Characterizing Failures in Mobile OSes: A Case Study with

android and SYMBIAN

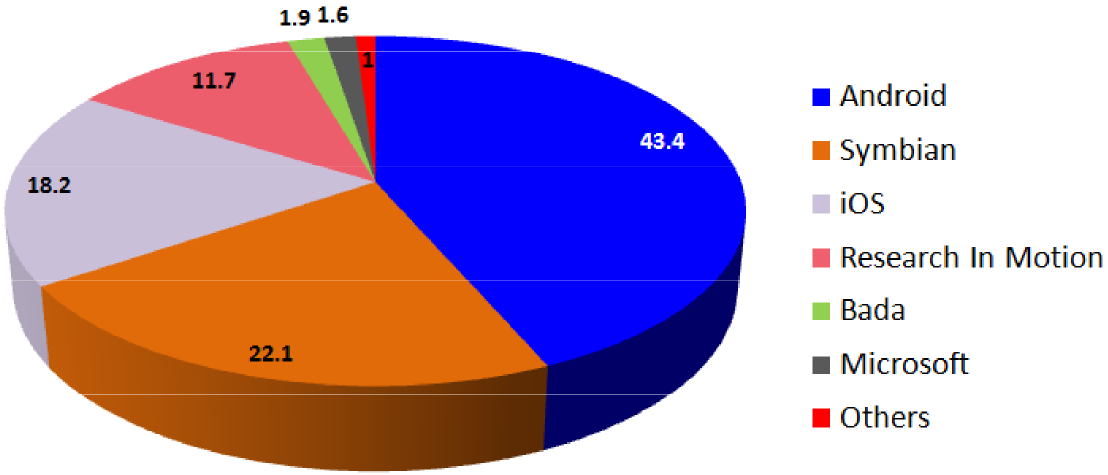
Amiya Kumar Maji, Kangli Hao,
Salmin Sultana, Saurabh Bagchi

Dependable Computing Systems Lab (DCSL)
School of Electrical and Computer Engineering
Purdue University

Emergence of Smartphones

- 14% of 1.2 billion mobile phone sales in 2009 are smartphones (Gartner)
- 19% of 1.6 billion mobile phone sales in 2010 are smartphones (Gartner)
  - 72.1% increase compared to 2009
- 25% of mobile phone sales in Q2 2011 are smartphones (Gartner)
- Smartphones expected to be the majority in US mobile market by end of 2011 (Nielsen)

Smartphone Market Share (Q2 2011)



Source: Gartner

The Changing Face of Mobile OSes

- *"There should be nothing that users can access on their desktop that they can't access on their cell phone."*
  – Andy Rubin

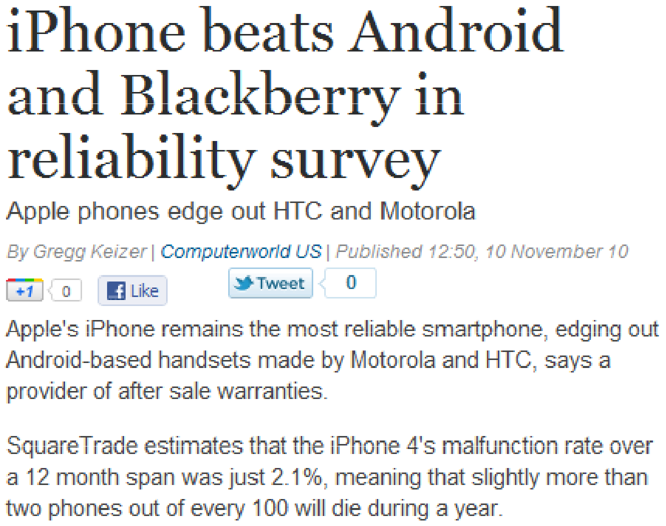
- Open source initiatives by Android and Symbian
- Public forums for bug reporting and bug fixes

How Reliable are Smartphones?

COMPUTERWORLDUK
THE VOICE OF IT MANAGEMENT

Warranty Claims

- iPhone 2.1%
- Motorola Droid 2.3%
- HTC 3.7%
- BlackBerry 6.3%



iPhone beats Android and Blackberry in reliability survey

Apple phones edge out HTC and Motorola

By Gregg Keizer | Computerworld US | Published 12:50, 10 November 10

Apple's iPhone remains the most reliable smartphone, edging out Android-based handsets made by Motorola and HTC, says a provider of after sale warranties.

