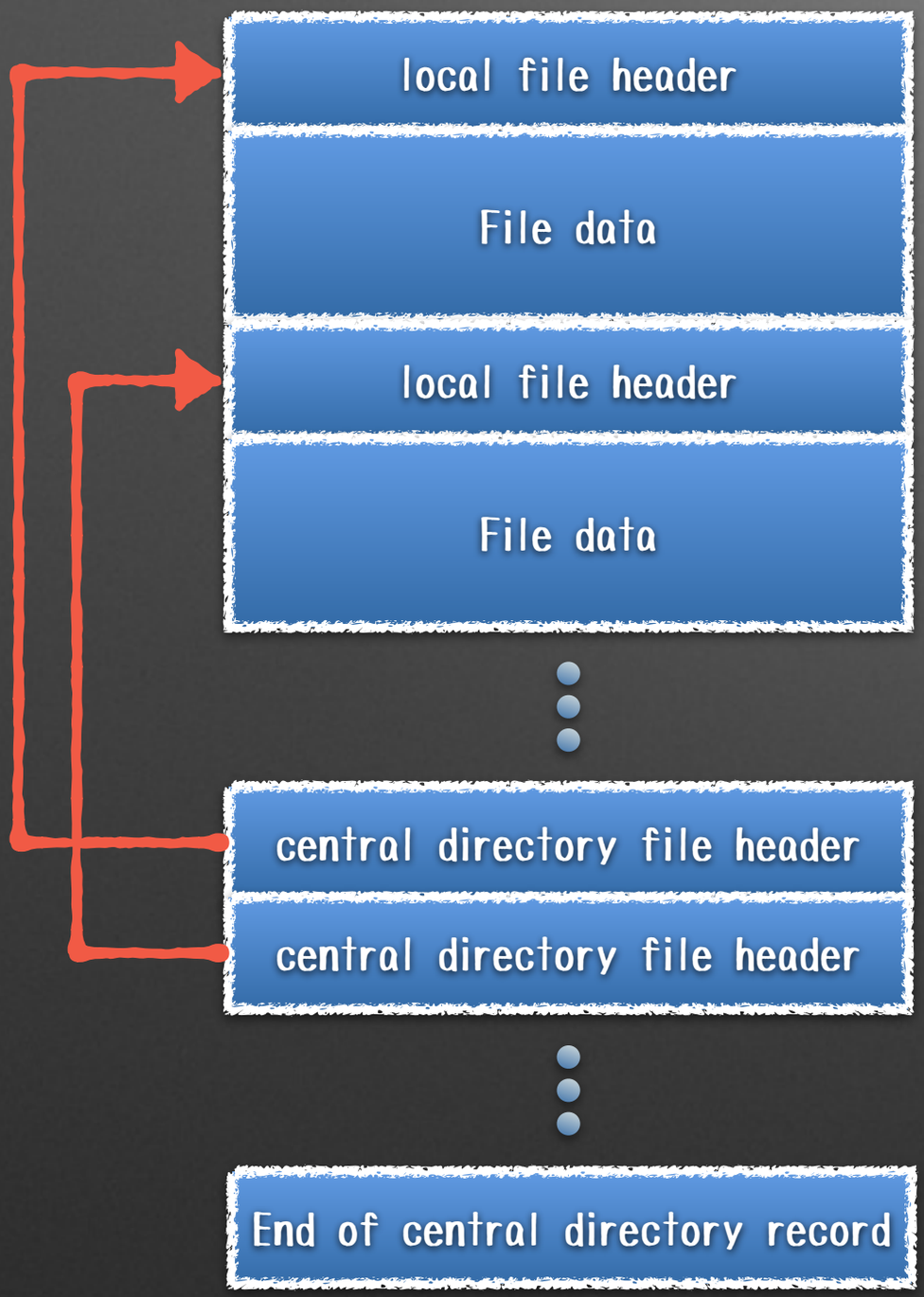
- Apk file = Zip file
- Zip file format
- Confirming a signature of Apk

http://www.flickr.com/photos/mrcam/206628965/

APK File
(Android Application Package)

APK file is Zip file.
It must include executable file for dalvik VM and manifest file.

