# Whole-System Dynamic Binary Analysis

Continued

# Repeatable Reverse Engineering for the Greater Good with PANDA

Brendan Dolan-Gavitt, Josh Hodosh, Patrick Hulin,
Tim Leek, Ryan Whelan

# PANDA’s Workflow

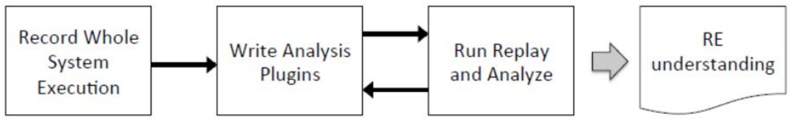



Fig. 1: Replay-based Reverse Engineering Workflow. PANDA's ability to record and replay whole system executions is the foundation of its use in reverse engineering. One captures a recording and then iteratively builds one or more plugins that perform dynamic analyses.

# Record&Replay Internals

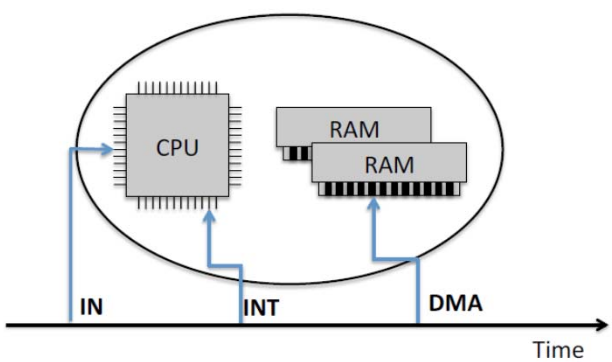



Fig. 2: PANDA Non-determinism log

# Log sizes for various replays

| Replay | Instructions | Log size | Instr/byte |
|---|---|---|---|
| freebsdboot.rr | 9.3B | 533MB | 17 |
| spotify.rr | 12B | 229MB | 52 |
| haikuurl.rr | 8.6B | 119MB | 72 |
| carberp1.rr | 9.1B | 43MB | 212 |
| win7iessl.rr | 8.6B | 9.4MB | 915 |
| Starcraft.rr | 60M | 1.8MB | 33 |

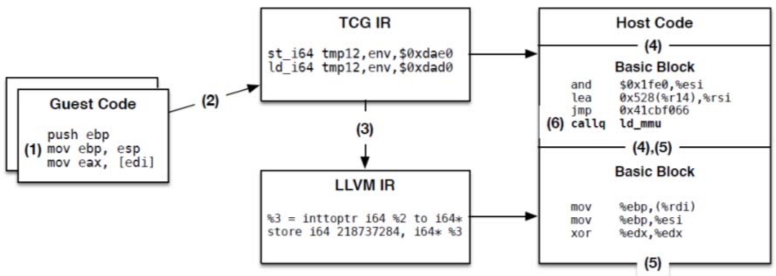
TABLE I: ND log sizes for various replays

# Record and Replay Runtime Overheads

| Environment | Time in sec | Slowdown wrt Qemu 2.1.0 |
|---|---|---|
| Qemu 2.1.0 | 35.6 | 1.0 |
| PANDA | 37.2 | 1.05 |
| PANDA+record | 66 | 1.85 |
| PANDA+replay | 127 | 3.57 |

TABLE II: PANDA, record, and replay slowdowns

# PANDA execution and instrumentation




# PANDA Plugins

- Tappan Zee (North) Bridge
- System Calls
- Shadow Callstack
- Taint Analysis
- Scissors

