

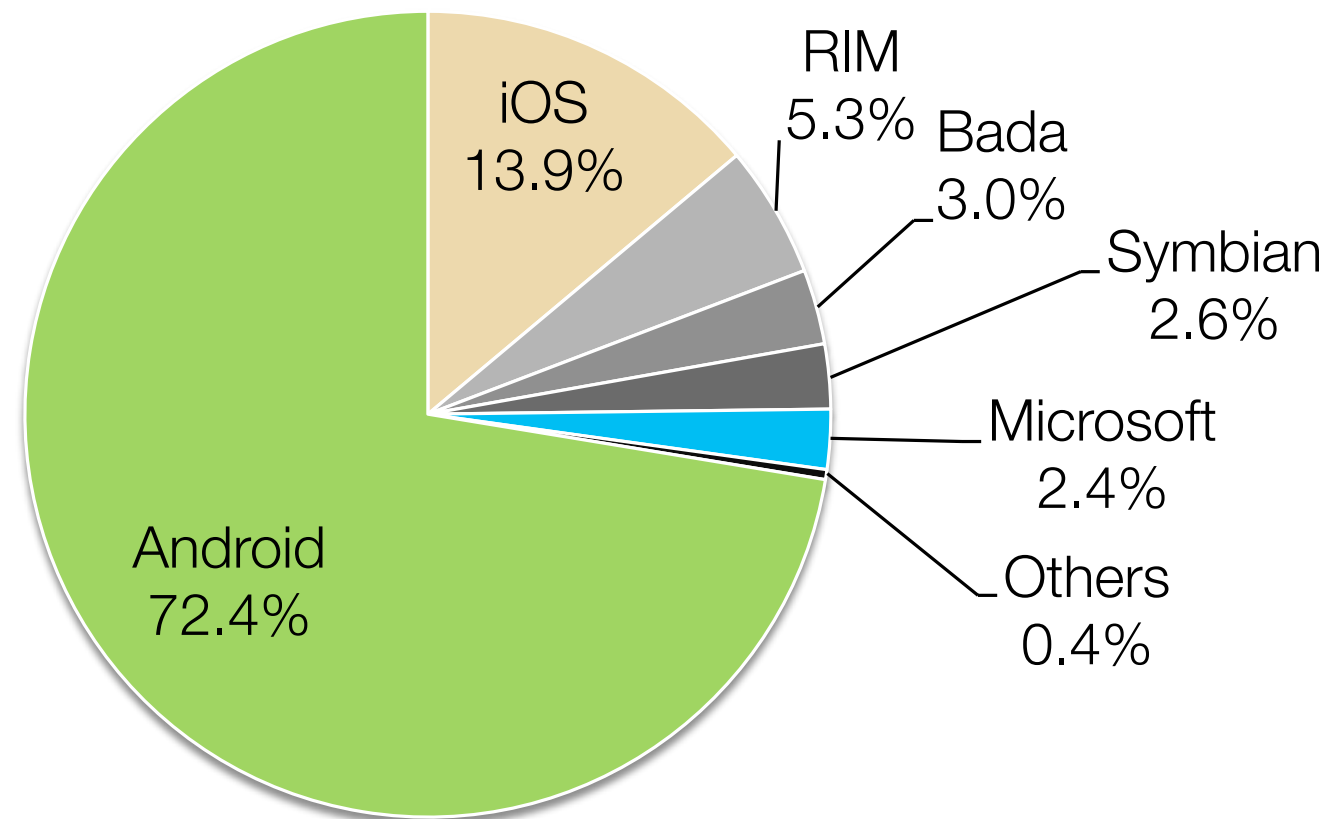


Systematic Audit of Third-Party Android Phones

Michael Mitchell, Guanyu Tian, **Zhi Wang**
Florida State University

Android Leads Smartphone Market

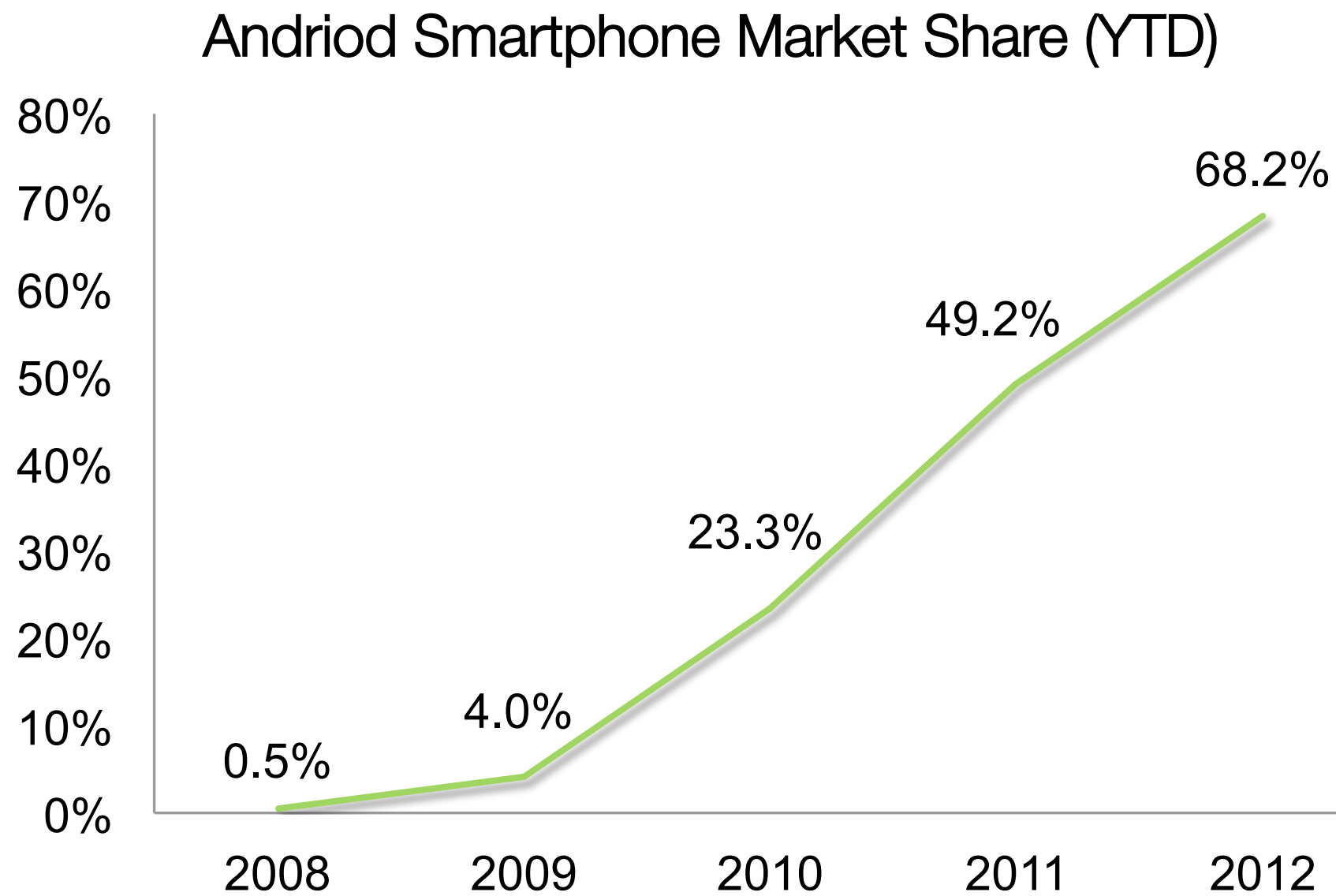
WW Smartphone Sales by OS in 3Q12



- Gartner: "Market Share: Mobile Phones by Region and Country, 3Q12."