Zip Format



Local File Header (file infomation)
File Data (compressed data)

Central Directory File Header
file name, file size,
offset of file data,
compression method…etc

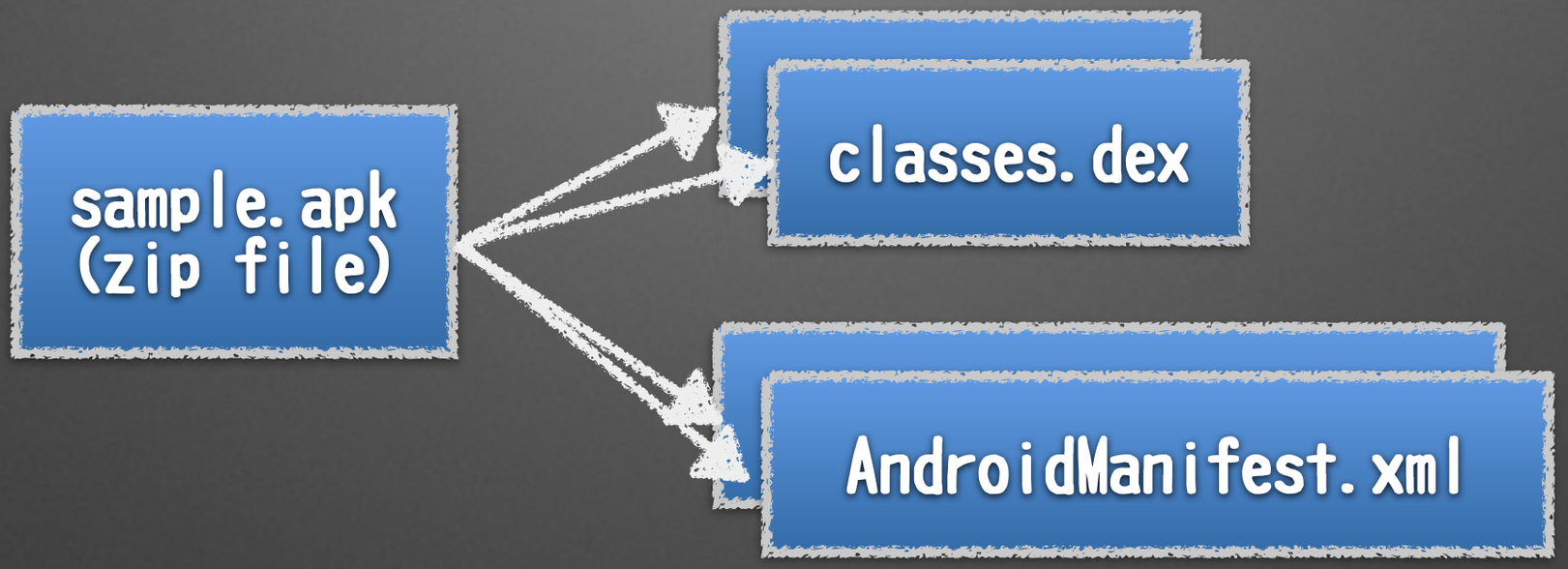
Confirming a signature

- All applications must be digitally signed.
- You can't change any files in the apk after signing.
- Android checks the sign when user installs an new application.

http://www.flickr.com/photos/gearys/276917907/

If an apk has duplicated file name entries...

APK File
duplicated files



According to the specification, zip file can contain duplicated file name entries.

It's the implementation problem.