# DroidScope:
## Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis

Lok Yan
Heng Yin
August 10, 2012

L.C.Smith College of Engineering and Computer Science

# Android

Java Components

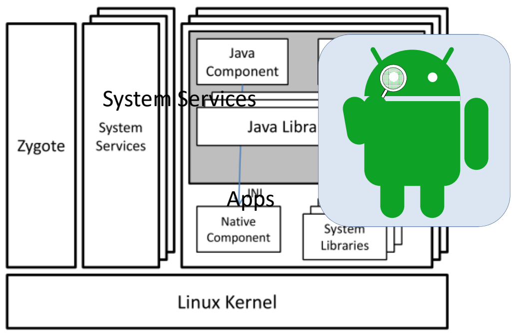
System Services

Native Components

Apps

10

# Android

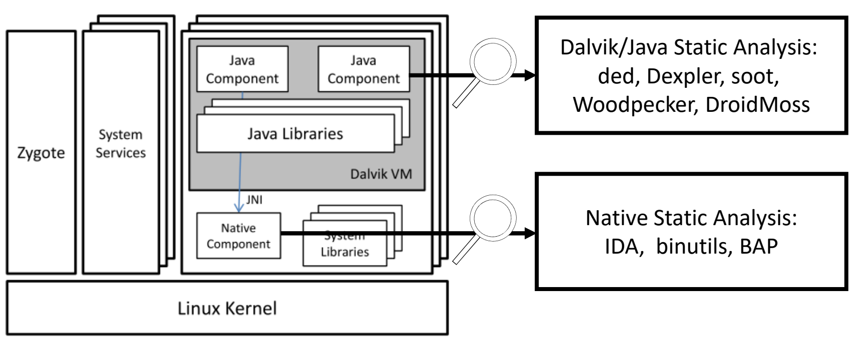



Java Components

Native Components

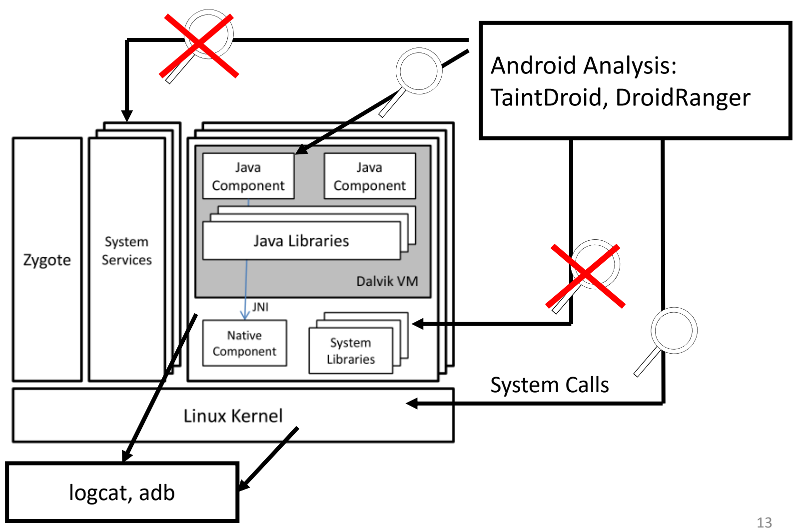
11

# Motivation: Static Analysis



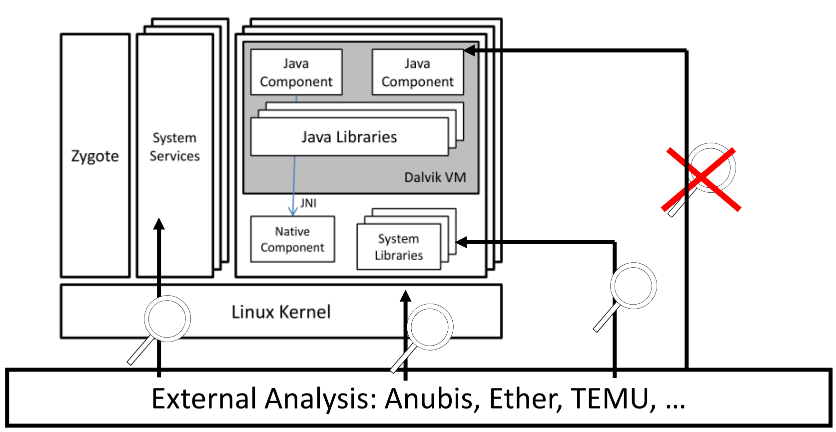

12

# Motivation: Dynamic Analysis



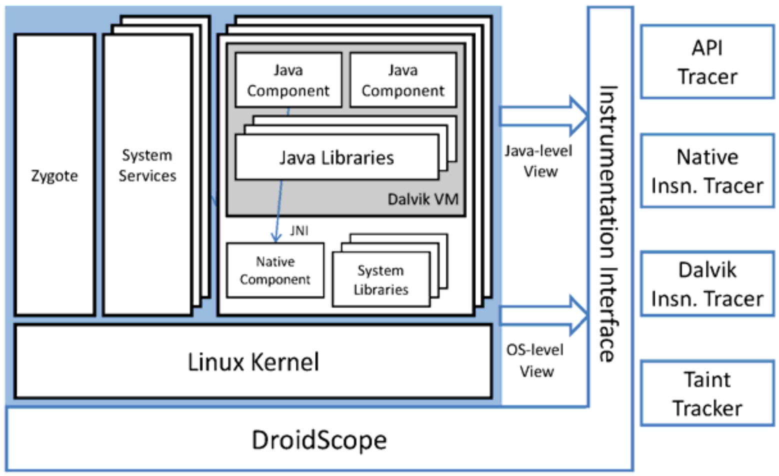

13

# Motivation: Dynamic Analysis




14

# DroidScope Overview




15

# Goals

- Dynamic binary instrumentation for Android
  - Leverage Android Emulator in SDK
  - No changes to Android Virtual Devices
  - External instrumentation
    - Linux context
    - Dalvik context
