Nifty stuff that you can still do with Android

Xavier 'xEU' Martin

HES 2013

May 2th 2013

Thank You!

- This presentation is a compilation of original research done by the following people:
  - Tim Strazzere (Black Hat USA 2012)
  - Patrick Schulz

Speaker's bio

- Bootstrapping Immunapp

  Immunapp is a developer library and a SaaS dashboard that helps Android app developers fight against rampant malware which are repackaged versions of legitimate apps.

- Past work

  RE'ed Video game console : Dreamcast, PlayStation2.

  Code & tools were used in Code Breaker (cheat device)

Outline

- Android architecture
- Dynamic DEX file loading
- Self modifying Dalvik bytecode

Android

- Applications are (mostly) written in Java
- Classes are merged onto a single file, suitable for Dalvik virtual machine
- Deployed in APK file
  - AndroidManifest.xml : describe application (package name, components, required permissions, compatibility level)
  - Certificate, digests of files
  - Assets : image, video, audio
  - Native code : .so libraries
  - Classes.dex : Dalvik virtual machine

classes.dex – Dalvik EXecutable

- Application sourcecode : java
- Compiled onto regular .class files

- Android specific steps
  - Merge multiple classes onto a single file
  - Convert bytecode, from stack machines (JVM) to register-based architecture (Dalvik)

DexClassLoader

- Application wants to load another dex file
  - Legitimate usage : in-app purchase
  - Abuse : from Command&Control server

- API
  - DexClassLoader(String dexPath, String optimizedDirectory, String libraryPath, ClassLoader parent)
  - Needs dex file on disk

Under the hood

- Dalvik internals: dvm_dalvik_system_DexFile

```
const DalvikNativeMethod dvm_dalvik_system_DexFile[] = {
    { "openDexFile",        "(Ljava/lang/String;Ljava/lang/String;I)I",
        Dalvik_dalvik_system_DexFile_openDexFile },
    { "openDexFile",        "([B)I",
        Dalvik_dalvik_system_DexFile_openDexFile_bytearray },
    { "closeDexFile",       "(I)V",
        Dalvik_dalvik_system_DexFile_closeDexFile },
    { "defineClass",        "(Ljava/lang/String;Ljava/lang/ClassLoader;I)Ljava/lang/Class;",
        Dalvik_dalvik_system_DexFile_defineClass },
    { "getClassNameList",   "(I)[Ljava/lang/String;",
        Dalvik_dalvik_system_DexFile_getClassNameList },
    { "isDexOptNeeded",     "(Ljava/lang/String;)Z",
        Dalvik_dalvik_system_DexFile_isDexOptNeeded },
    { NULL, NULL, NULL },
};
```

dvm_dalvik_system_DexFile

- Application must use native code (JNI, .so library)

- OnLoad method + dlsym

```
JNINativeMethod *dvm_dalvik_system_DexFile;

JNIEXPORT jint JNI_OnLoad(JavaVM* vm, void* reserved) {

        void *ldvm = (void*)dlopen("libdvm.so", RTLD_LAZY);

        dvm_dalvik_system_DexFile = (JNINativeMethod*)dlsym(ldvm, "dvm_dalvik_system_DexFile");
```

OpenDexFile

- From dvm_dalvik_system_DexFile
  - Find matching name
  - Check for correct signature ([B)I
  - Get pointer

OpenDexFile

```
void (*openDexFile)(const u4* args, JValue* pResult);
lookup(openDexFile, "dvm_dalvik_system_DexFile", "([B)I", &openDexFile)

int lookup (JNINativeMethod *table, const char *name, const char *sig, void (**fnPtrout)
(u4 const *, union JValue *)) {
  int i = 0;
  while (table[i].name != NULL)  {
    if ( (strcmp(name, table[i].name) == 0) &&  (strcmp(sig, table[i].signature) == 0) ) {
      *fnPtrout = table[i].fnPtr;
      return 1;
    }
    i++;
  }
  return 0;
}
```

OpenDexFile

- Invoke

```
ArrayObject *ao;     // header+dex content
u4 args[] = { (u4)ao };
JValue pResult ;
jint result ;
openDexFile(args, &pResult);
result = (jint)pResult.l;
return result;
```

Under the hood

- Dalvik internals: dvm_dalvik_system_DexFile

```
const DalvikNativeMethod dvm_dalvik_system_DexFile[] = {
    { "openDexFile",       "(Ljava/lang/String;Ljava/lang/String;I)I",
        Dalvik_dalvik_system_DexFile_openDexFile },
    { "openDexFile",       "([B)I",
        Dalvik_dalvik_system_DexFile_openDexFile_bytearray },
    { "closeDexFile",      "(I)V",
        Dalvik_dalvik_system_DexFile_closeDexFile },
    { "defineClass",       "(Ljava/lang/String;Ljava/lang/ClassLoader;I)Ljava/lang/Class;",
        Dalvik_dalvik_system_DexFile_defineClass },
    { "getClassNameList",  "(I)[Ljava/lang/String;",
        Dalvik_dalvik_system_DexFile_getClassNameList },
    { "isDexOptNeeded",    "(Ljava/lang/String;)Z",
        Dalvik_dalvik_system_DexFile_isDexOptNeeded },
    { NULL, NULL, NULL },
};
```

Dex Loading

- getClassNameList (I)[Ljava/lang/String;
  - List of classes available from loaded dex
- defineClass (Ljava/lang/String;Ljava/lang/ClassLoader;I)Ljava/lang/Class;
  - Oddity : expect / as separator (com.a.b.c.d => com/a/b/c/d)