SquareTrade estimates that the iPhone 4's malfunction rate over a 12 month span was just 2.1%, meaning that slightly more than two phones out of every 100 will die during a year.

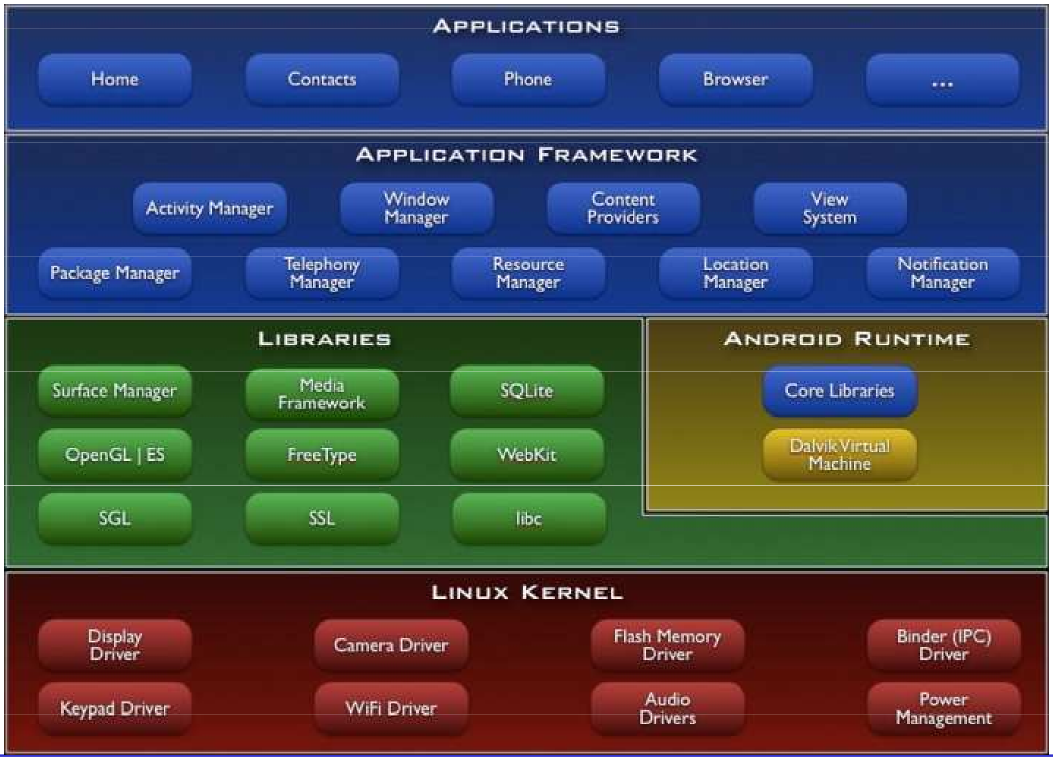
- Earlier study by Cinque *et al*. [DSN'07] looks at failure of Symbian phones using failure event logger

Our Objectives

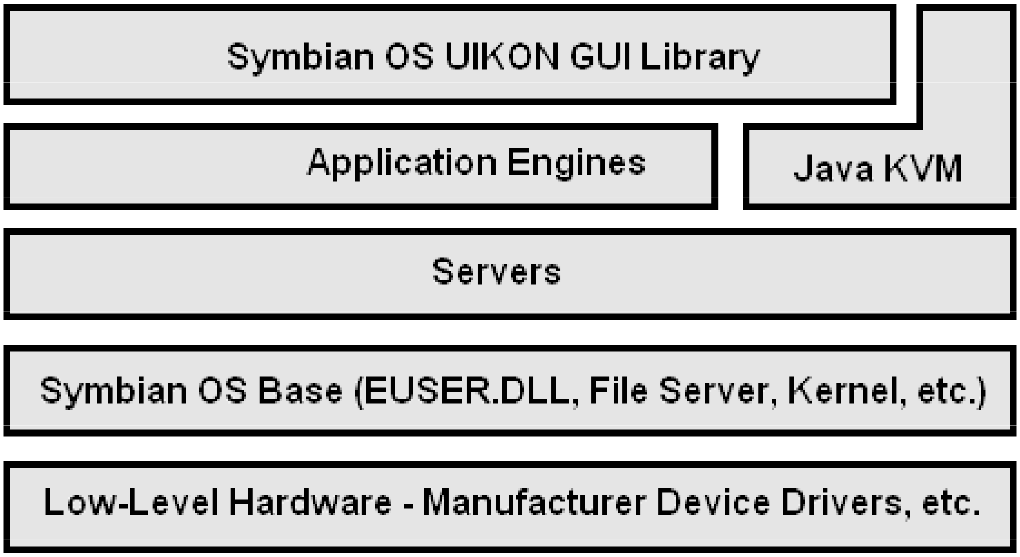
- To determine failure characteristics of smartphones from public bug databases
- Part I:
  - How failures manifest?
  - Are failures in Android and Symbian comparable?
- Part II:
  - Bug fix analysis
  - Tension between customizability, complexity, and bug density