  - Extensible: plugin-support / event-based interface
  - Performance
    - Partial JIT support
    - Instrumentation optimization

16

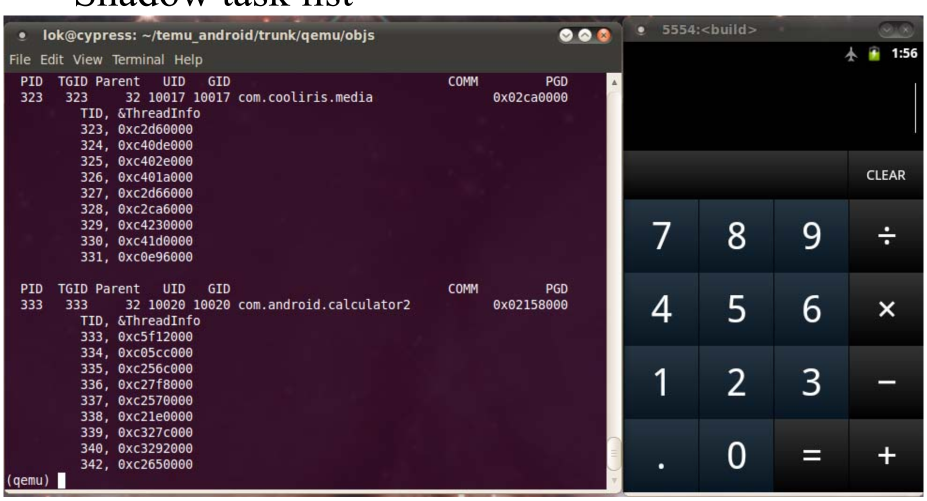
# Roadmap

- External instrumentation
  - Linux context
  - Dalvik context
- Extensible: plugin-support / event-based interface
- Evaluation
  - Performance
  - Usage

17

# Linux Context: Identify App(s)

- Shadow task list



# Java/Dalvik View

- Dalvik virtual machine
  - register machine (all on stack)
  - 256 opcodes
  - saved state, *glue*, pointed to by ARM R6, on stack in x86
- mterp
  - offset-addressing: *fetch opcode* then jump to *(dvmAsmInstructionStart + opcode * 64)*
  - *dvmAsmSisterStart* for emulation overflow
- Which Dalvik opcode?
  1. Locate dvmAsmInstructionStart in shadow memory map
  2. Calculate opcode = (R15 - dvmAsmInstructionStart) / 64.