Android Leads Smartphone Market



Growing Pains of Android

- Many manufacturers produce Android phones
 - Samsung, HTC, LG, Motorola, Sony....



Growing Pains of Android

- Many manufacturers produce Android phones
 - Samsung, HTC, LG, Motorola, Sony....
- Hundreds of similar products
 - all Android phones are rather similar, especially in software



Growing Pains of Android

- Many manufacturers produce Android phones
 - Samsung, HTC, LG, Motorola, Sony....
- Hundreds of similar products
 - all Android phones are rather similar, especially in software



Samsung



HTC



Motor



LG

Growing Pains of Android

- Many manufacturers produce Android phones
 - Samsung, HTC, LG, Motorola, Sony....
- Hundreds of similar products
 - all Android phones are rather similar, especially in software
- Vendors are eager to differentiate by customization
 - Android is open any way!



Samsung



HTC



Motor



LG

Growing Pains of Android

- Many manufacturers produce Android phones
 - Samsung, HTC, LG, Motorola, Sony....
- Hundreds of similar products
 - all Android phones are rather similar, especially in software
- Vendors are eager to differentiate by customization
 - Android is open any way!
 - **but are they safe???**



Samsung



HTC



Motor



LG

Deadly Customization



Deadly Customization

A screenshot of a Forbes article displayed on the ZDNet website. The article title is "Carrier IQ: The Rootkit Keylogger on Most US Smartphones". The text below the title states that Carrier IQ is software loaded on certain US cellphone networks like Sprint, but it does not appear on Apple's iPhones. The article explains that the software generates logs for network analysis to identify problems.

Search ZDNet

Forbes ▾ **New Posts** +3 posts this hour **Most Popular** Fiscal Cliff Explained **Lists** 100 Largest

Carrier IQ: The Rootkit Keylogger on Most US Smartphones

[+ Comment now](#)

But not it appears on [Apple's](#) iPhones.

Carrier IQ is a piece of software which certain US cellphone networks (Sprint for example) load onto their contract phones before they are released to consumers. The basic stated idea is quite simple: if there are problems then the software generates logs which the network can then analyse to see what the problems are.

Deadly Customization

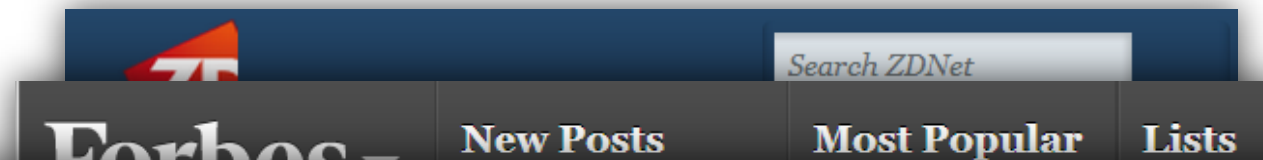


Table 3. Capability leak results of eight Android-based smartphones (E: explicit leaks; I: implicit leaks)

Permission	Legend		HTC				Motorola				Samsung		Google			
			EVO 4G		Wildfire S		Droid		Droid X		Epic 4G		Nexus One		Nexus S	
	E	I	E	I	E	I	E	I	E	I	E	I	E	I	E	I
ACCESS_COARSE_LOCATION	✓	✓	✓	✓	.	✓	.	.	✓	.	.	.	.	.	.	.
ACCESS_FINE_LOCATION	✓	.	✓	.	.	✓	.	.	✓	.	.	.	.	.	.	.
CALL_PHONE	.	.	.	.	.	.	.	.	.	.	✓	✓	.	.	.	.
CALL_PRIVILEGED	.	.	.	.	.	✓ ¹	.	.	.	.	.	.	.	.	.	.
CAMERA	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
DELETE_PACKAGES	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.
INSTALL_PACKAGES	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
MASTER_CLEAR	.	.	.	.	.	.	.	.	.	.	✓	.	.	.	.	.
READ_PHONE_STATE	.	✓	.	✓	.	✓	.	.	✓	.	.	✓	.	.	.	.
REBOOT	.	.	✓	.	.	.	.	.	.	.	.	.	.	.	.	.
RECORD_AUDIO	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
SEND_SMS	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
SHUTDOWN	.	.	✓	.	.	.	.	.	.	.	.	.	.	.	.	.
Total	6	2	8	2	4	4	1	0	4	0	3	2	1	0	1	0

Deadly Customization

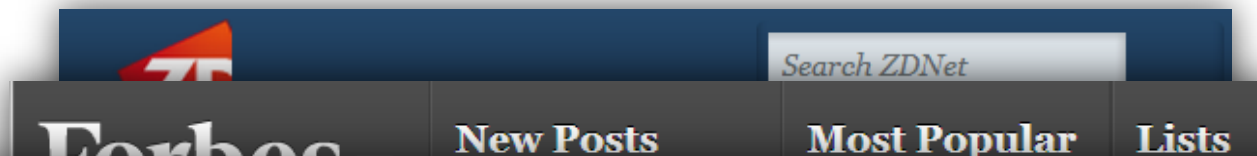


Table 3. Capability leak results of eight Android-based smartphones (E: explicit leaks; I: implicit leaks)

Permission	Legend		HTC				Motorola				Samsung		Google			
			EVO 4G		Wildfire S		Droid		Droid X		Epic 4G		Nexus One		Nexus S	
	E	I	E	I	E	I	E	I	E	I	E	I	E	I	E	I
ACCESS_COARSE_LOCATION	✓	✓	✓	✓	.	✓	.	.	✓	.	.	.	.	.	.	.
ACCESS_FINE_LOCATION	✓	.	✓	.	.	✓	.	.	✓	.	.	.	.	.	.	.
CALL_PHONE	.	.	.	.	.	.	.	.	.	.	✓	✓	.	.	.	.
CALL_PRIVILEGED	.	.	.	.	.	✓ ¹	.	.	.	.	.	.	.	.	.	.
CAMERA	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
DELETE_PACKAGES	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.
INSTALL_PACKAGES	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
MASTER_CLEAR	.	.	.	.	.	.	.	.	.	.	✓	.	.	.	.	.
READ_PHONE_STATE	.	✓	.	✓	.	✓	.	.	✓	.	.	✓	.	.	.	.
REBOOT	.	.	✓	.	.	.	.	.	.	.	.	.	.	.	.	.
RECORD_AUDIO	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
SEND_SMS	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
SHUTDOWN	.	.	✓	.	.	.	.	.	.	.	.	.	.	.	.	.
Total	6	2	8	2	4	4	1	0	4	0	3	2	1	0	1	0

Deadly Customization

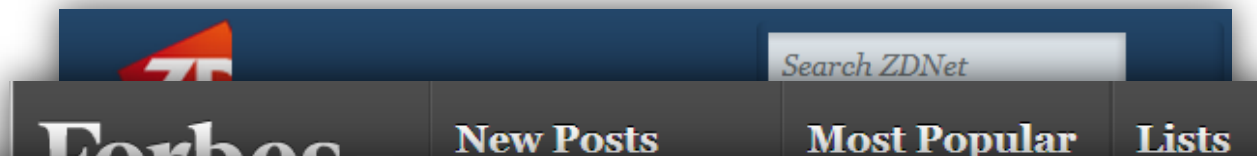


Table 3. Capability leak results of eight Android-based smartphones (E: explicit leaks; I: implicit leaks)