Dex Loading

```
int cookie = openDexFile(...);
Class<?> cls = null;
String as[] = getClassNameList(cookie);
for(int z=0; z<as.length; z++) {
    if(as[z].equals("com.immunapp.hes2013.MainActivity")) {
      cls=defineClass(as[z].replace('.', '/'), context.getClassLoader(), cookie );
    } else {
      defineClass(as[z].replace('.', '/'), context.getClassLoader(), cookie );
    }
}
if(cls!=null) {
  Intent intent = new Intent(this, newcls);
  startActivity(intent);
}
```

Self modifying Dalvik Bytecode

- JNI again
  - /proc/self/maps

```
49143000-49145000 r--s 00003000 1f:01 1013       /data/app/com.immunapp.hes2013.bc-1.apk

49145000-49146000 r--s 0003f000 1f:01 1013       /data/app/com.immunapp.hes2013.bc-1.apk

49146000-491b5000 r--p 00000000 1f:01 857
/data/dalvik-cache/data@app@com.immunapp.hes2013.bc-1.apk@classes.dex

491b5000-491be000 rw-p 00000000 00:07 14251      /dev/ashmem/dalvik-aux-structure (deleted)

491bf000-491c6000 r-xp 00000000 1f:01 837
/data/app-lib/com.immunapp.hes2013.bc-1/libdextest.so
```

Self modifying Dalvik Bytecode

- Search in memory : look for DEX signature

  dex\n035

- It'll be aligned on _SC_PAGESIZE, at offset 0x28

Self modifying Dalvik Bytecode

- DEX is found : easy part

- Parse it

  https://source.android.com/tech/dalvik/dex-format.html

Self modifying Dalvik Bytecode

- DEX header
  - String table
  - Method table
  - Class Def table

Self modifying Dalvik Bytecode

- DEX format

Variable-length quantity, ULEB128

127	0x7F
128	0x80 0x01

Strings : MUTF-8 (Modified UTF-8) Encoding

Self modifying Dalvik Bytecode

- Finding the right place
  - 1st pass : search class
  - 2nd pass : look for your method

- encoded_method
  - code_off   uleb128
  - offset from the start of the file to the code structure for this method, or 0 if this method is either abstract or native.
  - The offset should be to a location in the data section.

Self modifying Dalvik Bytecode

- bytecode
  - insns ushort[insns_size] (offset 0x10)
  - actual array of bytecode, described in a document "Bytecode for the Dalvik VM".

Self modifying Dalvik Bytecode

- Unlock memory

  Align address to closest _SC_PAGESIZE

  ```
  mprotect((unsigned char*)aligned,
  PROT_WRITE | PROT_READ, len);
  ```

- Insert your payload

  ```
  memcpy((unsigned char*)code_off,
  opcodes, len);
  ```

Self modifying Dalvik Bytecode

- Unlock memory

  Align address to closest _SC_PAGESIZE

  ```
  mprotect((unsigned char*)aligned,
  PROT_WRITE | PROT_READ, len);
  ```

- Insert your payload

  ```
  memcpy((unsigned char*)code_off,
  opcodes, len);
  ```

Self modifying Dalvik Bytecode

- Sample

```
public static int dummyMethod() {
    return 42;
    // bytecode:
    /*
    13 00 2A 00        const/16     v0, 0x2A
    0F 00              return v0
    */
}
```

Self modifying Dalvik Bytecode

- sample

```
native static int searchDex();


native static int patchDex(int addr,
String methodName, byte[] opcode);
```

Self modifying Dalvik Bytecode

- sample

```
int dexInMemory = searchDex();

patchDex(dexInMemory, "dummyMethod",
new byte[] { 0x13, 0x00, 0x55, 0x00,
0x0F, 0x00 });

int r = dummyMethod();
Log.d("dummy()", ""+r);
```

Self modifying Dalvik Bytecode

```
I/bytecode( 9205): 49145000-49146000 r--s 00034000 1f:01 1759       /data/app/com.example.sample4-1.apk
I/bytecode( 9205): 49146000-491b5000 r--p 00000000 1f:01 1456
/data/dalvik-cache/data@app@com.immunapp.hes2013.bc-1.apk@classes.dex
I/bytecode( 9205): 491b5000-491be000 rw-p 00000000 00:07 123159     /dev/ashmem/dalvik-aux-structure (deleted)
I/bytecode( 9205): 491bf000-491c2000 r-xp 00000000 1f:01 1409
/data/app-lib/com.immunapp.hes2013.bc-1/libdextest.so
I/bytecode( 9205): 491c2000-491c3000 r--p 00002000 1f:01 1409
/data/app-lib/com.immunapp.hes2013.bc-1/libdextest.so
I/bytecode( 9205): 491c3000-491c4000 rw-p 00003000 1f:01 1409
/data/app-lib/com.immunapp.hes2013.bc-1/libdextest.so
I/bytecode( 9205): be95d000-be972000 rw-p befeb000 00:00 0          [stack]
I/bytecode( 9205): found at 49146000, dex at 49146028
I/bytecode( 9205): methodName=dummyMethod (11)
I/bytecode( 9205): opcodes length=6
I/bytecode( 9205): string_ids_size=00000fac
I/bytecode( 9205): string_ids_off=00000070
I/bytecode( 9205): method_ids_size=00000d81
I/bytecode( 9205): method_ids_off=000085d4
I/bytecode( 9205): method[3280] 000007fe
I/bytecode( 9205): class_defs_size=0000013f
I/bytecode( 9205): class_defs_off=0000f1dc
I/bytecode( 9205): found method[3280] at 0002b470 : 491714a8
I/bytecode( 9205): aligned page 49171000
I/bytecode( 9205): unlocked
I/bytecode( 9205): bytecode patched
D/dummy() ( 9205): 85
```