Part I
Manifestation of Failures

Overview of Android



Overview of Symbian



Symbian OS UIKON GUI Library

Application Engines

Java KVM

Servers

Symbian OS Base (EUSER.DLL, File Server, Kernel, etc.)

Low-Level Hardware - Manufacturer Device Drivers, etc.

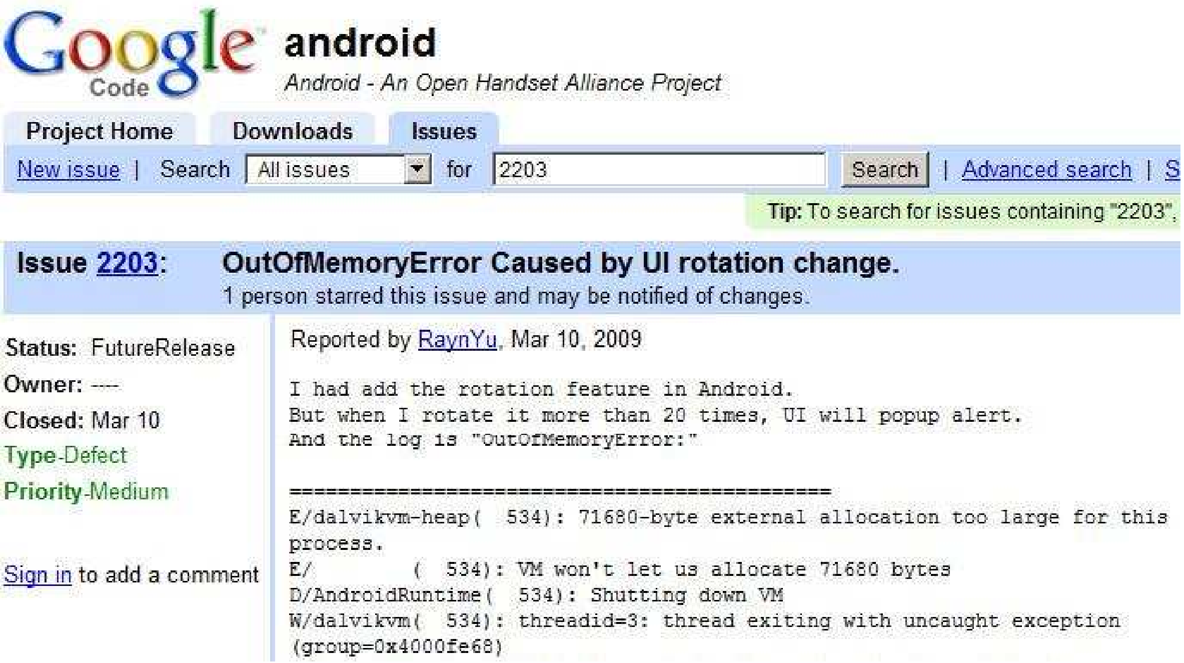
Data Collection

- Source:
  - Android Issue Reports:
    - Posted by app developers or users (with sufficient details)

      `http://code.google.com/p/android/issues/`
  - Symbian Bug Tracker:
    - Posted primarily by developers

      `http://developer.symbian.org/bugs/`

An Example Bug Report in Android


Google Code
android
Android - An Open Handset Alliance Project
Project Home
Downloads
Issues
New issue | Search All issues for 2203 Search | Advanced search | S
Tip: To search for issues containing "2203",

Issue 2203: OutOfMemoryError Caused by UI rotation change.

1 person starred this issue and may be notified of changes.

