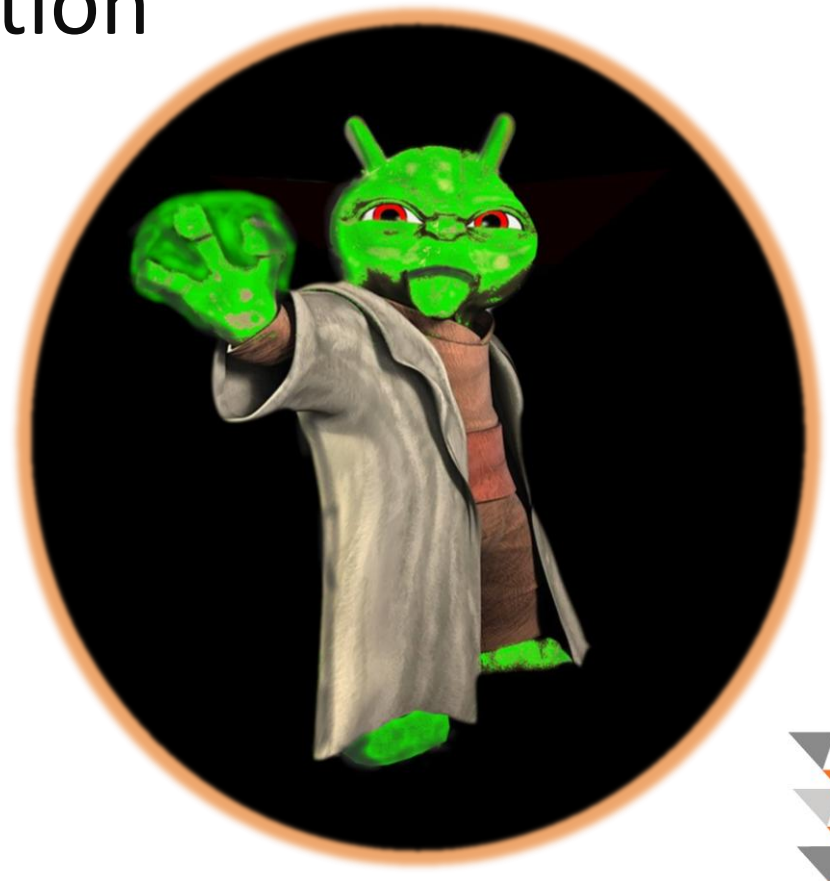
"I am not the D'r.0,1d You are Looking For": An Analysis of Android Malware Obfuscation



Samir Mody
Senior Manager
K7 Threat Control Lab

Making a Hash of it?

