The AndBug Debugger

A Scriptable Debugger for Dalvik

Scott Dunlop, (sdunlop@ioactive.com)

The Devolution of Android Development

Taking Google's SDK From Making Apps to Breaking Apps.

Debugging with the Eclipse SDK (and Source)

- Primary focus of Dalvik's debugging support.
- Can't dump thread contexts.
- Requires source code.*

Debugging with the Eclipse / NetBeans (with Apktool)

- Tricking Eclipse with synthetic Baksmali source.
- Large parts of Eclipse / NetBeans won't work
- Demands Smali literacy. :)

Debugging with JDB (with or without Source)

- Dalvik does not specifically support JDB.
- Not only do many parts not work, they crash the process.
- Clumsy shell; was considered a proof of concept.
- Doesn't require source, but many commands require it.

```
sdunlop @ slip :: ~
$ rlwrap jdb -attach 127.0.0.1:1141
Set uncaught java.lang.Throwable
Set deferred uncaught java.lang.Throwable
Initializing jdb ...
> threads
Group system:
  (java.lang.Thread)0xc14050d848 <6> Compiler          cond. waiting
  (java.lang.Thread)0xc14050d6d8 <4> Signal Catcher    cond. waiting
  (java.lang.Thread)0xc14050d630 <3> GC                cond. waiting
  (java.lang.Thread)0xc14050d578 <2> HeapWorker        cond. waiting
Group main:
  (java.lang.Thread)0xc14001f1a8 <1> main              running
  (java.lang.Thread)0xc14050f6b0 <8> Binder Thread #2 running
  (java.lang.Thread)0xc14050f5e8 <7> Binder Thread #1 running
```

Common Elements of Dalvik Debuggers

Same Protocols, Same Processes, Different Interfaces.

The Android SDK

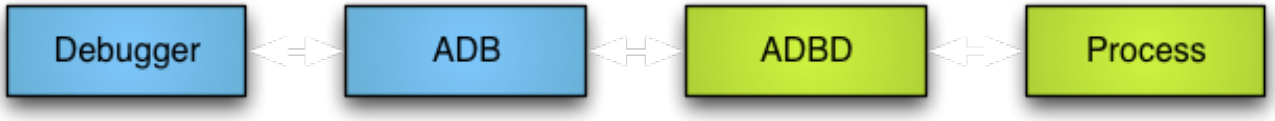
- Many small tools wrapped by "Android" and Eclipse.
- Includes a QEMU-derived emulator.
- Can use TCP/IP or USB as a bridge to device or emulator.
- Key dependency of ANY Android debugger.

The Dalvik Virtual Machine

- A Java-like VM for low power hardware.*
- Uses a register IL, instead of Sun's stack IL.
- Newer versions support Just in Time compilation.

The Java Debug Wire Protocol (JDWP)

- Very high level debugging API for Java.
- Asynchronous packets, send requests, hope for responses.
- Closely related to the Java Native Interface.
- Dalvik explicitly only supports parts of JDWP Eclipse uses.



The AndBug Debugger

Cannibalizing the SDK to Make an RE-friendly Debugger

AndBug From the Command Line

- Offers a number of CLI utilities.
- Can dump loaded classes, methods, thread state, static attributes.
- Can act as an API tracer (strace for Java classes).

The AndBug Shell

- An interactive debugger, similar to GDB or JDB.
- Can set breakpoints, suspend and resume execution.
- Includes all of the commands from the CLI.

The AndBug Module

- AndBug is 90% Python, 10% C.
- Really designed to be as a package by custom tools.

```
@andbug.command.action('<class-path>', aliases=('ct', 'ctrace'))
def class_trace(ctxt, cpath):
    'reports calls to dalvik methods associated with a class'
    cpath = andbug.options.parse_cpath(cpath)

    with andbug.screed.section('Setting Hooks'):
        for c in ctxt.sess.classes(cpath):
            c.hookEntries(func = report_hit)
            andbug.screed.item('Hooked %s' % c)

    ctxt.block_exit()
```

Navi -- The AndBug Process Browser

- Normally used from the shell or CLI.
- Starting from thread state, can browse the stack, its objects, their attributes, and on and on..
- Requires a suspended process.

End-Matter

- Thanks to Google, JesusFreke and Brut.all -- Google must love RE's.. They gave us dexdump.
- Source code is available at https://github.com/swdunlop/andbug.
- No, I will not be doing this again for Blackberry.