Permission	Legend		HTC				Motorola				Samsung		Google			
			EVO 4G		Wildfire S		Droid		Droid X		Epic 4G		Nexus One		Nexus S	
	E	I	E	I	E	I	E	I	E	I	E	I	E	I	E	I
ACCESS_COARSE_LOCATION	✓	✓	✓	✓	.	✓	.	.	✓	.	.	.	.	.	.	.
ACCESS_FINE_LOCATION	✓	.	✓	.	.	✓	.	.	✓	.	.	.	.	.	.	.
CALL_PHONE	.	.	.	.	.	.	.	.	.	.	✓	✓	.	.	.	.
CALL_PRIVILEGED	.	.	.	.	.	✓ ¹	.	.	.	.	.	.	.	.	.	.
CAMERA	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
DELETE_PACKAGES	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.
INSTALL_PACKAGES	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
MASTER_CLEAR	.	.	.	.	.	.	.	.	.	.	✓	.	.	.	.	.
READ_PHONE_STATE	.	✓	.	✓	.	✓	.	.	✓	.	.	✓	.	.	.	.
REBOOT	.	.	✓	.	.	.	.	.	.	.	.	.	.	.	.	.
RECORD_AUDIO	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
SEND_SMS	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
SHUTDOWN	.	.	✓	.	.	.	.	.	.	.	.	.	.	.	.	.
Total	6	2	8	2	4	4	1	0	4	0	3	2	1	0	1	0

Deadly Customization

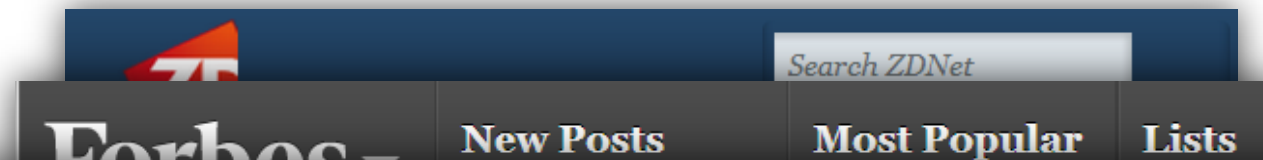


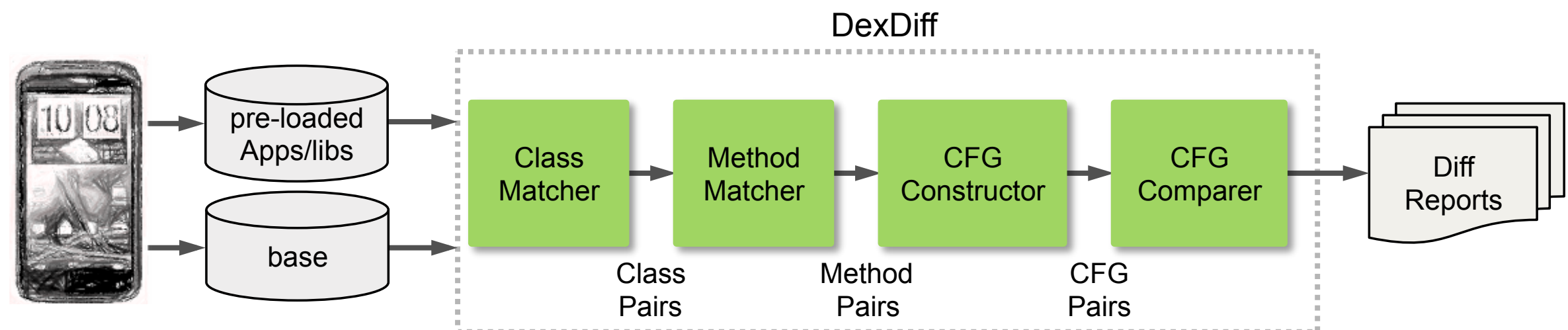
Table 3. Capability leak results of eight Android-based smartphones (E: explicit leaks; I: implicit leaks)

Permission	Legend		HTC				Motorola				Samsung		Google			
			EVO 4G		Wildfire S		Droid		Droid X		Epic 4G		Nexus One		Nexus S	
	E	I	E	I	E	I	E	I	E	I	E	I	E	I	E	I
ACCESS_COARSE_LOCATION	✓	✓	✓	✓	.	✓	.	.	✓	.	.	.	.	.	.	.
ACCESS_FINE_LOCATION	✓	.	✓	.	.	✓	.	.	✓	.	.	.	.	.	.	.
CALL_PHONE	.	.	.	.	.	.	.	.	.	.	✓	✓	.	.	.	.
CALL_PRIVILEGED	.	.	.	.	.	✓ ¹	.	.	.	.	.	.	.	.	.	.
CAMERA	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
DELETE_PACKAGES	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.	✓ ²	.
INSTALL_PACKAGES	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
MASTER_CLEAR	.	.	.	.	.	.	.	.	.	.	✓	.	.	.	.	.
READ_PHONE_STATE	.	✓	.	✓	.	✓	.	.	✓	.	.	✓	.	.	.	.
REBOOT	.	.	✓	.	.	.	.	.	.	.	.	.	.	.	.	.
RECORD_AUDIO	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
SEND_SMS	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	.	.	.
SHUTDOWN	.	.	✓	.	.	.	.	.	.	.	.	.	.	.	.	.
Total	6	2	8	2	4	4	1	0	4	0	3	2	1	0	1	0

These vulnerabilities are not in the original Android!!!

Our Approach

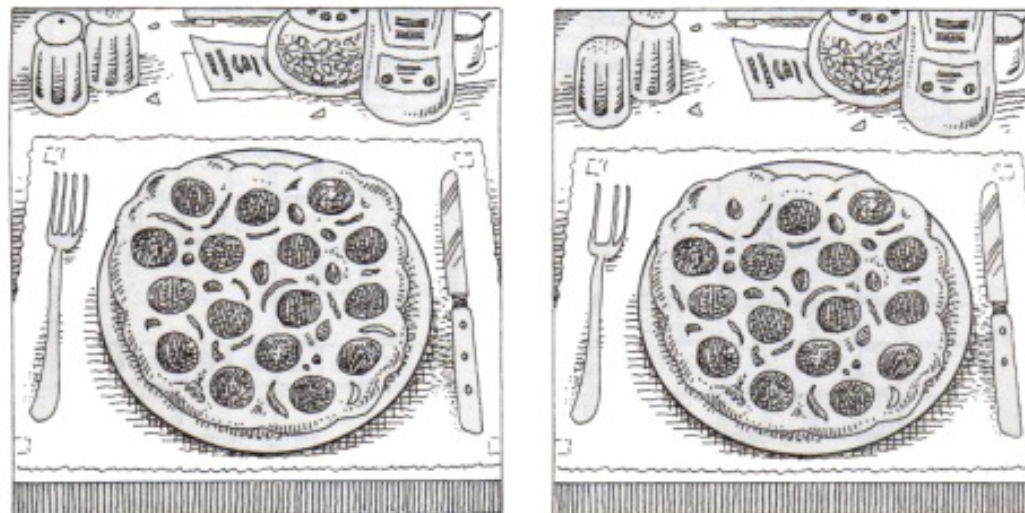
- Systematically compare **third-party phones** to **Google's original releases**
 - what changes have been made?
 - especially to the core **Android framework and services**
 - are these changes safe?
 - if unsafe, can we exploit it (manually)?