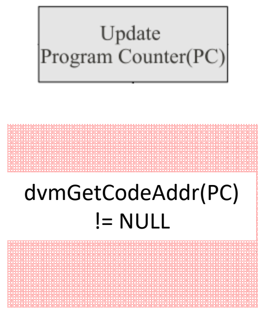
19

# Just In Time (JIT) Compiler

- Designed to boost performance
- Triggered by counter - mterp is always the default
- Trace based
  - Multiple basic blocks
  - Multiple exits or *chaining cells*
  - Complicates external introspection
  - Complicates instrumentation

20

# Disabling JIT






# Roadmap

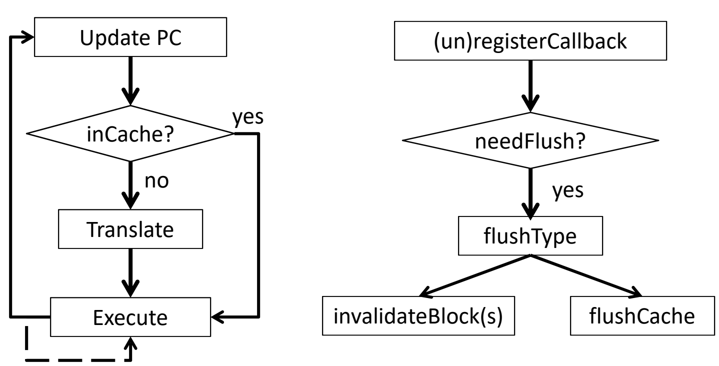
- ✓ External instrumentation
  - Linux context
  - Dalvik context
- ➢ Extensible: plugin-support / event-based interface
- Evaluation
  - Performance
  - Usage

# Instrumentation Design

- Event based interface
  - Execution: e.g. native and Dalvik instructions
  - Status: updated shadow task list
- Query and Set, e.g. interpret and change cpu state
- Performance
  - Example: Native instructions vs. Dalvik instructions
  - Instrumentation Optimization

23

# Dynamic Instrumentation




24

# Instrumentation

| | NativeAPI | LinuxAPI | DalvikAPI |
|---|---|---|---|
| Events | instruction begin/end | context switch | Dalvik instruction begi |
| | register read/write | system call | method begin |
| | memory read/write | task begin/end | |
| | block begin/end | task updated | |
| | | memory map updated | |
| Query & Set | memory read/write | query symbol database | query symbol database |
| | memory r/w with pgd | get current context | interpret Java object |
| | register read/write | get task list | get/set DVM state |
| | taint set/check | | taint set/check objects |
| | | | disable JIT |

25

# Dalvik Instruction Tracer (Example)

```
void opcode_callback(uint32_t opcode) {
  printf("[%x] %s\n", GET_RPC, opcodeToStr(opcode));
}

void module_callback(int pid) {
  if (bInitialized || (getIBase(pid) == 0))
    return;


  getModAddr("dfk@classes.dex", &startAddr, &endAddr);
  addDisableJITRange(pid, startAddr, endAddr);
  disableJITInit(getGetCodeAddrAddress(pid));
  addMterpOpcodesRange(pid, startAddr, endAddr);
  dalvikMterpInit(getIBase(pid));
  registerDalvikInsnBeginCb(&opcode_callback);
  bInitialized = 1;
}

void _init() {
  setTargetByName("com.andhuhu.fengyinchuanshuo");
  registerTargetModulesUpdatedCb(&module_callback);
}
```

