FEAST
2017

ReDroid: Prioritizing Data Flows and Sinks for App Security Transformation

Ke Tian*, Gang Tan^, Daphne Yao*, Barbara Ryder*

*Department of Computer Science
Virginia Tech

^Department of CSE
Penn State University

Transformation Motivation and Challenge

THE STACK NEWS | DIRECTORY | EDUCATION | MAGAZINE | ABOUT US

SECURITY

CIA to open private 'app store' for intelligence operatives via Amazon Web Services

Martin Anderson Fri 27 Feb 2015 2.18pm

3 SHARES

Next month the US Central Intelligence Agency will launch a private app marketplace via provisioning from Amazon Web Services. The announcement was made by the CIA's Chief Information Officer Doug Wolfe at the 4th annual Cloudera Federal Forum in Virginia on Wednesday.

U.S. Department of Defense to Open Its Own App Store

1.8k SHARES Share on Facebook Share on Twitter

WHAT'S THIS?

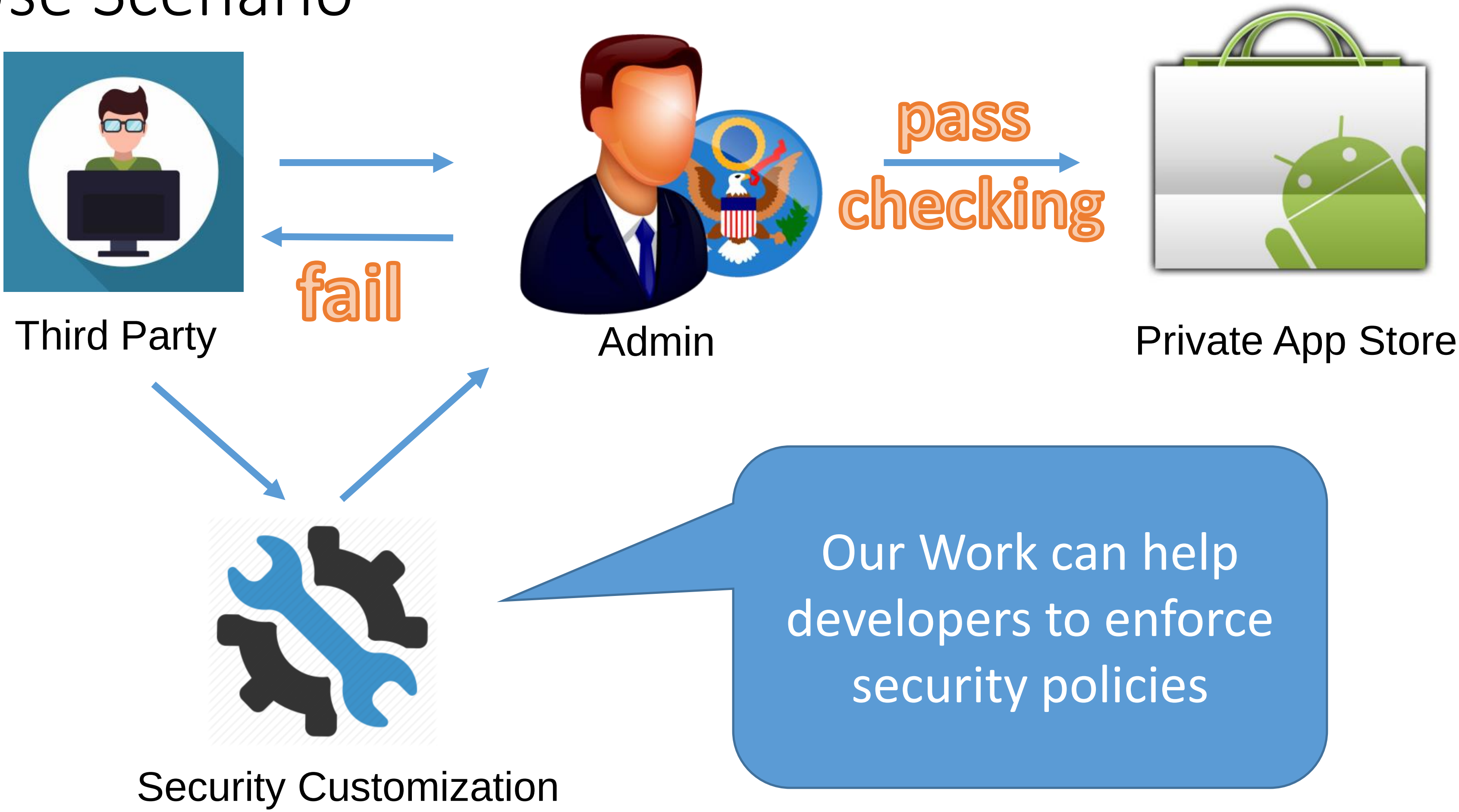
Third Party App Store:

- Security and privacy concerns
- Apps have more security restrictions

The need for application customization is growing

http://mashable.com/2013/10/30/department-of-defense-app-store/#iJuBpfyLJaq4
https://thestack.com/security/2015/02/27

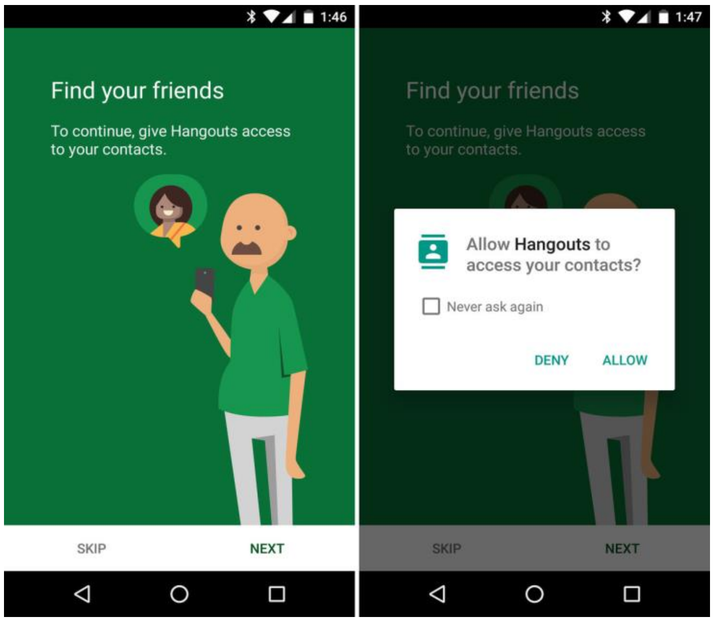
Use Scenario



Our goal is to enforce more fine-grained policies via transformation

Existing Solutions

- Google Dynamic Permission[1]



Permission-level

- RetroSkeleton[2]
- Method-level rewriting approach

method-level

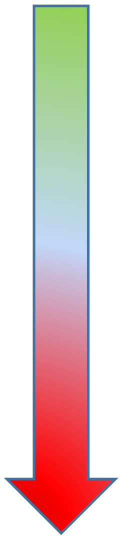
- Ours
- Sink-level
- Sink sensitivity awareness

Rewriting Granularity	RetroSkeleton [18] and [10][19]	ReDroid **(Ours)**
Package-level (Repackage)	✓	✓
Class-level (Class Inject)	✓	✓
Method-level (Method Invoc.)	✓	✓
ICC-level (Intent Redirect)	–	✓
Prioritization-based Rewriting	–	✓

[1]https://developer.android.com/training/permissions/requesting.html

[2]RetroSkeleton: Retrofitting Android Apps. In Proc. of MobiSys (2013)