APK File
duplicated files

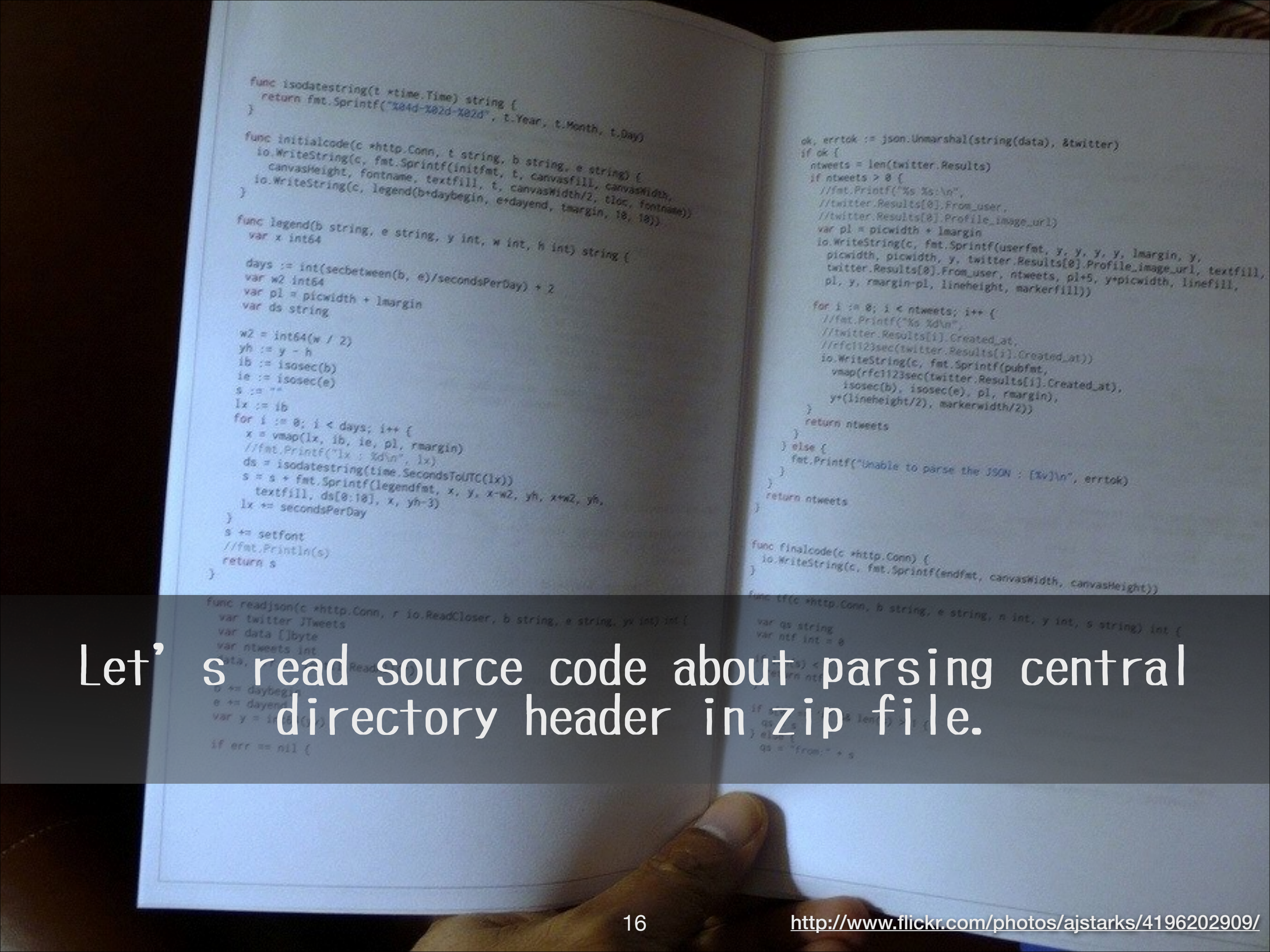
Inject classes.dex and manifest file into apk,
and install it...


You can install the application
includes another code with
original signature!

Why ?

- Android OS has a number of zip implementation.
  - checking signature → java.util.zip
  - installing application → C++
    - frameworks/native/libs/utils/ZipFileRO.cpp
    - dalvik/libdex/ZipArchive.cpp

The behavior differences between java and C++ implementation cause the issue when an Apk includes duplicate entries.