Why DexDiff

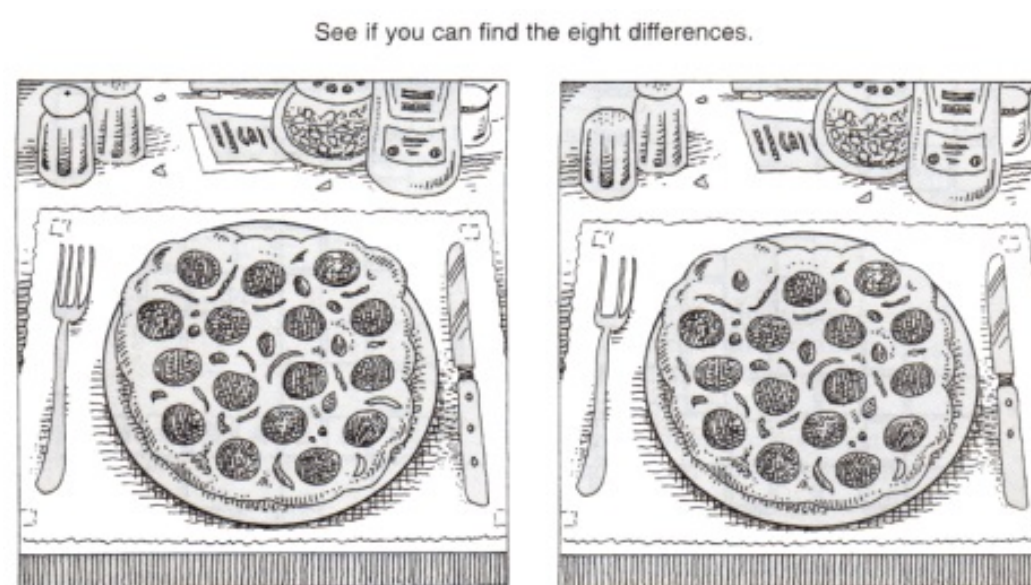
- Android is open-sourced with **Apache/BSD**/GPL licenses
 - vendors are free to keep changes to framework closed
 - those apps are too large to process manually

See if you can find the eight differences.



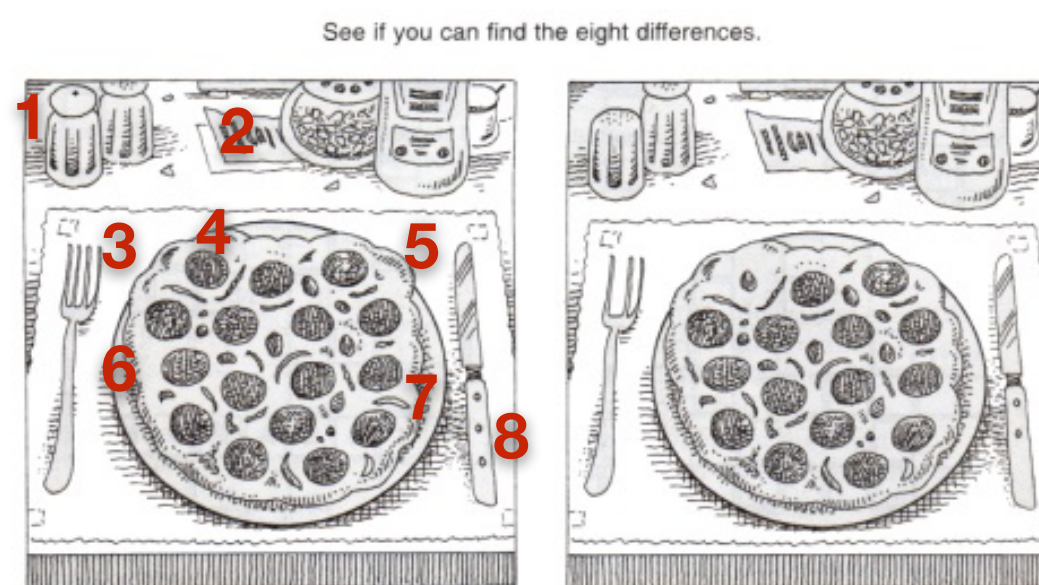
Why DexDiff

- Android is open-sourced with **Apache/BSD**/GPL licenses
 - vendors are free to keep changes to framework closed
 - those apps are too large to process manually
- Most Android apps are Java-based in **Dalvik** bytecode
 - no systems available to compare Dalvik binaries
 - complimentary to systems for native binaries (BinHunt, BinDiff...)



Why DexDiff

- Android is open-sourced with **Apache/BSD**/GPL licenses
 - vendors are free to keep changes to framework closed
 - those apps are too large to process manually
- Most Android apps are Java-based in **Dalvik** bytecode
 - no systems available to compare Dalvik binaries
 - complimentary to systems for native binaries (BinHunt, BinDiff...)