26

# Plugins

- API Tracer
  - System calls
    - *open, close, read, write,* includes parameters and return values
  - Native library calls
  - Java API calls
    - Java Strings converted to C Strings
- Native and Dalvik Instruction Tracers
- Taint Tracker
  - Taints ARM instructions
  - One bit per byte
  - Data movement & Arithmetic instructions including barrel shifter
  - Does not support control flow tainting

27

# Roadmap

✓ External instrumentation
  - Linux context
  - Dalvik context

✓ Extensible: plugin-support / event-based interface

➢ Evaluation
  - Performance
  - Usage

28

# Implementation

- Configuration
  - QEMU 0.10.50 – part of Gingerbread SDK
  - Gingerbread
    - “user-eng”
    - No changes to source
  - Linux 2.6.29, QEMU kernel branch

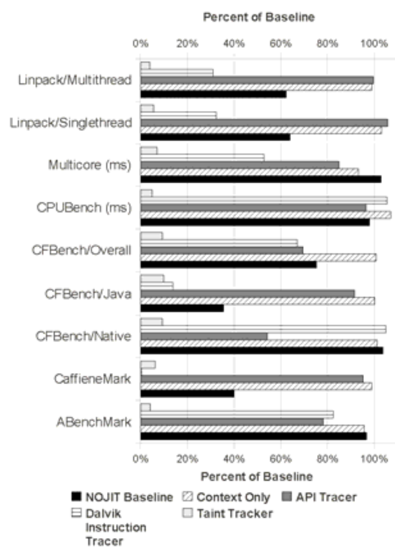
29

# Performance Evaluation

- Seven free benchmark Apps
  - AnTuTu Benchmark
  - (ABenchMark) by AnTuTu
  - CaffeineMark by Ravi Reddy
  - CF-Bench by Chainfire
  - Mobile processor benchmark (Multicore) by Andrei Karpushonak
  - Benchmark by Softweg
  - Linpack by GreeneComputing
- Six tests repeated five times each
  - Baseline
  - NO-JIT Baseline – uses a build with JIT disabled at runtime
  - Context Only
  - API Tracer
  - Dalvik Instruction Trace
  - Taint Tracker

30

# Select Performance Results




APITracer vs. NOJIT

Results are not perfect

Dynamic Symbol Retrieval Overhead

31

# Usage Evaluation

- Use DroidScope to analyze real world malware
  - API Tracer
  - Dalvik Instruction Tracer + dexdump
  - Taint Tracker – taint IMEI/IMSI @ *move_result_object* after *getIMEI/getIMSI*
- Analyze included exploits
  - Removed patches in Gingerbread
  - Intercept system calls
  - Native instruction tracer

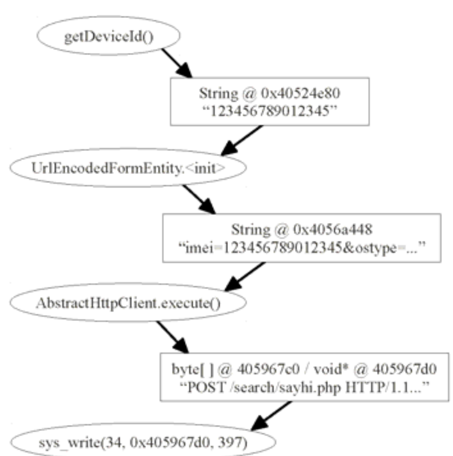
32

# Droid Kung Fu

- Three encrypted payloads
  - ratc (Rage Against The Cage)
  - killall (ratc wrapper)
  - gjsvro (udev exploit)
- Three execution methods
  - piped commands to a shell (default execution path)
  - Runtime.exec() Java API (instrumented path)
  - JNI to native library terminal emulator (instrumented path)
  - Instrumented return values for *isVersion221* and *getPermission* methods