Let’s read source code about parsing central directory header in zip file.

http://www.flickr.com/photos/ajstarks/4196202909/

ZipFileRO & libdex/ZipArchive

- libdex/ZipArchive is almost the same as ZipFileRO.

- It parses Central Directory File Header.
  Then it sets the file names into hash table.

```
bool ZipFileRO::parseZipArchive(void)
{
  :
  :
    const unsigned char* ptr = cdPtr;
    for (int i = 0; i < numEntries; i++) {
      :
      :
        unsigned int fileNameLen, extraLen, commentLen, hash;

        fileNameLen = get2LE(ptr + kCDENameLen);
        extraLen = get2LE(ptr + kCDEExtraLen);
        commentLen = get2LE(ptr + kCDECommentLen);

        /* add the CDE filename to the hash table */
        hash = computeHash((const char*)ptr + kCDELen, fileNameLen);
        addToHash((const char*)ptr + kCDELen, fileNameLen, hash);

        ptr += kCDELen + fileNameLen + extraLen + commentLen;
    :
    :
  }
:
:
}
```

append file name into hash table

compute hash value with file name

ZipFileRO & libdex/ZipArchive

```
void ZipFileRO::addToHash(const char* str, int strLen, unsigned int
hash)
{
    int ent = hash & (mHashTableSize-1);

    /*
     * We over-allocate the table, so we're guaranteed to find an empty
slot.
     */
    while (mHashTable[ent].name != NULL)
        ent = (ent + 1) & (mHashTableSize-1);

    mHashTable[ent].name = str;
    mHashTable[ent].nameLen = strLen;
}
```

ZipFileRO finds next empty element and sets the file name into the hash table, if the hash value is duplicated.
→ use first item.

Zip implementation in Java

```
public class ZipFile implements Closeable, ZipConstants {

    private final LinkedHashMap<String, ZipEntry> mEntries = new LinkedHashMap<String, ZipEntry>();

    private void readCentralDir() throws IOException {
        :
        :
        RAFStream rafStream = new RAFStream(raf, centralDirOffset);
        BufferedInputStream bufferedStream = new BufferedInputStream(rafStream, 4096);
        byte[] hdrBuf = new byte[CENHDR]; // Reuse the same buffer for each entry.
        for (int i = 0; i < numEntries; ++i) {
            ZipEntry newEntry = new ZipEntry(hdrBuf, bufferedStream);
            mEntries.put(newEntry.getName(), newEntry);
        }
    }

}
```