DexDiff Overview

- Input: two Android (Java) apps
- Output: differences between these two apps

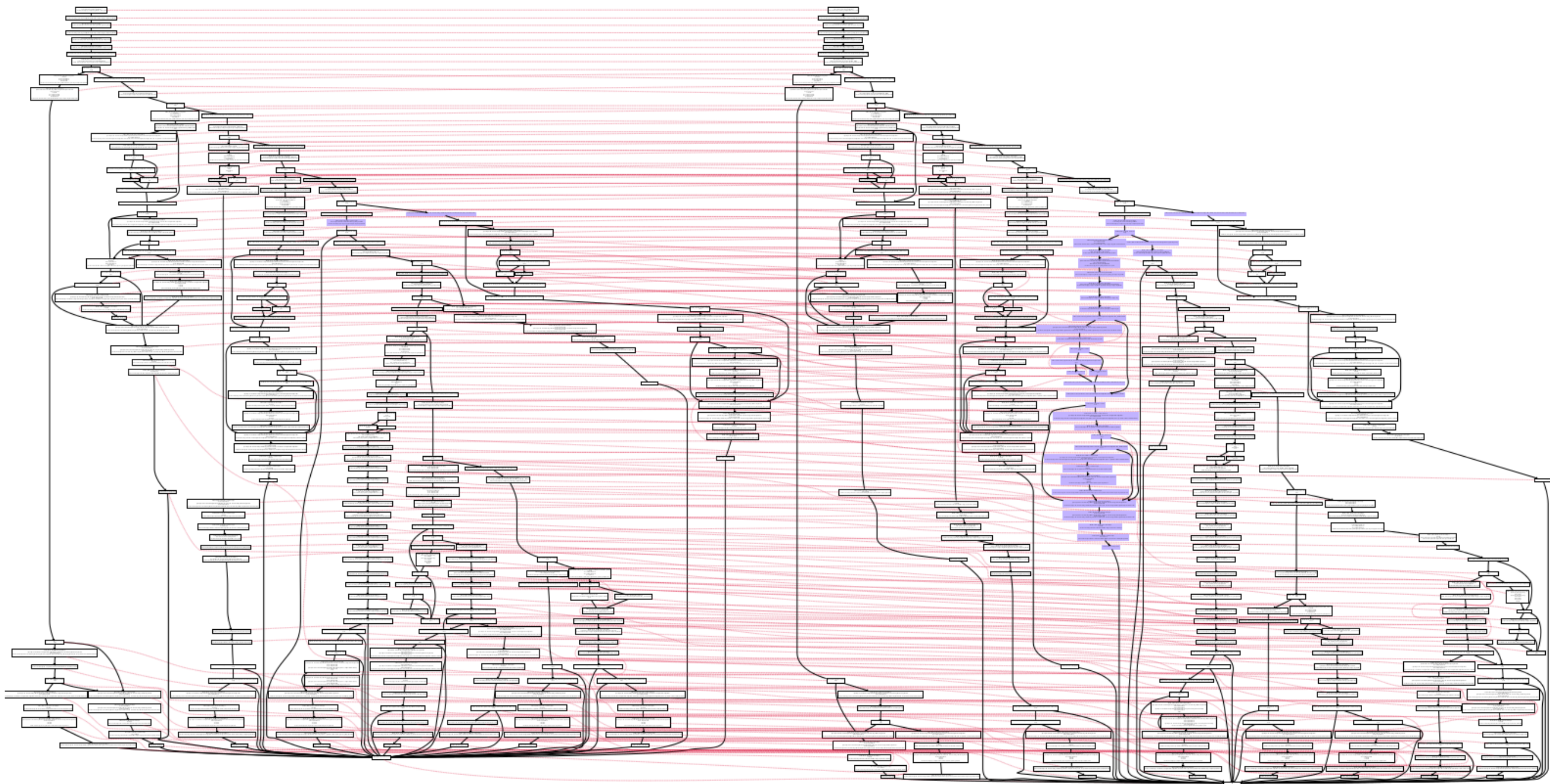


DexDiff Overview

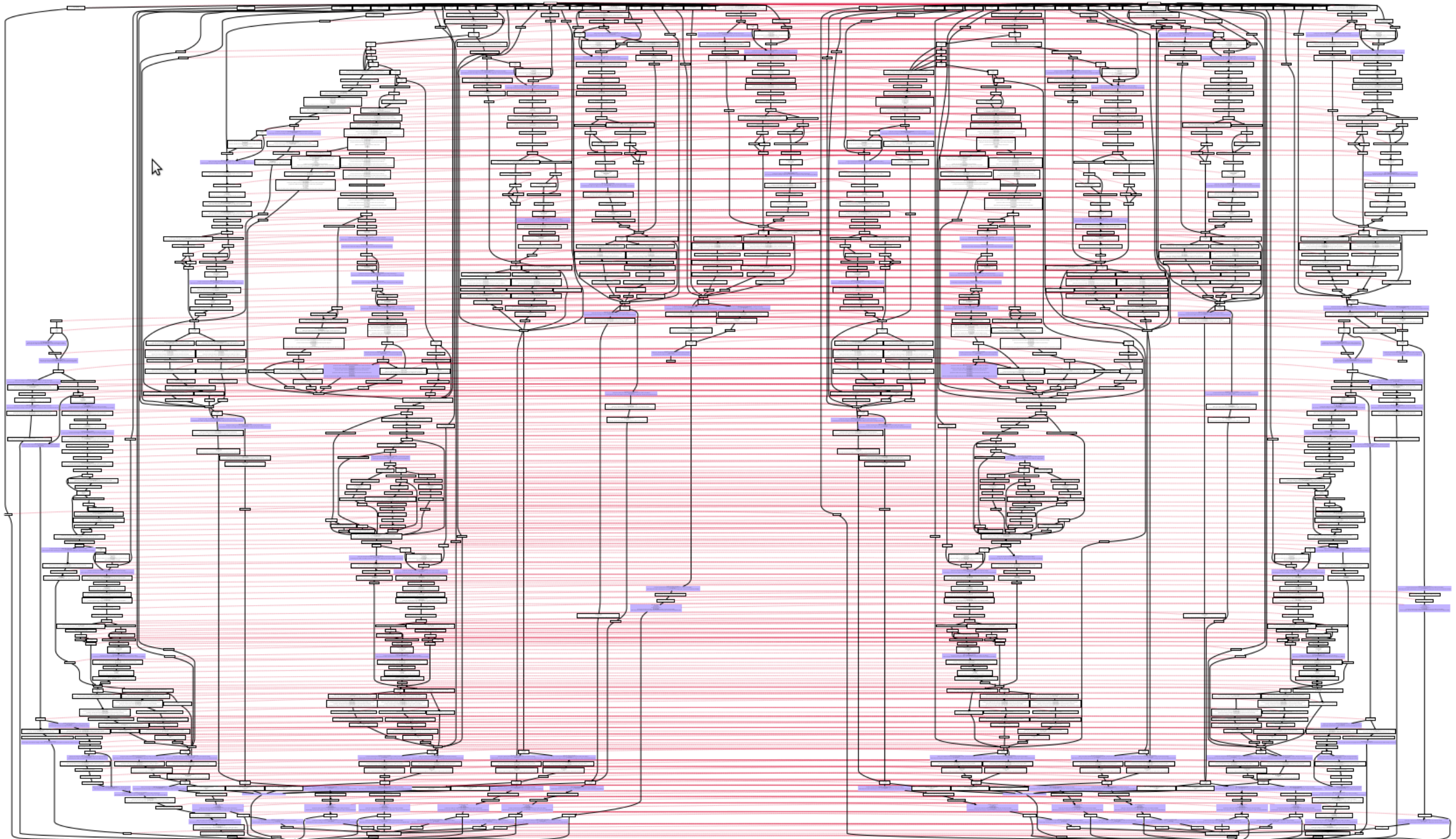
- Input: two Android (Java) apps
- Output: differences between these two apps
- Process:
 - parse the apps into Java classes
 - find matching classes $\Rightarrow$ pairs of matched classes
 - find matching methods in them $\Rightarrow$ pairs of matched methods
 - construct CFGs for matched methods
 - compare CFGs (graph isomorphism)



DexDiff Sample Output



DexDiff Sample Output





Matching Classes

- Parse the (two) Android binaries into classes
 - official Gingerbread (2.3.3) framework.dex: **3924** classes
 - HTC EVO 4G framework.dex: **5423** classes



Matching Classes

- Parse the (two) Android binaries into classes
 - official Gingerbread (2.3.3) framework.dex: **3924** classes
 - HTC EVO 4G framework.dex: **5423** classes
- Calculate pair-wise similarity score for classes
 - convert classes into string arrays
 - calculate string similarity using **n-gram**



Matching Classes

- Parse the (two) Android binaries into classes
 - official Gingerbread (2.3.3) framework.dex: **3924** classes
 - HTC EVO 4G framework.dex: **5423** classes
- Calculate pair-wise similarity score for classes
 - convert classes into string arrays
 - calculate string similarity using **n-gram**
- Find a matching that maximizes overall similarity
 - **Hungarian algorithm**



Matching Classes: Similarity

- Convert classes into string arrays

Matching Classes: Similarity

- Convert classes into string arrays
 - each terminal in the BNF is a string (semantic unit)

C	$\rightarrow$	$class_name \cdot superclass_name \cdot I_List \cdot F_List \cdot M_List$
I_List	$\rightarrow$	$I_List \cdot I$
I	$\rightarrow$	$interface_signature \cdot interface_name$
F_List	$\rightarrow$	$F_List \cdot F$
F	$\rightarrow$	$field_type \cdot field_name$
M_List	$\rightarrow$	$M_List \cdot M$
M	$\rightarrow$	$method_signature \cdot method_name \cdot INST_List$
$INST_List$	$\rightarrow$	$INST_List \cdot instruction$