A policy example: to monitor the most dangerous sink



Sink 1: add a bookmark 

Sink 2: set time clock 

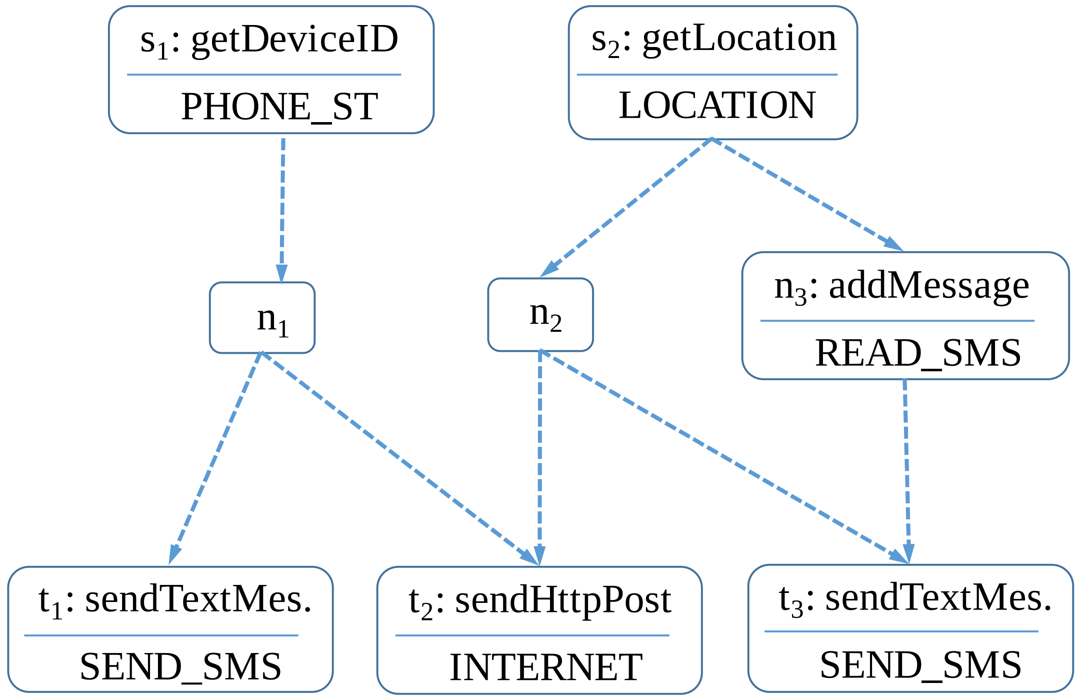
Sink 3: Write to some local storage 

Sink 4: send text messages  

Policy: sensitive privacy leakage must be prevented

- How to measure the sensitivity?
- How to enforce the security policy?

The need for sink ranking

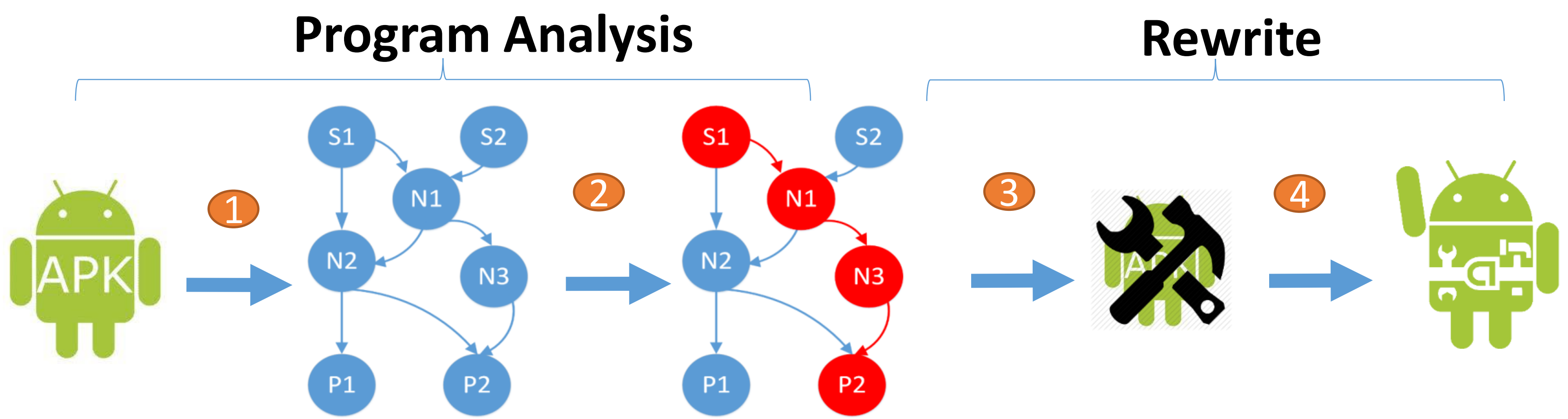


Design choices:

1. Sensitive API-based risk vs. Permission-based risk (permission -> score)
2. Sink Rewriting vs. Flow Rewriting (sink)

How to quantify risks of sinks in order to prioritize them?