Status: FutureRelease
Owner: ----
Closed: Mar 10
Type-Defect
Priority-Medium

Sign in to add a comment

Reported by RaynYu, Mar 10, 2009

```
I had add the rotation feature in Android.
But when I rotate it more than 20 times, UI will popup alert.
And the log is "OutOfMemoryError:"

=============================================
E/dalvikvm-heap(  534): 71680-byte external allocation too large for this
process.
E/         (  534): VM won't let us allocate 71680 bytes
D/AndroidRuntime(  534): Shutting down VM
W/dalvikvm(  534): threadid=3: thread exiting with uncaught exception
(group=0x4000fe68)
```

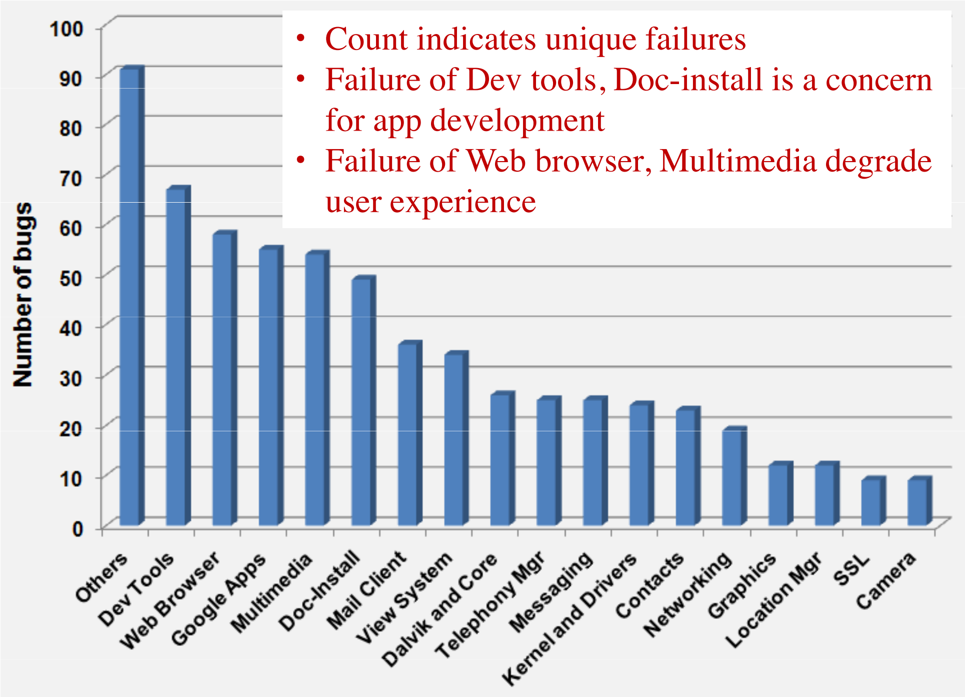
Dataset Summary

- Selection keywords:
  - Crash, shutdown, freeze, broken, failure, error, exception, and security
- Further data pruning due to:
  - Duplicates, pre-release bugs, too little details
  - Questions, enhancements
- Android
  - Timespan: October 2008-October 2009
  - Number of bug reports: 628
- Symbian
  - Timespan: Feb 2010-April 2010
  - Number of bug reports: 153