33

# Droid Kung Fu: TaintTracker



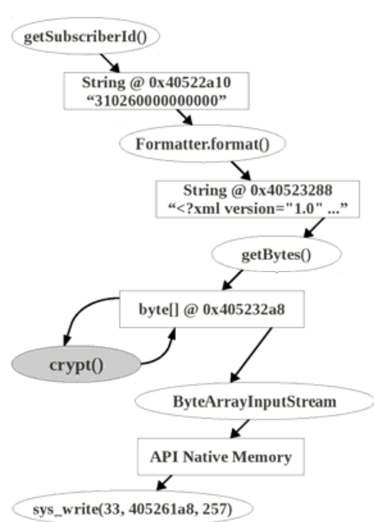

34

# DroidDream

- Same payloads as DroidKungFu
- Two processes
  - Normal *droiddream* process clears logcat
  - *droiddream:remote* is malicious
- xor-encrypts private information before leaking
- Instrumented *sys_connect* and *sys_write*

35

# Droid Dream: TaintTracker




36

# DroidDream: crypt trace

```
[43328f40] aget-byte v2(0x01), v4(0x405232a8), v0(186)
  Getting Tainted Memory: 40523372(2401372)
  Adding M@410accec(42c5cec) len = 4
[43328f44] sget-object v3(0x0000005e), KEYVALUE// field@0003
[43328f48] aget-byte v3(0x88), v3(0x4051e288), v1(58)
[43328f4c] xor-int/2addr v2(62), v3(41)
  Getting Tainted Memory: 410accec(42c5cec)
  Adding M@410accec(42c5cec) len = 4
[43328f4e] int-to-byte v2(0x17), v2(23)
  Getting Tainted Memory: 410accec(42c5cec)
  Adding M@410accec(42c5cec) len = 4
[43328f50] aput-byte v2(0x17), v4(0x405232a8), v0(186)
  Getting Tainted Memory: 410accec(42c5cec)
  Adding M@40523372(2401372) len = 1
```

37

# Summary

- DroidScope
  - Dynamic binary instrumentation for Android
  - Built on Android Emulator in SDK
  - External Introspection & Instrumentation support
  - Four plugins
    - API Tracer
    - Native Instruction Tracer
    - Dalvik Instruction Tracers
    - TaintTracker
  - Partial JIT support

38

# Related Works

- Static Analysis
  - ded, Dexpler, soot
  - Woodpecker, DroidMoss
- Dynamic Analysis
  - TaintDroid
  - DroidRanger
  - PIN, Valgrind, DynamoRIO
  - Anubis, TEMU, Ether, PinOS
- Introspection
  - Virtuoso
  - VMWatcher

39

# Challenges

- JIT
  - Full JIT support
  - Flushing JIT cache
- Emulation detection
  - Real Sensors: GPS, Microphone, etc.
  - Bouncer
- Timing assumptions, timeouts, events
- Closed source systems, e.g. iOS

40

# Questions?

Q0. Where can I get DroidScope?

L.C.Smith College of Engineering and Computer Science

## ratc

- Vulnerability
  - *setuid()* fails when RLIMIT_NPROC reached
  - *adbd* fails to verify *setuid()* success
- Three generation (stage) exploit
  - Locate *adbd* in */proc* and spawns child
  - Child *fork()* processes until -*11 (-EAGAIN)* is returned then spawns child – continues *fork()*
  - Grandchild *kill() adbd* and waits for process to re-spawn