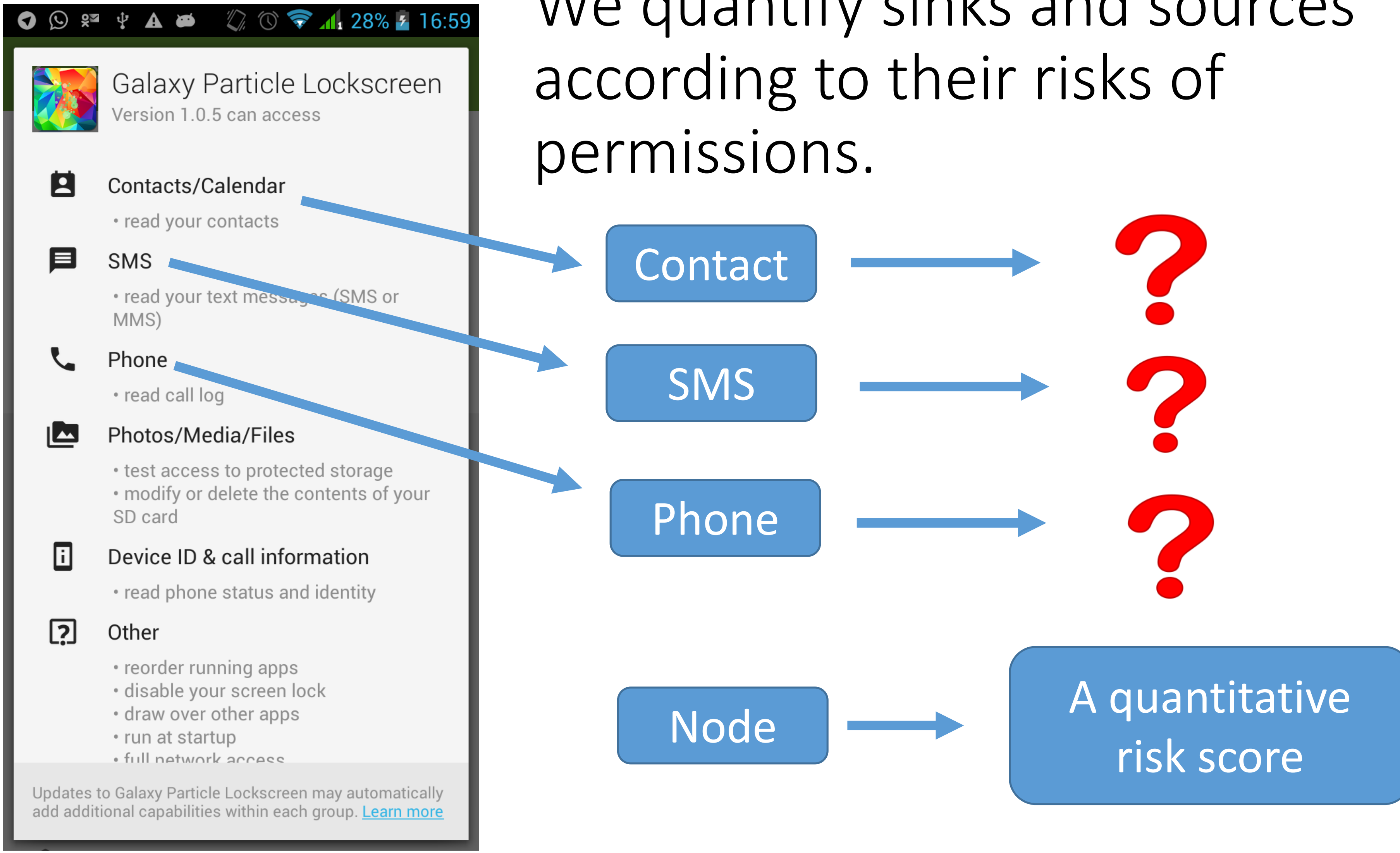
Our workflow: sink ranking + bytecode rewriting