Location of Manifestation of Faults

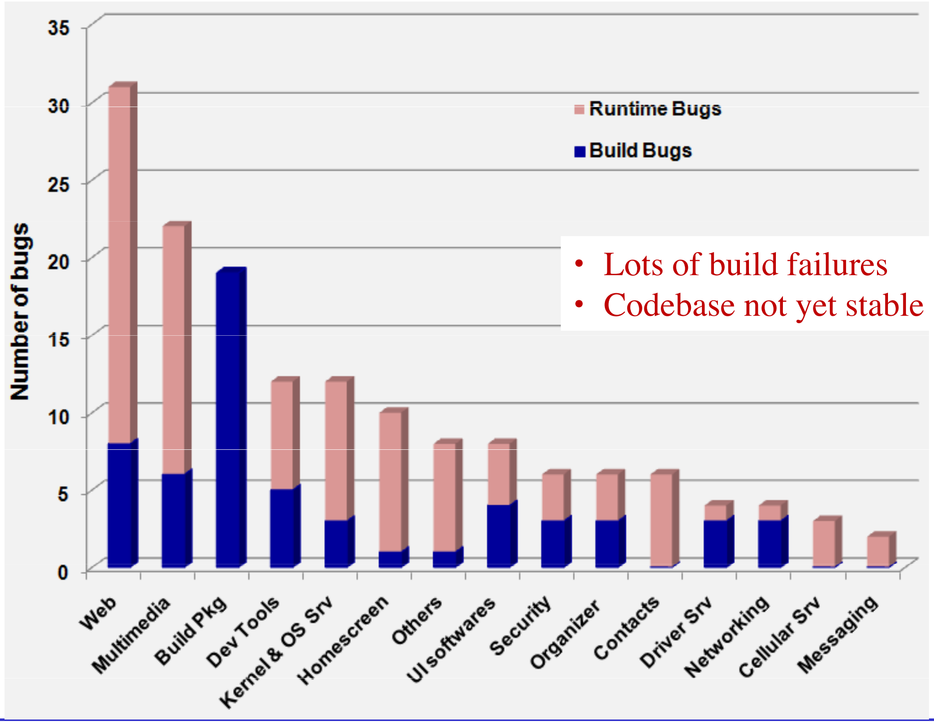
- Initial counts of faulty applications/libraries
  - Android: 55
  - Symbian: 41
- Aggregate related packages into “segments”
  - Eclipse, Android Dev Tool (ADT), Android Debug Bridge (ADB) as Development Tools
  - wrttools, web, websrv, and webuis as Web
- Count of segments
  - Android: 18
  - Symbian: 15

Distribution of Bugs: Android



- Count indicates unique failures
- Failure of Dev tools, Doc-install is a concern for app development
- Failure of Web browser, Multimedia degrade user experience

Distribution of Bugs: Symbian



- Lots of build failures
- Codebase not yet stable

Comparing the Graphs

- 4 of top 6 failure-prone segments are identical
  - Web, Multimedia, Development Tools, Documentation and Installation
- Less bugs in Kernel and Drivers
- Failure of Development Tools is a concern
- Persistence of bugs
  - More than 90% are permanent in nature (can be reproduced predictably)

Looking at User Forums

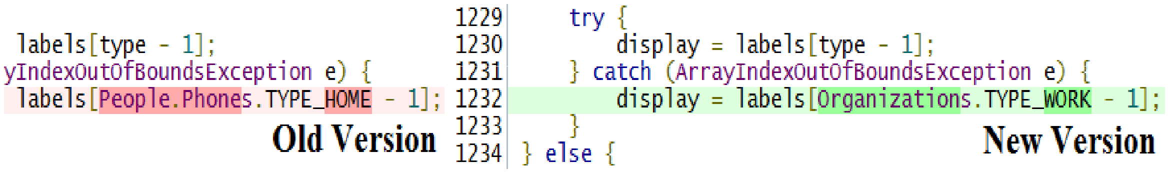
- T-Mobile G1 (Android) User Forum
  - 105 failure reports related to Messaging, Google Applications, Phone and Data Connections, Operating System and Software Development
  - Most frequent failures
    - Mail Client (15)
    - SD Card (11)
    - Media Player (9)
    - Messaging (9)
    - GPS and Location (8)
    - Web Browser (8)
  - Recovery actions similar to Cinque et al. [DSN'07]
    - Restart application, wait for some time, restart phone, modify settings, take out battery, factory reset, update firmware etc.

Part II
Analysis of Bug Fixes

Data Collection

- Source:
  - Android Code Review:

    `https://review.source.android.com`
- Timespan: October 2008-October 2009
- Count: 233 bug fixes from 29 projects
- Example



```
labels[type - 1];
yIndexOutOfBoundsException e) {
labels[People.Phones.TYPE_HOME - 1];
```

