Updated Dalvik VM Dex File Format

November 21, 2008, Tim

(lame dex file photoshopping joke huh?)

In my quest to writing a successful injector I' ve had to do a ton of digging into the dex file format. While mostly everything is open source, it' s not exactly easy to find all of the information – let alone understand it. A great resource I' ve mentioned previously was the "Dalvik VM Dex File Format" over at retrodev.org. This resource is sadly out dated and no longer updated by pavone, but it does provide a wealth of information. I figured I' d post my results just as pavone has done so that anyone looking for the information will hopefully find it. Note that pavone' s version of the dex file he was examining was 'dex 009' according to the magic. The current one as of this posting is 'dex 035'. I' ll repost this data as I figure out more about it and exactly how it is modified.

Magic – 8 bytes – "dex\n035\0"
Checksum – 4 bytes – Adler32 checksum from bytes offset 12 and on
Signature – 20 bytes – SHA-1 of bytes from 32 on
File Size – 4 bytes – Exactly what it sounds like, the file size
Header Size – 4 bytes – Will always be "70"
Endian Tag – 8 bytes – Will always be "78563412"
Zeros – 8 bytes – Exactly that, eight bytes of zeros
Map Offset – 4 bytes – Leads to below, need more research on this though
String Table Size – 4 bytes – Size of the string' s table
String Table Offset – 4 bytes – Offset to the string table
TypeTable Size – 4 bytes – Size of the type' s table
Type Table Offset – 4 bytes – Offset to the type table
Prototype Table Size – 4 bytes – Size of the prototype' s table
Prototype Table Offset – 4 bytes – Offset to the prototype table
Field Table Size – 4 bytes – Size of the field' s table
Field Table Offset – 4 bytes – Offset to the field table
Method Table Size – 4 bytes – Size of the method' s table
Method Table Offset – 4 bytes – Offset to the method table
Class Table Size – 4 bytes – Size of the class' s table
Class Table Offset – 4 bytes – Offset to the class table

You can easily note that all the sizes of these fields end up adding up to 0×70, which is the "Header Size". Also if above isn' t clear enough, after a dex file is created, the signature is applied – which is a SHA-1 digest of all the bytes below it' s position. The checksum is an Alder32 hash of all the bytes below itself, including the signature. I actually discussed this in a previous post where I posted the code for "ReDEX", the post was entitled "DEX File signature and checksums".

I’ m actually revamping the “ReDEX” code to check and spit out this relevant information and more, though it’ s not fully done. I’ m also doing more research into the “Map” field and will hopefully be able to explain more about what is store, how it is stored and what not – more like the information originally presented on retrodev. Until then, this information will have to suffice, enjoy!

Tim
3 Comments

1.

 Reply

There was a mistake in your information, “Endian Tag” field, should be 4 bytes.

dandycheung July 12, 2009 at 11:48 pm

2.

 Reply

The table formats have also been altered. What a pain!

From what I’ ve figured out, the string table is simply a list of offsets, (unsigned 4 bytes big endian), that point to each string constant. The string constants have the format where is a one-byte unsigned length and is a zero byte.

Michael Maloney December 10, 2009 at 8:45 pm

3.

 Reply

Err, correction. It’ s actually LITTLE endian. But I suppose the Endian field will tell you which way to do it

Michael Maloney December 10, 2009 at 10:13 pm

Your Name Email Website

Submit Comment

Strazzere

...it all can be reversed

Search Search

Twitter: timstrazz

- Ops, twitter doesn't seem to be responding...

Categories

- android (97)
- archos (3)
- coding (13)
- dex bytecode (10)
- life (9)
- max os x (1)
- other (15)
- random (11)
- reverse engineering (36)
- reversing (2)
- secure code (10)
- updating (8)
- windows (1)

Android

- @timstrazz
- AndroidXRef
- DexLabs
- Github : strazzere
- i, Claud (Chinese)
- Mobile Forensics/Malware
- SecKungFu (Chinese)
- Thomas Cannon's Blog

Malware

- Contagio
- Contagio Mobile
- DarkLapu's Malware Notes

OSX

- Reverse Engineering OSX

Recent Comments

- What’ s a known source of malware doing in an iOS app? Ars investigates – Ars Technica | Finance Chit Chat on Javascript Malware Cross-Contamination in Android apks
- What’ s a known source of malware doing in an iOS app? Ars investigates – Ars Technica | BREAKINGNEWSHOURLY.COM on Javascript Malware Cross-Contamination in Android apks
- What’ s a known source of malware doing in an iOS app? Ars investigates – Ars Technica | Net News Online on Javascript Malware Cross-Contamination in Android apks
- What’ s a known source of malware doing in an iOS app? Ars investigates – Ars Technica | Latest News on Javascript Malware Cross-Contamination in Android apks
- What’ s a known source of malware doing in an iOS app? Ars investigates – Ars Technica | So Non Fiction on Javascript Malware Cross-Contamination in Android apks