Matching Classes: Similarity

- Convert classes into string arrays
 - each terminal in the BNF is a string (semantic unit)
 - registers are omitted for instructions
 - e.g., **new-array** v0, v1 [**Ljava/lang/String**;
 - Java: late-binding $\Rightarrow$ types retained in bytecodes

C	$\rightarrow$	$class_name \cdot superclass_name \cdot I_List \cdot F_List \cdot M_List$
I_List	$\rightarrow$	$I_List \cdot I$
I	$\rightarrow$	$interface_signature \cdot interface_name$
F_List	$\rightarrow$	$F_List \cdot F$
F	$\rightarrow$	$field_type \cdot field_name$
M_List	$\rightarrow$	$M_List \cdot M$
M	$\rightarrow$	$method_signature \cdot method_name \cdot INST_List$
$INST_List$	$\rightarrow$	$INST_List \cdot instruction$

Matching Classes: Similarity

- Slide a window of n on the array, step one at a time
- Each window of data (strings) is a n -gram
 - e.g., 2-gram set of abcd: {ab, bc, cd}
 - 2-gram set of adbc: {ad, db, bc}
- Similarity of class a and b:

$$\textit{Similarity}(c_a, c_b) = \frac{|c_a \cap c_b|}{|c_a \cup c_b|}$$



Matching Classes: Similarity

- However, string comparison is slow
 - framework.dex: **3924** classes v.s **5423** classes



Matching Classes: Similarity

- However, string comparison is slow
 - framework.dex: **3924** classes v.s **5423** classes
- Apply a hash function to n-grams
 - **feature hashing**

Matching Classes: Similarity

- However, string comparison is slow
 - framework.dex: **3924** classes v.s **5423** classes
- Apply a hash function to n-grams
 - **feature hashing**
- Calculate similarity with hash values

$$\textit{Similarity}(c_a, c_b) = \frac{|c_a \cap c_b|}{|c_a \cup c_b|}$$



Matching Classes

- Hungarian algorithm solves the **assignment problem**
 - n workers and n tasks, find an optimal (least cost/most benefit) assignment of the workers to the tasks



Matching Classes

- Hungarian algorithm solves the **assignment problem**
 - n workers and n tasks, find an optimal (least cost/most benefit) assignment of the workers to the tasks
- DexDiff uses it to find a matching of classes to **maximize total similarity**



Matching Classes

- Hungarian algorithm solves the **assignment problem**
 - n workers and n tasks, find an optimal (least cost/most benefit) assignment of the workers to the tasks
- DexDiff uses it to find a matching of classes to **maximize total similarity**
- Output: pairs of matched classes



Matching Methods

- Input: a pair of matched classes
- Output: pairs of matched methods in these classes



Matching Methods

- Input: a pair of matched classes
- Output: pairs of matched methods in these classes
- Method: the same as matching classes
 - calculate a matrix of similarity scores for methods
 - apply Hungarian algorithm for a maximal matching



Constructing CFGs

- A normal CFG construction algorithm
 - break methods into basic blocks
 - link basic blocks together

Constructing CFGs

- A normal CFG construction algorithm
 - break methods into basic blocks
 - link basic blocks together
- Hungarian algorithm **cannot** be used to match BBs
 - many basic blocks are similar to each other
 - connectivity of CFGs should be used to refine results



Comparing CFGs

- Input: two CFGs



Comparing CFGs

- Input: two CFGs
- Output: maximum common isomorphic sub-graphs
 - unmatched nodes are BBs **modified**, **inserted**, or **deleted**

Comparing CFGs

- Input: two CFGs
- Output: maximum common isomorphic sub-graphs
 - unmatched nodes are BBs **modified**, **inserted**, or **deleted**
- **Maximum sub-graph isomorphism** is NP-complete
 - efficient algorithm exists for two similar graphs
 - two inputs in DexDiff are similar

Comparing CFGs

```
1: procedure BACKTRACK( $\mathcal{D}$ )
2:   if EXTENDABLE( $\mathcal{D}$ ) then
3:      $v = \text{PICKNODE}(\mathcal{D})$ 
4:      $\mathcal{Z} = \text{GETMAPPABLENODES}(v, \mathcal{D})$ 
5:     for all  $w \in \mathcal{Z}$  do
6:        $\mathcal{M} = \mathcal{M} + \{(v, w)\}$ 
7:        $\mathcal{D}' = \text{REFINE}(\mathcal{D})$ 
8:       BACKTRACK( $\mathcal{D}'$ )
9:        $\mathcal{M} = \mathcal{M} - \{(v, w)\}$ 
10:    end for
11:     $\mathcal{V} = \mathcal{V} - \{v\}$ 
12:    BACKTRACK( $\mathcal{D}$ )
13:     $\mathcal{V} = \mathcal{V} + \{v\}$ 
14:  else if  $|\mathcal{M}| > |\mathcal{R}|$  then
15:     $\mathcal{R} = \mathcal{M}$ 
16:  end if
17: end procedure
```



Comparing CFGs

- Backtracking is a brute-force algorithm
 - it converges fast if **Refine** can quickly prune dead trees

Comparing CFGs

- Backtracking is a brute-force algorithm
 - it converges fast if **Refine** can quickly prune dead trees
- Timeout is used to prevent waiting for too long
 - limit the number of recursive calls (1 million)
 - **38** timeouts out of **6560** CFG pairs (**0.58%**) for framework.dex

Comparing CFGs

- Backtracking is a brute-force algorithm
 - it converges fast if **Refine** can quickly prune dead trees
- Timeout is used to prevent waiting for too long
 - limit the number of recursive calls (1 million)
 - **38** timeouts out of **6560** CFG pairs (**0.58%**) for framework.dex
- Leading causes of timeout
 - large CFGs with many unmatched basic blocks
 - many identical basic blocks (switch/exception)
 - nodes with many similar parents/children



Implementation

- Implemented in 13.5K lines of “C” source code
- Use “dot” to generate CFGs side-by-side
 - matching is efficient, “dot” is slow



Evaluation

- Experiments with HTC EVO 4G
 - Carrier IQ: details of information collection in the browser
 - subsequently removed in a firmware update
 - a hook in `com.android.calculator2.CalculatorImageButton.onTouch` survives
 - Vulnerable device management app (newly discovered with DexDiff)
 - local app with INTERNET access can obtain private information, **install/delete apps, wipe/lock/brick** the phone



Changes to the Framework

Name	Base			Phone			Modified		New		Time
	Size	Class#	Method#	Size	Class#	Method#	Class#	Method#	Class#	Method#	
am	25K	6	38	26K	6	39	1	1	0	1	0.10
policy	179K	96	833	258K	140	1,200	29	105	44	368	10.23
test.runner	172K	105	1,001	172K	105	1,002	2	1	0	1	0.17
bmgr	12K	2	25	14K	2	27	1	2	0	2	0.05
bouncycastle	678K	507	3,186	677K	507	3,186	31	47	0	0	1.37
location	6.3K	4	56	6.3K	4	56	0	0	0	0	0.01
core	4.1M	3,009	27,952	4.1M	3,017	28,093	353	1264	8	141	17.04
core-junit	21K	19	142	21K	19	142	0	0	0	0	0.05
ext	1.2M	960	6,896	1.2M	960	6896	209	468	0	0	40.01
framework	6.6M	3,924	38,283	9.9M	5,423	52,566	1,198	4,556	1,504	14,290	144.70
ime	5.6K	1	10	5.6K	1	10	0	0	0	0	0.03
input	3.6K	1	7	3.6K	1	7	0	0	0	0	0.01
javax	53K	24	164	53K	24	164	2	9	0	0	6.55
monkey	79K	50	237	76K	50	237	1	1	0	0	0.17
pm	25K	7	43	25K	7	43	1	2	0	0	0.09
services	1.3M	437	4,014	1.7M	531	5,153	124	384	95	1,139	159.86
sqlite-jdbc	130K	29	858	130K	29	858	3	16	0	0	2.49
svc	7K	6	26	7.3K	6	26	1	1	0	0	0.03