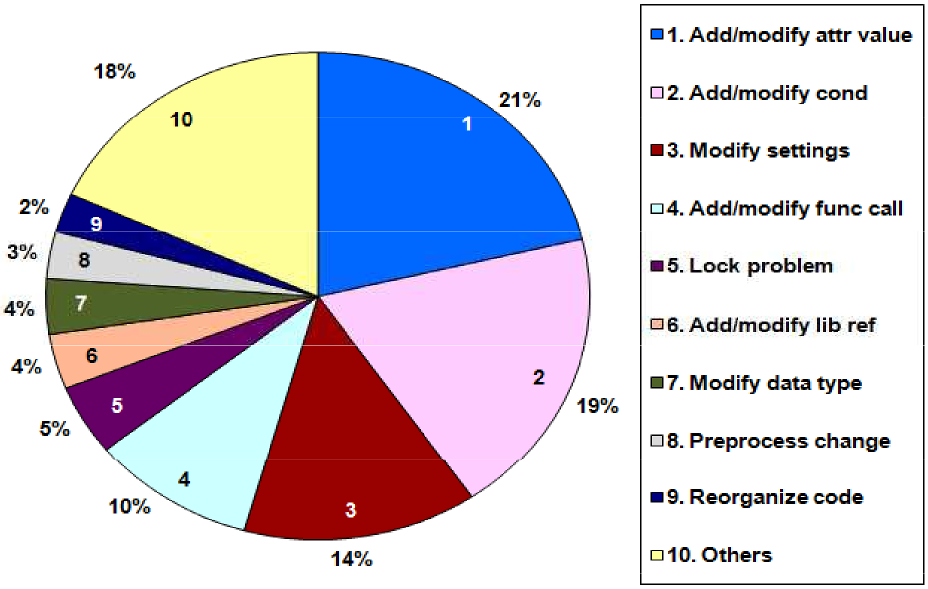
```
    try {
        display = labels[type - 1];
    } catch (ArrayIndexOutOfBoundsException e) {
        display = labels[Organizations.TYPE_WORK - 1];
    }
} else {
```

Categorization of Code Modifications

- Classify programmer errors responsible for failure
- Categories:
  - Add/modify attr value
  - Add/modify cond
  - Modify settings
  - Add/modify func call
  - Lock problems
  - Add/modify lib ref etc.

Categories for Bug Fixes

- 77% minor code change
- 23% major change



Observations

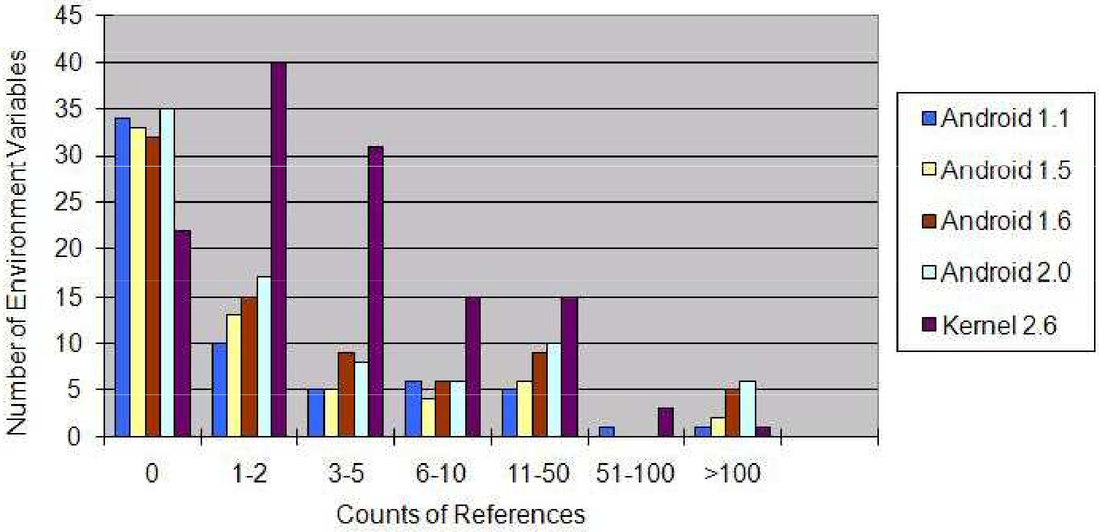
- Android is relatively new and still undergoing major modifications
- Detailed specification of program behavior can avoid significant number of bugs (specially in add/modify cond)
  - *if* statement missing *else* clause
- Modify settings is third largest category in bug fixes
  - Customizability does have its negative impact!