Default detection strategy

Checksums & Cloud-lookups

Resource limitations

Memory, Disk space, Power

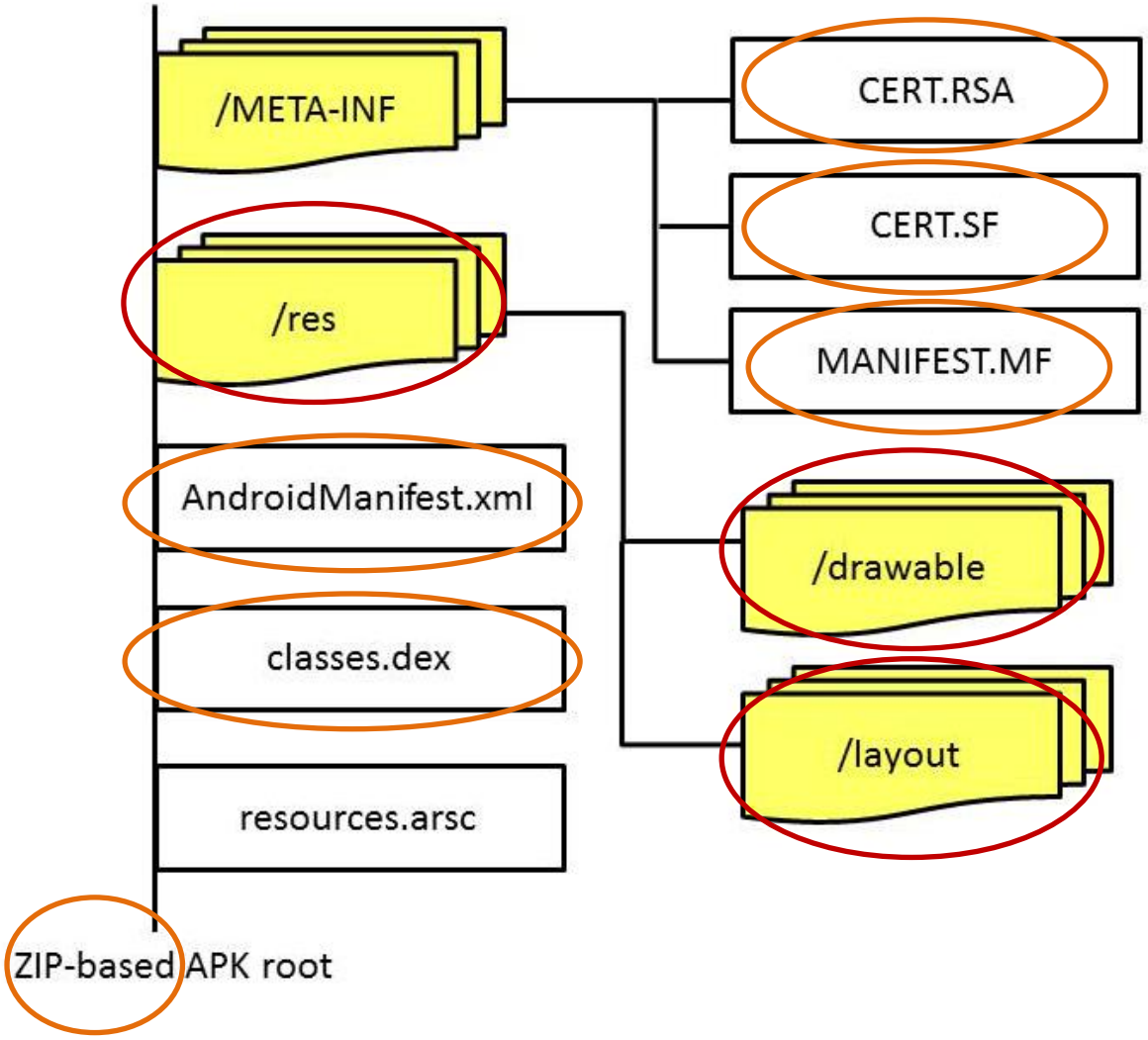
Relatively primitive obfuscation

Previously…

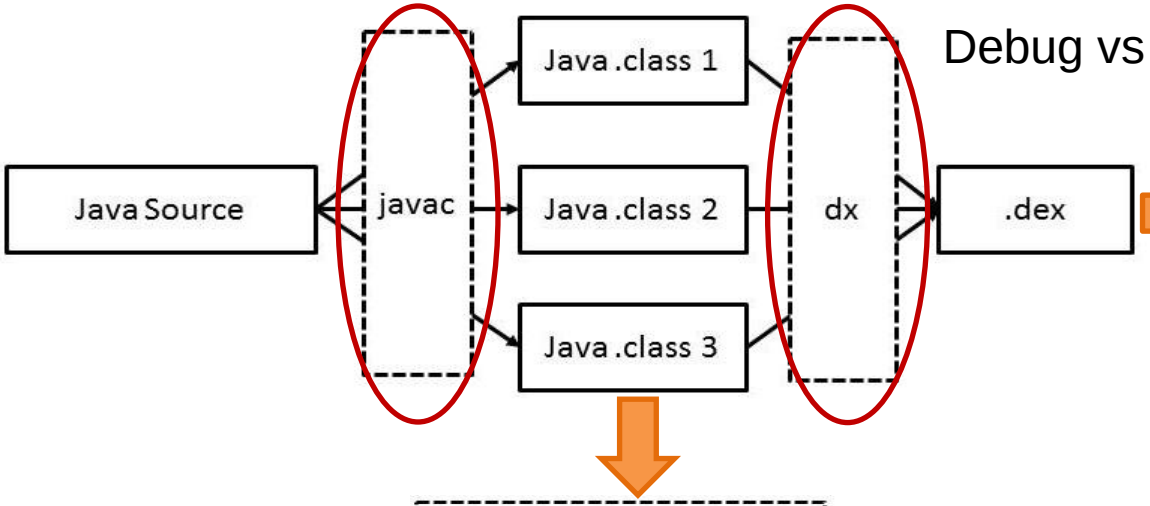
Metamorphosis

Windows style…

Typical APK Structure



Obfuscating Classes.dex via the Build

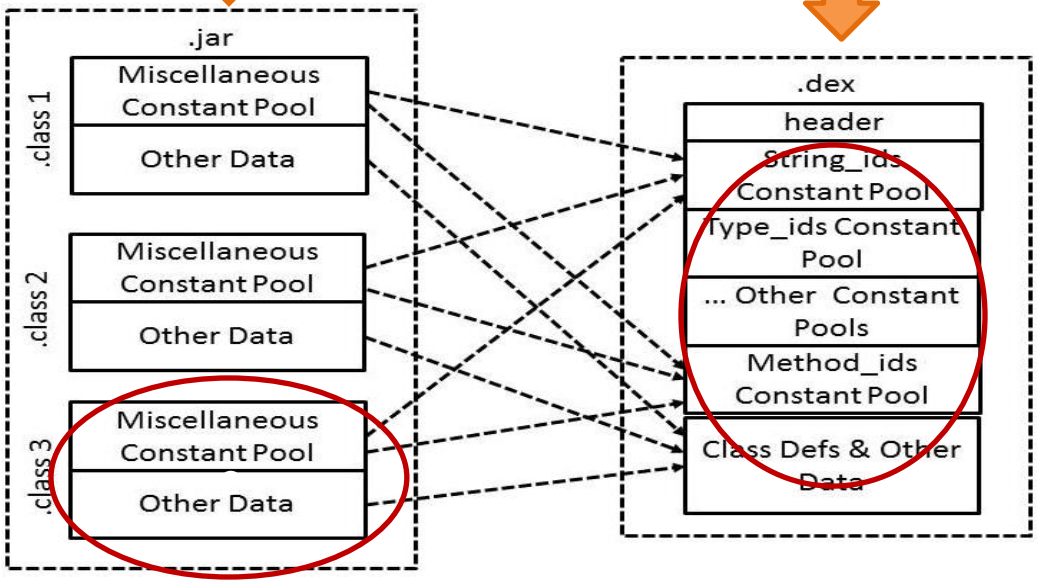


Debug vs Release

Build automation possible via Ant framework

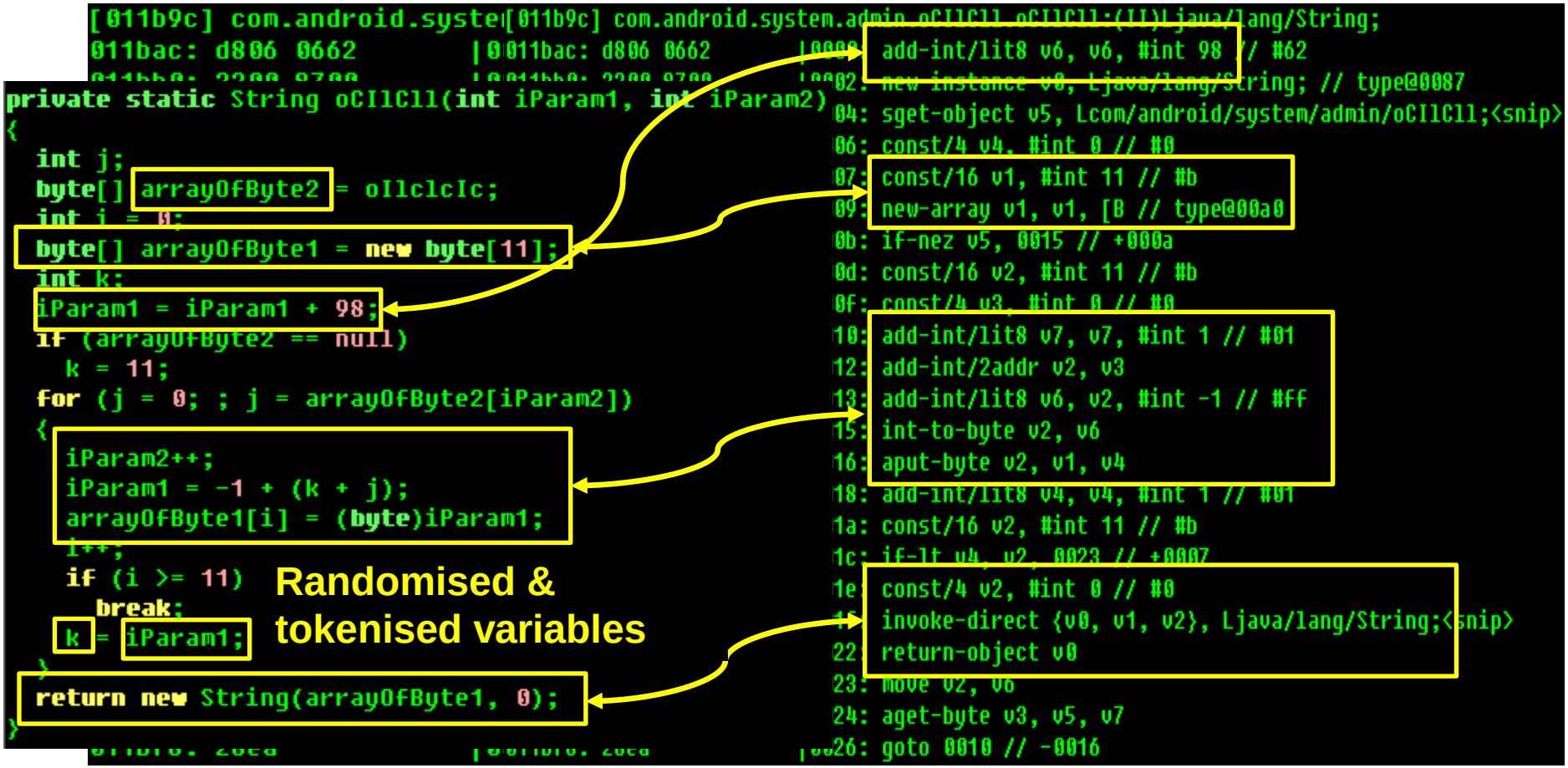
Induc-like code insertion?

Dummy dependencies/ classes



String Manipulation: Android.Obad.A

String Decryptor



Proguard

K7 COMPUTING™

"... obfuscates your code by removing unused code and renaming classes, fields and methods with semantically obscure names. The result is a smaller sized .apk file that is more difficult to reverse engineer"