Changes to the Framework

Name	Base			Phone			Modified		New		Time
	Size	Class#	Method#	Size	Class#	Method#	Class#	Method#	Class#	Method#	
am	25K	6	38	26K	6	39	1	1	0	1	0.10
policy	179K	96	833	258K	140	1,200	29	105	44	368	10.23
test.runner	172K	105	1,001	172K	105	1,002	2	1	0	1	0.17
bmgr	12K	2	25	14K	2	27	1	2	0	2	0.05
bouncycastle	678K	507	3,186	677K	507	3,186	31	47	0	0	1.37
location	6.3K	4	56	6.3K	4	56	0	0	0	0	0.01
core	4.1M	3,009	27,952	4.1M	3,017	28,093	353	1264	8	141	17.04
core-junit	21K	19	142	21K	19	142	0	0	0	0	0.05
ext	1.2M	960	6,896	1.2M	960	6896	209	468	0	0	40.01
framework	6.6M	3,924	38,283	9.9M	5,423	52,566	1,198	4,556	1,504	14,290	144.70
ime	5.6K	1	10	5.6K	1	10	0	0	0	0	0.03
input	3.6K	1	7	3.6K	1	7	0	0	0	0	0.01
javax	53K	24	164	53K	24	164	2	9	0	0	6.55
monkey	79K	50	237	76K	50	237	1	1	0	0	0.17
pm	25K	7	43	25K	7	43	1	2	0	0	0.09
services	1.3M	437	4,014	1.7M	531	5,153	124	384	95	1,139	159.86
sqlite-jdbc	130K	29	858	130K	29	858	3	16	0	0	2.49
svc	7K	6	26	7.3K	6	26	1	1	0	0	0.03



Changes to the Framework

Name	Base			Phone			Modified		New		Time
	Size	Class#	Method#	Size	Class#	Method#	Class#	Method#	Class#	Method#	
am	25K	6	38	26K	6	39	1	1	0	1	0.10
policy	179K	96	833	258K	140	1,200	29	105	44	368	10.23
test.runner	172K	105	1,001	172K	105	1,002	2	1	0	1	0.17
bmgr	12K	2	25	14K	2	27	1	2	0	2	0.05
bouncycastle	678K	507	3,186	677K	507	3,186	31	47	0	0	1.37
location	6.3K	4	56	6.3K	4	56	0	0	0	0	0.01
core	4.1M	3,009	27,952	4.1M	3,017	28,093	353	1264	8	141	17.04
core-junit	21K	19	142	21K	19	142	0	0	0	0	0.05
ext	1.2M	960	6,896	1.2M	960	6896	209	468	0	0	40.01
framework	6.6M	3,924	38,283	9.9M	5,423	52,566	1,198	4,556	1,504	14,290	144.70
ime	5.6K	1	10	5.6K	1	10	0	0	0	0	0.03
input	3.6K	1	7	3.6K	1	7	0	0	0	0	0.01
javax	53K	24	164	53K	24	164	2	9	0	0	6.55
monkey	79K	50	237	76K	50	237	1	1	0	0	0.17
pm	25K	7	43	25K	7	43	1	2	0	0	0.09
services	1.3M	437	4,014	1.7M	531	5,153	124	384	95	1,139	159.86
sqlite-jdbc	130K	29	858	130K	29	858	3	16	0	0	2.49
svc	7K	6	26	7.3K	6	26	1	1	0	0	0.03



Changes to the Framework

Name	Base			Phone			Modified		New		Time
	Size	Class#	Method#	Size	Class#	Method#	Class#	Method#	Class#	Method#	
am	25K	6	38	26K	6	39	1	1	0	1	0.10
policy	179K	96	833	258K	140	1,200	29	105	44	368	10.23
test.runner	172K	105	1,001	172K	105	1,002	2	1	0	1	0.17
bmgr	12K	2	25	14K	2	27	1	2	0	2	0.05
bouncycastle	678K	507	3,186	677K	507	3,186	31	47	0	0	1.37
location	6.3K	4	56	6.3K	4	56	0	0	0	0	0.01
core	4.1M	3,009	27,952	4.1M	3,017	28,093	353	1264	8	141	17.04
core-junit	21K	19	142	21K	19	142	0	0	0	0	0.05
ext	1.2M	960	6,896	1.2M	960	6896	209	468	0	0	40.01
framework	6.6M	3,924	38,283	9.9M	5,423	52,566	1,198	4,556	1,504	14,290	144.70
ime	5.6K	1	10	5.6K	1	10	0	0	0	0	0.03
input	3.6K	1	7	3.6K	1	7	0	0	0	0	0.01
javax	53K	24	164	53K	24	164	2	9	0	0	6.55
monkey	79K	50	237	76K	50	237	1	1	0	0	0.17
pm	25K	7	43	25K	7	43	1	2	0	0	0.09
services	1.3M	437	4,014	1.7M	531	5,153	124	384	95	1,139	159.86
sqlite-jdbc	130K	29	858	130K	29	858	3	16	0	0	2.49
svc	7K	6	26	7.3K	6	26	1	1	0	0	0.03