Analysis of Environment Variables

	# env vars	Total refs	Max refs
Android 1.1	62	819	577
Android 1.5	63	854	584
Android 1.6	76	1545	584
Android 2.0	82	2083	592
Linux Kernel 2.6.32	127	953	158

- Number of environment variables steadily increasing in Android

Distribution of References to Environment Variables



- Android: Majority of references to only a few env variables

Android: Cyclomatic Complexity vs. Bug Density

Projects	Bug Density X 10^4	# Bugs	SLOC	Avg. Cyclomatic	Max. Cyclomatic
kernel/omap	0.04	21	5,311,427	1.12	4973
kernel/msm	0.06	29	4,724,260	5.60	4973
kernel/common	0.07	31	4,688,175	5.82	4973
dalvik	0.18	14	771,865	2.23	766
development	0.46	10	216,344	2.18	169
framework/base	0.79	51	645,978	2.40	221
packages/apps/camera	1.33	2	14,962	2.15	20
packages/apps/mms	1.74	4	23,013	2.02	46
system/core	1.90	13	68,798	4.31	167
hardware/msm7k	2.42	3	12,382	4.00	23

Symbian: Cyclomatic Complexity vs. Bug Density

Segments	Bug Density X 10^4	# Bugs	SLOC	Avg. Cyclomatic	Max. Cyclomatic
Kernel and OS Services	0.03	12	3,684,192	3.02	1470
Security	0.08	6	752,148	2.29	134
Multimedia	0.12	22	1,866,577	2.44	558
Web	0.17	31	1,807,828	3.01	2442
HomeScreen	0.38	10	263,305	2.25	149
Build Pkg	0.63	19	299,868	2.24	268

Comparing Cyclomatic Complexity: Android and Symbian

- Bug density in both the systems is significantly low
- Low average CC due to default functions
- High max CC due to inlining and macros
- Max CC in Android Kernel (4973) is much higher than in Symbian (1470)

In a Nutshell

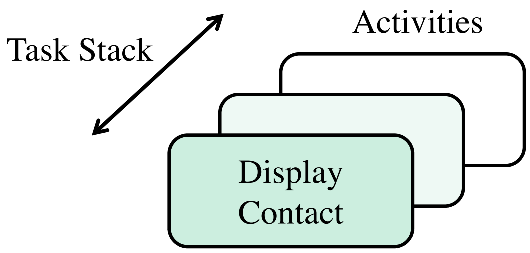
- Most of the bugs are permanent in nature suggesting immature codebase
- Kernel in both systems is robust. More rigorous testing is needed for middleware.
- Failures in Dev tools, Web, Multimedia, and Doc-Install are common in both systems
- Customizability does lead to significant fraction of bugs

How Robust is Input Validation in Android? (with Fahad Arshad)

- Test various components in Android with random input
  - Activity
  - Services
  - Broadcast Receivers
- Send random messages to these components
  - Monitor stack trace from logcat

Activities: Search a Contact

- Main
- Search
- Display Contact
- Activities
  - *UI component*



Task Stack

Activities

Display Contact

Intents

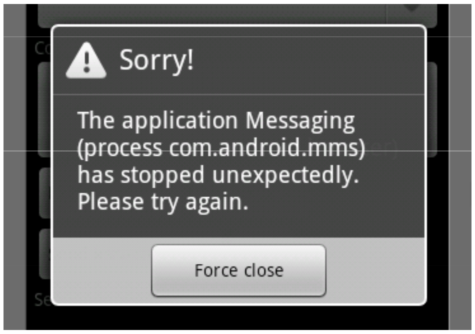
- Intent: abstract operation to be performed
- Components Interact using **Intent messages**
- Intent-filter: component advertise Intents
- Intent Resolution
  - Caller calls callee by component name
  - Runtime determines callee based on Intent

INTENT

◊ Component Name
◊ Action
◊ Data
◊ Category
◊ Extras

Fuzzing Methodology

- IntentFuzzer
  - Send random Intent messages to these components
  - Monitor stack trace
- Crash ⟹ Uncaught Exception



Sorry!

The application Messaging (process com.android.mms) has stopped unexpectedly. Please try again.