- http://developer.android.com/tools/help/proguard.html

- Classes: `.myclass(es) -> .a(,b,c,d,...)`
- Methods & variables: explicit string references stripped

Massaging the DEX Header

Strong content validation

No timestamp

No unused metadata fields

Entry	Type	Description
Signature	BYTE[0x8]	DEX_FILE_MAGIC. Currently "dex\n035\0"
Checksum	DWORD	adler32 data corruption check
Hash	BYTE[0x14]	SHA1 data security check
File_size	DWORD	Size of entire file on disk
Header_size	DWORD	Currently 0x70
Endianess_flag	DWORD	0x12345678 (little-endian) or 0x78563412 (big-endian)
Link_size	DWORD	Link section size, or 0 if not statically-linked
Link_raw_offset	DWORD	Offset from start of file or 0 if not statically-linked
Map_raw_offset	DWORD	Offset from start of file to map item, 0 if absent
String_ids_sz_off_pair	{ DWORD, DWORD }	Size and location from start of file to string id list
Type_ids_sz_off_pair	{ DWORD, DWORD }	Size and location from start of file to type id list
Proto_ids_sz_off_pair	{ DWORD, DWORD }	Size and location from start of file to prototype id list
Field_ids_sz_off_pair	{ DWORD, DWORD }	Size and location from start of file to field id list
Method_ids_sz_off_pair	{ DWORD, DWORD }	Size and location from start of file to method id list
Class_def_sz_off_pair	{ DWORD, DWORD }	Size and location from start of file to class definition list
Data_sz_off_pair	{ DWORD, DWORD }	Size (aligned) and location from start of file to data section