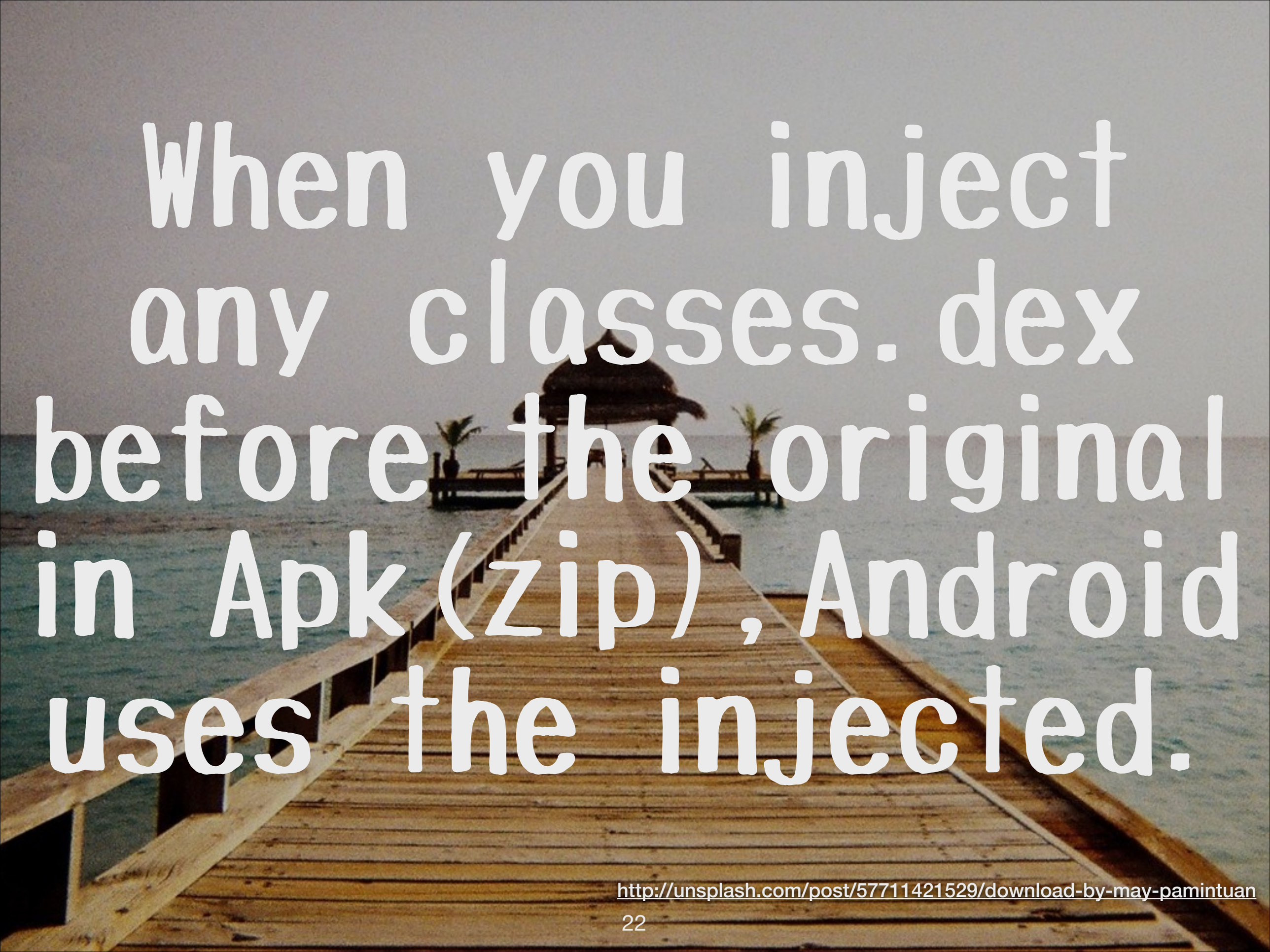
Java sets zip entries into HashMap. The file names are used for the key of HashMap.

duplicated file name → overwrite HashMap → use last item

…therefore

- Checking signature → use last item
  - use last zip entry
- Installing application → use first item
  - use first zip entry

http://www.flickr.com/photos/49121171@N00/6831724767/



http://unsplash.com/post/57711421529/download-by-may-pamintuan

Patch

Android security bug 8219321

```
// Seek to the first CDE and read all entries.
RAFStream rafs = new RAFStream(mRaf, centralDirOffset);
BufferedInputStream bin = new BufferedInputStream(rafs, 4096);
byte[] hdrBuf = new byte[CENHDR]; // Reuse the same buffer for each entry.
for (int i = 0; i < numEntries; ++i) {
    ZipEntry newEntry = new ZipEntry(hdrBuf, bin);
-       mEntries.put(newEntry.getName(), newEntry);
+       String entryName = newEntry.getName();
+       if (mEntries.put(entryName, newEntry) != null) {
+           throw new ZipException("Duplicate entry name: " + entryName);
+       }
    }
}
```