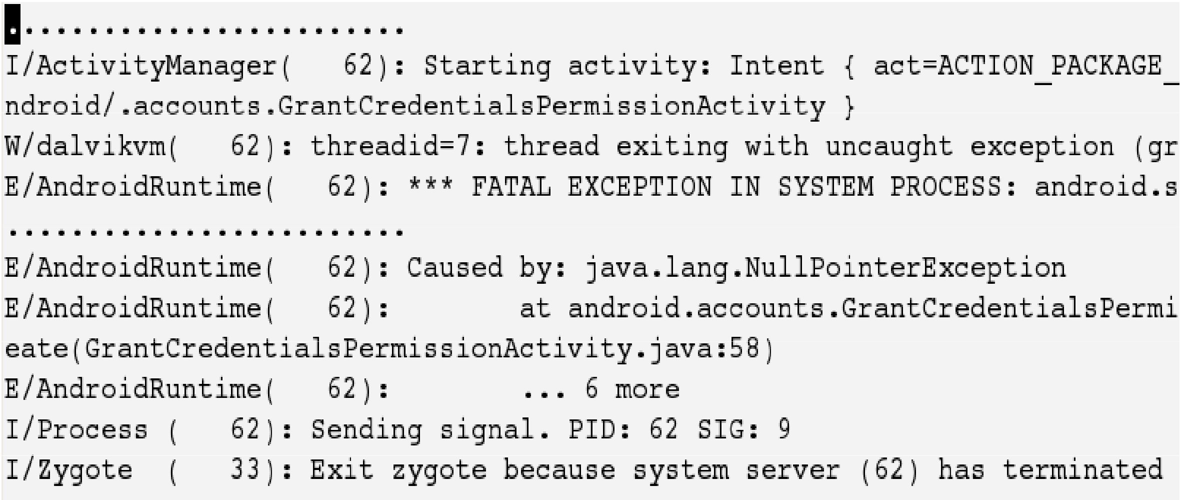
Exception Handling Errors

Component Type	No of Components Tested	No of Components Crashed	Type of Exception
Broadcast Receiver	42	8	NullPointerException
Services	27	3	NullPointerException
Activities Round 1	294	15	NullPointerException
		4	ClassNotFoundException
		1	IllegalArgumentException
		1	ActivityNotFoundException
Activities Round 2	294	10	NullPointerException
		3	ClassNotFoundException
		2	IllegalArgumentException
		1	ActivityNotFoundException
Detected 36 Bugs		1	UnsupportedOperationException

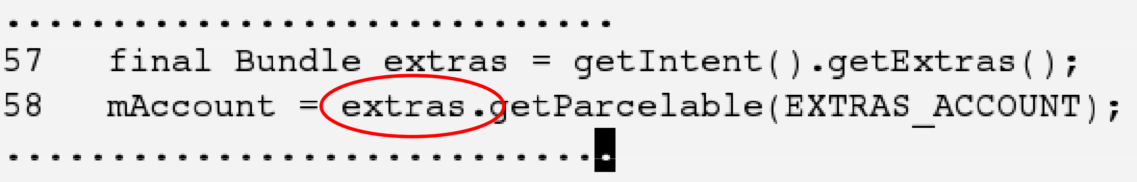
Security Concerns

- 4 of 36 detected bugs caused Android system process (android.server.ServerThread) to crash
- No additional permission was needed to run IntentFuzzer
  - Was able to run activities under privileged process
- App developers must be careful when dealing with Intents
  - Exception handling is a must!

System Crash



```
.........................
I/ActivityManager(   62): Starting activity: Intent { act=ACTION_PACKAGE_
ndroid/.accounts.GrantCredentialsPermissionActivity }
W/dalvikvm(   62): threadid=7: thread exiting with uncaught exception (gr
E/AndroidRuntime(   62): *** FATAL EXCEPTION IN SYSTEM PROCESS: android.s
.........................
E/AndroidRuntime(   62): Caused by: java.lang.NullPointerException
E/AndroidRuntime(   62):        at android.accounts.GrantCredentialsPermi
eate(GrantCredentialsPermissionActivity.java:58)
E/AndroidRuntime(   62):        ... 6 more
I/Process (   62): Sending signal. PID: 62 SIG: 9
I/Zygote  (   33): Exit zygote because system server (62) has terminated
```



```
..............................
57   final Bundle extras = getIntent().getExtras();
58   mAccount = extras.getParcelable(EXTRAS_ACCOUNT);
..............................
```

Conclusion

- Input validation in Android needs more attention
- Intent passing and default security permissions are a concern
- Development tools, Web browser, Multimedia need to be more robust
- Both Android and Symbian show similar fault manifestation

Looking Forward

- Evaluation of Inter Component Communication in Android
  - Can the detected bugs be exploited?
- "Mobile phones are more personal than personal computers"
  - What are the privacy implications?
- Smartphones have lesser physical security
  - Encryption vs. usability

Thanks

Questions?