DEX Byte-code Obfuscation: Nop

Original byte-code

```
0009e0: 0000                |0000: nop // spacer ******
0009e2: 1a01 0000           |0001: const-string v1, ""
0009e6: 1200                |0003: const/4 v0, #int 0
0009e8: 0000                |0004: nop // spacer ******
0009ea: 6e10 2800 0700      |0005: invoke-virtual {v7},
Ljava/lang/String;.length:()I
0009f0: 0a04                |0008: move-result v4
```

DEX Byte-code Obfuscation: Goto

Original byte-code

Obfuscated byte-code

DEX Byte-code Obfuscation: Move & Binop

Original byte-code

```
000a14: 1309 3a00 |001a: const/16 v9, #int 58 // #3a
000a18: d098 4000 |001c: add-int/lit16 v8, v9, #int 64 // #40
000a1c: 0184      |001e: move v4, v8
000a1e: 3642 0a00 |001f: if-gt v2, v4, 0029 // +000a
000a22: e006 0208 |0021: shl-int/lit8 v6, v2, #int 8 // #08
000a26: d766 0090 |0023: xor-int/lit16 v6, v6, #int -28672 //
#9000
000a2a: d762 00b0 |0025: xor-int/lit16 v2, v6, #int -20480 //
#b000
000a2e: e102 0208 |0027: shr-int/lit8 v2, v2, #int 8 // #08
000a32: 8e23      |0029: int-to-char v3, v2
```

DEX Byte-code Obfuscation: Invoke

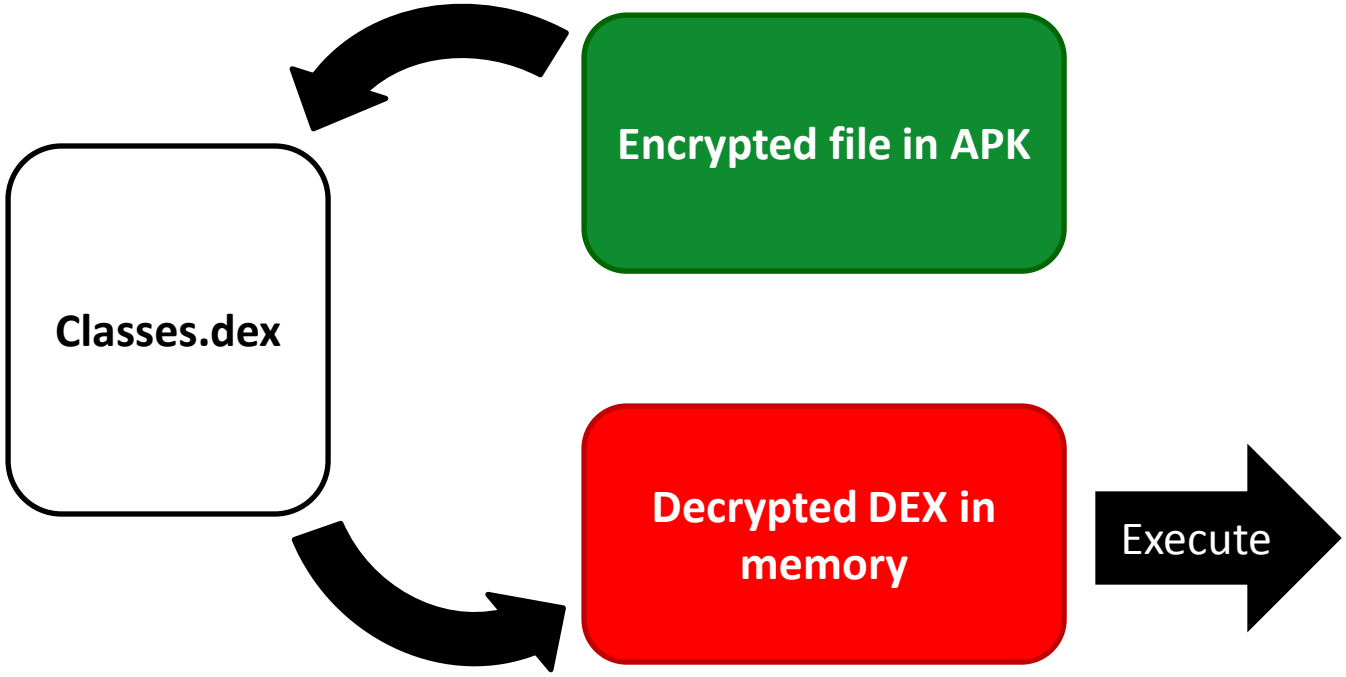
Dynamic method call via reflection

```
import java.lang.reflect.Method;
//Some other directives here
try {
Class cls = Class.forName("com.some.ClassName");
Object obj = cls.newInstance();
//Inspect class for method and invoke
Method mthd = cls.getDeclaredMethod(decr_mthd_str, null);
mthd.invoke(obj, null);
```