42

# ratc: exploit diagnosis

```
  ;;;setgid returns from kernel back to adbd
  0000813c: pop {r4, r7}
  00008140: movs r0, r0
  00008144: bxpl lr : Read Oper[0]. R14, Val = 0xc3a5
  ;; Return back to 0xc3a4 (caller) in Thumb mode

;;;adbd_main sets up for setuid
0000c3a4: movs r0, #250
0000c3a6: lsls r0, r0, #3 : Write Oper[0]. R0, Val = 0x7d0
  ;; 250 * 8 = 0x7d0 = 2000 = AID_SHELL

  ...

  ;;;Start of setuid section
  ;;;  213 is syscall number for sys_setuid
  00008be0: push {r4, r7} : Write Oper[0]. M@be910bb8, Val = 0x7d0
    ;; push AID_SHELL onto the stack
  00008be4: mov r7, #213
  00008be8: svc 0x00000000
    ;; Make sys call

    ;;; === TRANSITION TO KERNEL SPACE ===

    ;;;sys_setuid then calls set_user in kernel mode

      ;;;inside sys_setuid
      ;; Has rlimit been reached?
      c0048944: cmp r2, r3  : Read Oper[0]. R3, Val = 300 Read Oper[1]. R2, Val = 300

      ;;; RLIMIT(300) is reached and !init_user so return -11
      c0048960: mvn r0, #10 : Write Oper[0]. R0, Val = 0xfffffff5
          ;; the return value is now -11 or -EAGAIN
      c0048964: ldmib sp, {r4, r5, r6, fp, sp, pc}

      ;;;Return back to sys_setuid which returns back to userspace

    ;;; === RETURN TO USERSPACE ===

  ;;;setuid continues
  00008bec: pop {r4, r7}
  00008bf0: movs r0, r0 : Read Oper[0]. R0, Val = 0xfffffff5
      ;; -11 is still here

;;;Return back to adb_main at 0xc3ac (the return address) above
;;; Immediately starts other work, does not check return code
0000c3ac: ldr r7, [pc, #356] : Read Oper[0]. M@0000c514, Val = 0x19980330
  Write Oper[0]. R7, Val = 0x19980330
    ;; 0x19980330 is _LINUX_CAPABILITY_VERSION
```