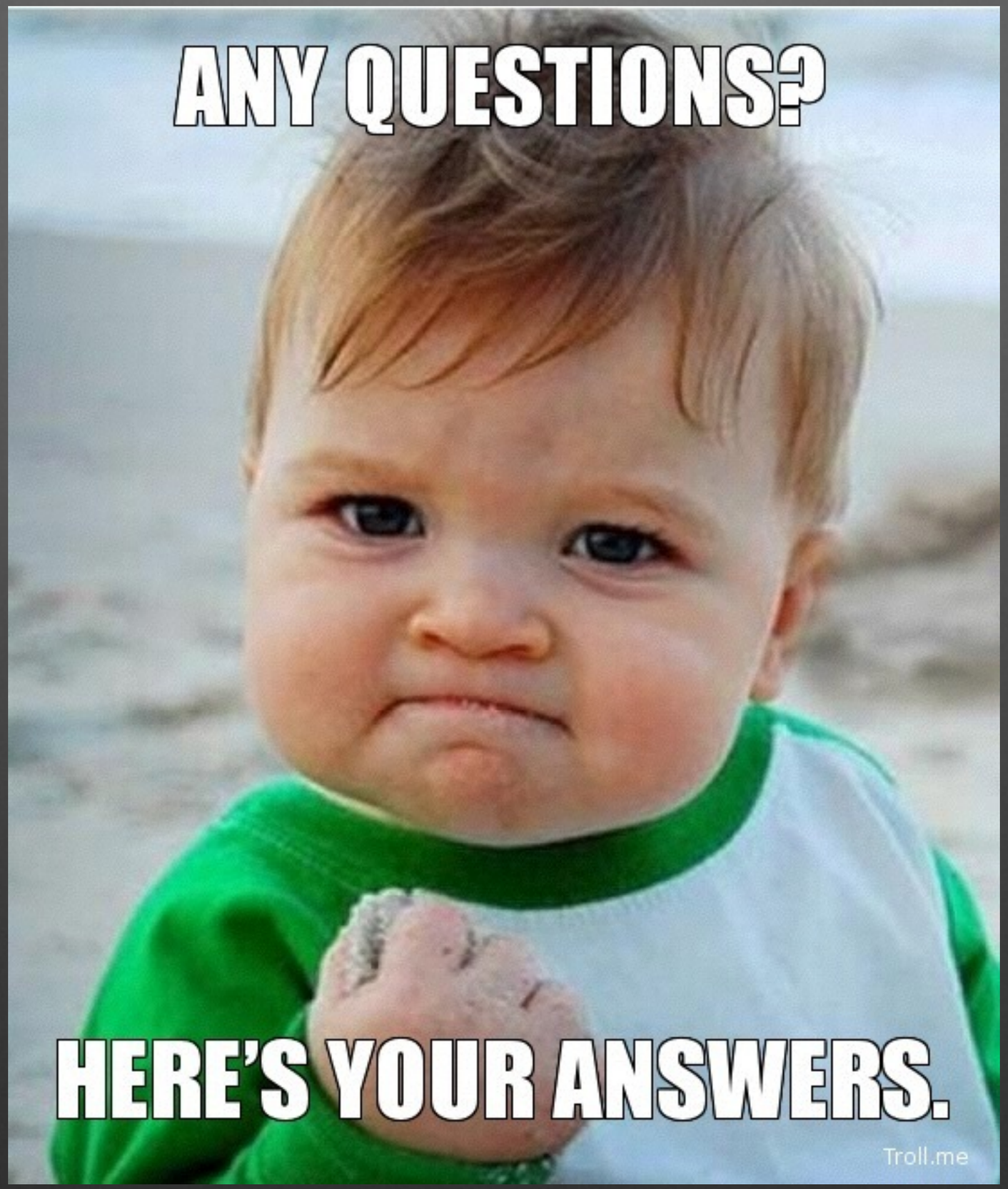
Fix Java code.
Throw exception when duplicated entry is found.

Appendix：Iffy zip implementation in Android

http://www.flickr.com/photos/26207806@N00/1315037834/

Zip implementation in Android

- Android doesn't check zip file header in detail when it installs an application.
- Ignoring encrypt flag.
- Only deflate and no compressed are allowed as compression method.


ANY QUESTIONS?
HERE'S YOUR ANSWERS.
Troll.me