DEX Byte-code Obfuscation via JNI

Source: ‘Code Protection in Android’ – Schulz, uni-bonn.de

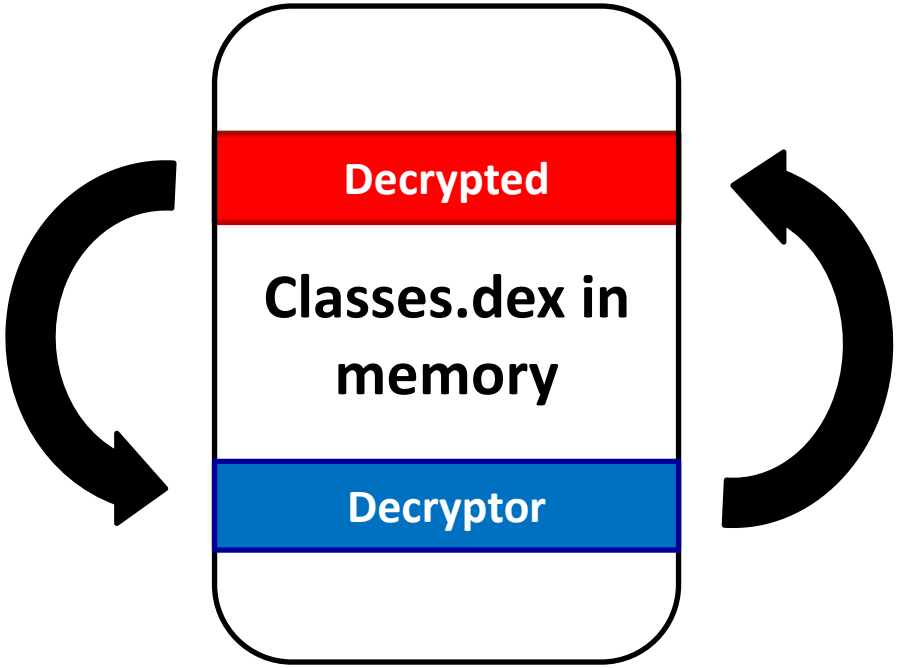
The Dropper



DEX Byte-code Obfuscation via JNI

Source: ‘Code Protection in Android’ – Schulz, uni-bonn.de

The Self-modifier



Detection Strategies

Checksums will not work very well

But judicious parsed object matching possible

Cloud scan: time lag and confidentiality issues

Dexopt: updated to flag non-standard code/data

IEEE Taggant System: Dexguard, Arxan, etc

But what about free Proguard?

K7 COMPUTING™

Queries