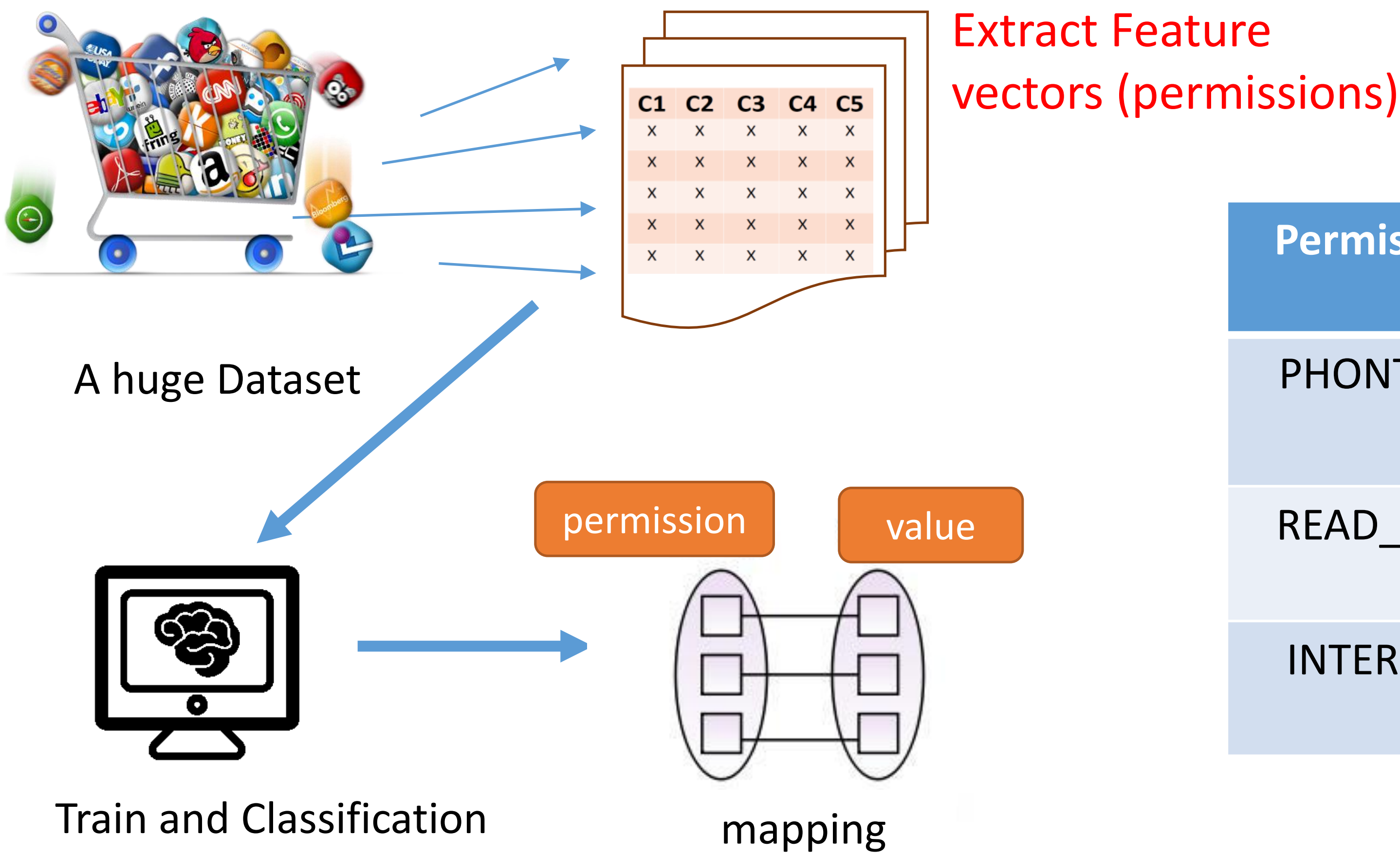
Pipeline

1. Data flow graph construction
2. **Risk score ranking and computation**
3. IR transformation
4. **Modification and recompile**

We quantify sinks and sources according to their risks of permissions.