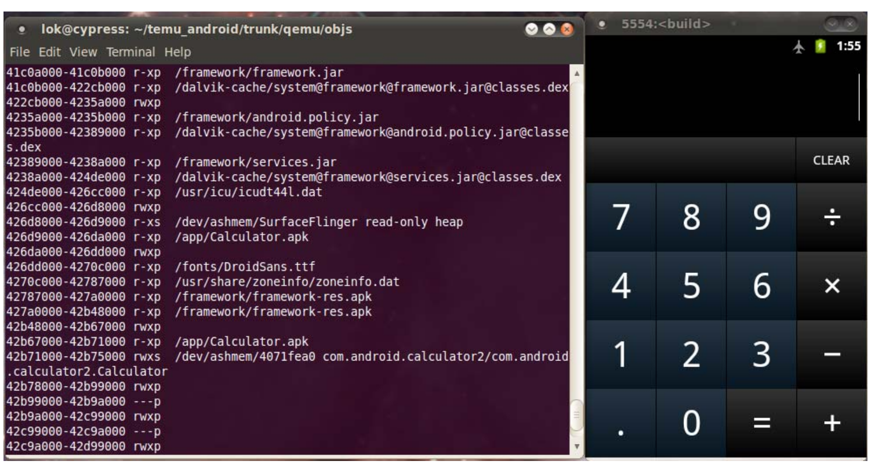
43

# Symbol Information

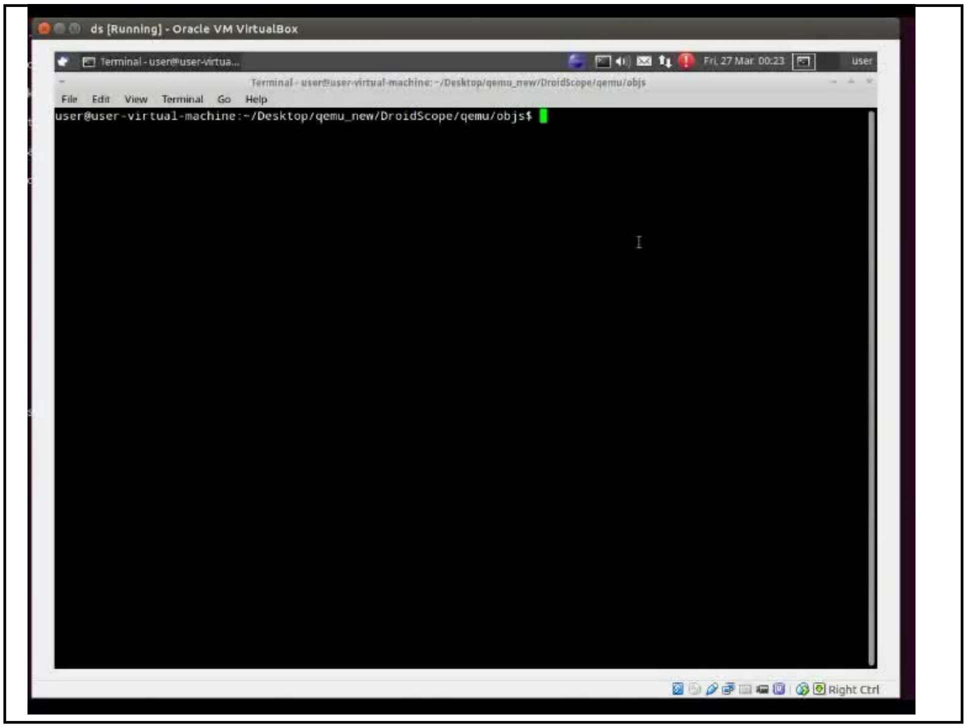
- Native library symbols - Static
  - From *objdump* of libraries
- Java symbols - Dynamic
  - Dalvik data structures -> address of string
  - Given address, load from
    - Memory
    - File mapped into memory
  - *dexdump* as backup

44

45


lok@cypress: ~/temu_android/trunk/qemu/objs
File Edit View Terminal Help
41c0a000-41c0b000 r-xp  /framework/framework.jar
41c0b000-422cb000 r-xp  /dalvik-cache/system@framework@framework.jar@classes.dex
422cb000-4235a000 rwxp
4235a000-4235b000 r-xp  /framework/android.policy.jar
4235b000-42389000 r-xp  /dalvik-cache/system@framework@android.policy.jar@classe
s.dex
42389000-4238a000 r-xp  /framework/services.jar
4238a000-424de000 r-xp  /dalvik-cache/system@framework@services.jar@classes.dex
424de000-426cc000 r-xp  /usr/icu/icudt44l.dat
426cc000-426d8000 rwxp
426d8000-426d9000 r-xs  /dev/ashmem/SurfaceFlinger read-only heap
426d9000-426da000 r-xp  /app/Calculator.apk
426da000-426dd000 rwxp
426dd000-4270c000 r-xp  /fonts/DroidSans.ttf
4270c000-42787000 r-xp  /usr/share/zoneinfo/zoneinfo.dat
42787000-427a0000 r-xp  /framework/framework-res.apk
427a0000-42b48000 r-xp  /framework/framework-res.apk
42b48000-42b67000 rwxp
42b67000-42b71000 r-xp  /app/Calculator.apk
42b71000-42b75000 rwxs  /dev/ashmem/4071fea0 com.android.calculator2/com.android
.calculator2.Calculator
42b78000-42b99000 rwxp
42b99000-42b9a000 ---p
42b9a000-42c99000 rwxp
42c99000-42c9a000 ---p
42c9a000-42d99000 rwxp
5554:<build>
1:55
CLEAR
7 8 9 ÷
4 5 6 ×
1 2 3 −
. 0 = +

46


ds [Running] - Oracle VM VirtualBox
Terminal - user@user-virtua...
Fri, 27 Mar, 00:23
user
Terminal - user@user-virtual-machine: ~/Desktop/qemu_new/DroidScope/qemu/objs
File Edit View Terminal Go Help
user@user-virtual-machine:~/Desktop/qemu_new/DroidScope/qemu/objs$
Right Ctrl