Changes to the Framework

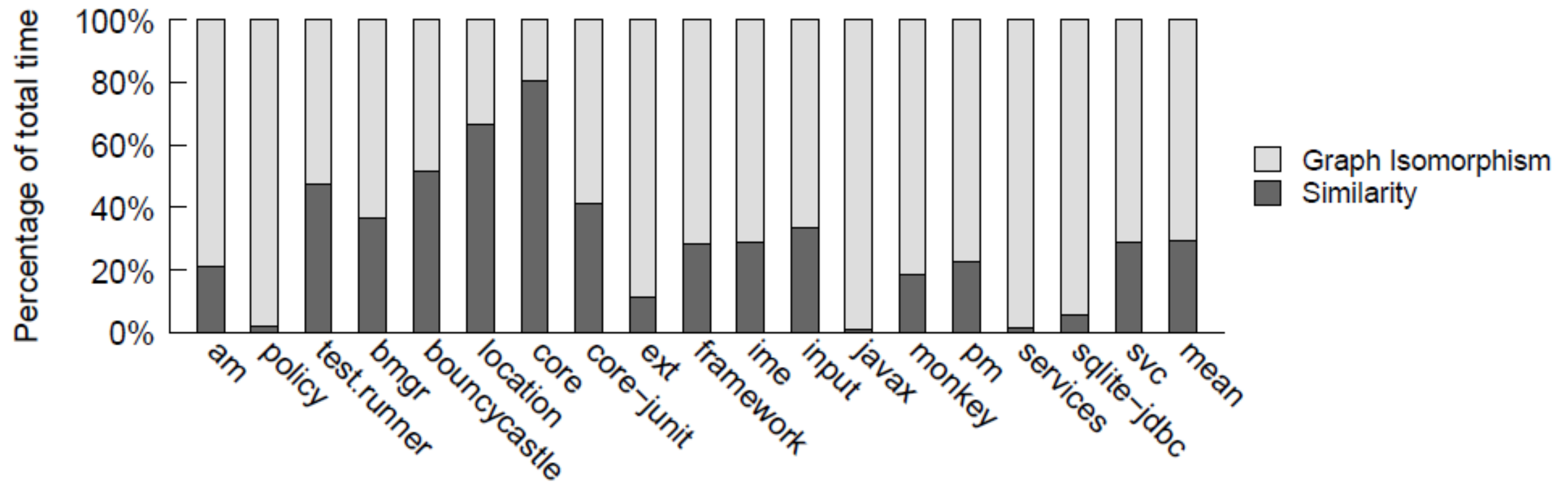
Name	Base			Phone			Modified		New		Time
	Size	Class#	Method#	Size	Class#	Method#	Class#	Method#	Class#	Method#	
am	25K	6	38	26K	6	39	1	1	0	1	0.10
policy	179K	96	833	258K	140	1,200	29	105	44	368	10.23
test.runner	172K	105	1,001	172K	105	1,002	2	1	0	1	0.17
bmgr	12K	2	25	14K	2	27	1	2	0	2	0.05
bouncycastle	678K	507	3,186	677K	507	3,186	31	47	0	0	1.37
location	6.3K	4	56	6.3K	4	56	0	0	0	0	0.01
core	4.1M	3,009	27,952	4.1M	3,017	28,093	353	1264	8	141	17.04
core-junit	21K	19	142	21K	19	142	0	0	0	0	0.05
ext	1.2M	960	6,896	1.2M	960	6896	209	468	0	0	40.01
framework	6.6M	3,924	38,283	9.9M	5,423	52,566	1,198	4,556	1,504	14,290	144.70
ime	5.6K	1	10	5.6K	1	10	0	0	0	0	0.03
input	3.6K	1	7	3.6K	1	7	0	0	0	0	0.01
javax	53K	24	164	53K	24	164	2	9	0	0	6.55
monkey	79K	50	237	76K	50	237	1	1	0	0	0.17
pm	25K	7	43	25K	7	43	1	2	0	0	0.09
services	1.3M	437	4,014	1.7M	531	5,153	124	384	95	1,139	159.86
sqlite-jdbc	130K	29	858	130K	29	858	3	16	0	0	2.49
svc	7K	6	26	7.3K	6	26	1	1	0	0	0.03



Changes to the Framework

Name	Base			Phone			Modified		New		Time
	Size	Class#	Method#	Size	Class#	Method#	Class#	Method#	Class#	Method#	
am	25K	6	38	26K	6	39	1	1	0	1	0.10
policy	179K	96	833	258K	140	1,200	29	105	44	368	10.23
test.runner	172K	105	1,001	172K	105	1,002	2	1	0	1	0.17
bmgr	12K	2	25	14K	2	27	1	2	0	2	0.05
bouncycastle	678K	507	3,186	677K	507	3,186	31	47	0	0	1.37
location	6.3K	4	56	6.3K	4	56	0	0	0	0	0.01
core	4.1M	3,009	27,952	4.1M	3,017	28,093	353	1264	8	141	17.04
core-junit	21K	19	142	21K	19	142	0	0	0	0	0.05
ext	1.2M	960	6,896	1.2M	960	6896	209	468	0	0	40.01
framework	6.6M	3,924	38,283	9.9M	5,423	52,566	1,198	4,556	1,504	14,290	144.70
ime	5.6K	1	10	5.6K	1	10	0	0	0	0	0.03
input	3.6K	1	7	3.6K	1	7	0	0	0	0	0.01
javax	53K	24	164	53K	24	164	2	9	0	0	6.55
monkey	79K	50	237	76K	50	237	1	1	0	0	0.17
pm	25K	7	43	25K	7	43	1	2	0	0	0.09
services	1.3M	437	4,014	1.7M	531	5,153	124	384	95	1,139	159.86
sqlite-jdbc	130K	29	858	130K	29	858	3	16	0	0	2.49
svc	7K	6	26	7.3K	6	26	1	1	0	0	0.03

Processing Time



Carrier IQ Hooks in Browser

Hook	Methods (Hooking Points)
callBrowserStopLoading	BrowserActivity.stopLoading
callOnPageStarted	Tab\$2.onPageStarted
callOnProgressChanged	Tab\$3.onProgressChanged
callOnReceivedError	Tab\$2.onReceivedError
callOnReceivedTitle	Tab\$3.onReceivedTitle
callReloadPage	BrowserActivity.onOptionsItemSelected BrowserActivity.resumeBrowser BrowserActivity\$StopLoadingPageTimer.ResumeLoadingPage htc.ui.HtcTitleBar.onClick
callUserCancel	BrowserActivity.onKeyUp BrowserActivity.onOptionsItemSelected

Carrier IQ Hooks in Browser

Hook	Methods (Hooking Points)
callBrowserStopLoading	BrowserActivity.stopLoading
callOnPageStarted	Tab\$2.onPageStarted
callOnProgressChanged	Tab\$3.onProgressChanged
callOnReceivedError	Tab\$2.onReceivedError
callOnReceivedTitle	Tab\$3.onReceivedTitle
callReloadPage	BrowserActivity.onOptionsItemSelected BrowserActivity.resumeBrowser BrowserActivity\$StopLoadingPageTimer.ResumeLoadingPage htc.ui.HtcTitleBar.onClick
callUserCancel	BrowserActivity.onKeyUp BrowserActivity.onOptionsItemSelected

Carrier IQ Hooks in Browser

Hook	Methods (Hooking Points)
callBrowserStopLoading	BrowserActivity.stopLoading
callOnPageStarted	Tab\$2.onPageStarted
callOnProgressChanged	Tab\$3.onProgressChanged
callOnReceivedError	Tab\$2.onReceivedError
callOnReceivedTitle	Tab\$3.onReceivedTitle
callReloadPage	BrowserActivity.onOptionsItemSelected BrowserActivity.resumeBrowser BrowserActivity\$StopLoadingPageTimer.ResumeLoadingPage htc.ui.HtcTitleBar.onClick
callUserCancel	BrowserActivity.onKeyUp BrowserActivity.onOptionsItemSelected



Vulnerable Device Management App

- Three new APIs in framework.dex
 - void **broadcastKeyEvent** (boolean); void **broadcastMotionEvent** (boolean); void **broadcastTrackballEvent**(boolean);



Vulnerable Device Management App

- Three new APIs in framework.dex
 - void **broadcastKeyEvent** (boolean); void **broadcastMotionEvent** (boolean); void **broadcastTrackballEvent**(boolean);
- Protected by vendor permissions
 - e.g., com.htc.Manifest.permission.BROADCAST_KEYIN_EVENT



Vulnerable Device Management App

- Three new APIs in framework.dex
 - void **broadcastKeyEvent** (boolean); void **broadcastMotionEvent** (boolean); void **broadcastTrackballEvent**(boolean);
- Protected by vendor permissions
 - e.g., com.htc.Manifest.permission.BROADCAST_KEYIN_EVENT
- Used only by **/system/app/HtcDm.apk**
 - system management app, can install/delete apps, factory reset...
 - dangerous functionalities are unimplemented yet
 - **no authentication at all**
 - any local app with network permission can send commands to it

Thank you!