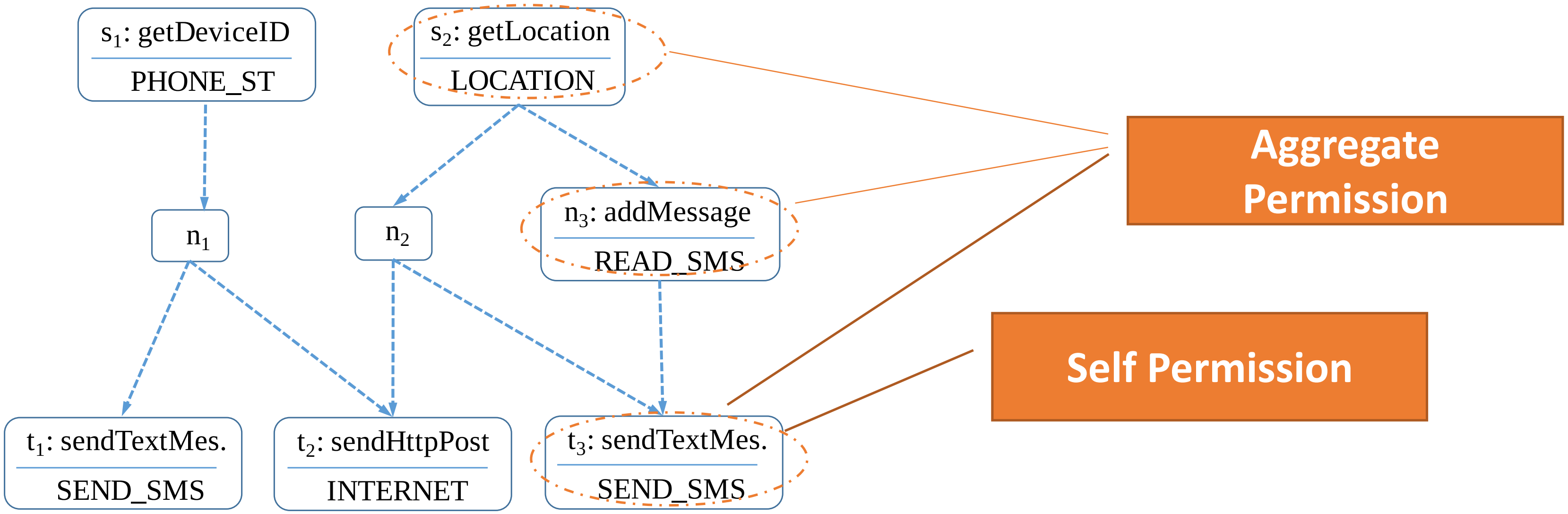


Our approach for quantifying the risks of permissions: Machine Learning



Permission	Risk Value
PHONT_ST	0.140
READ_SMS	0.107
INTERNET	0.007

Sensitivity propagations



	Self permission	Aggregate permission	Risk Score
t_1	SEND_SMS	SEND_SMS,PHONT_ST	0.189
t_2	INTERNET	INTERNET, PHONT_ST	0.147
t_3	SEND_SMS	SEND_SMS, READ_SMS, LOCATION	0.151

Evaluate and validate risk metrics

jp.co.jags, android.TigerJumping ⟵ – – ⟶ url.init() ⟵ – – ⟶ jp.Adlantis (Ads- http)

org.ohny.weekend, org.qstar.guardx and uk.org.crampton.battery ⟵ – – ⟶ execute() ⟵ – – ⟶ com.android.Flurry

Geinimi-037c*. ⟵ – – ⟶ sendTextMessage() ⟵ – – ⟶ SMS trojan

Matching with real-world security reports

T_i	T_1	T_2	T_3	T_4
C	com.ju6.a	uk.co. lilhermit. android. core.Native	com.adwo. adsdk.L	com.adwo. adsdk.i
M	a()	runcmd _wrapper()	a()	a()
F	Android.util.Log int e()			
$r(T_i)$	0.170	0.156	0.007	0

Same sink in different locations

Different risk values

Next, we describe how our bytecode transformation works on Android apps

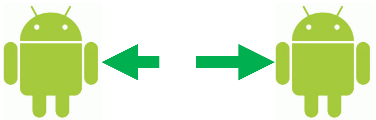
Two rewriting strategies for vetting sensitive sinks

- Passive and proactive rewriting strategies — System logs

- Proactive dynamic checking & ICC relay — App communication channel

Summary of our transformation capabilities

Type	Vulnerability	Our Framework Addresses
Inter-app Com. (IAC)	ICC hijacking	✓
	Collusion	✓
Stand-alone App	Privacy Leak	✓
	Reflection	✓
	String Obfuscation	✓
	Dynamic Code Loading	✓



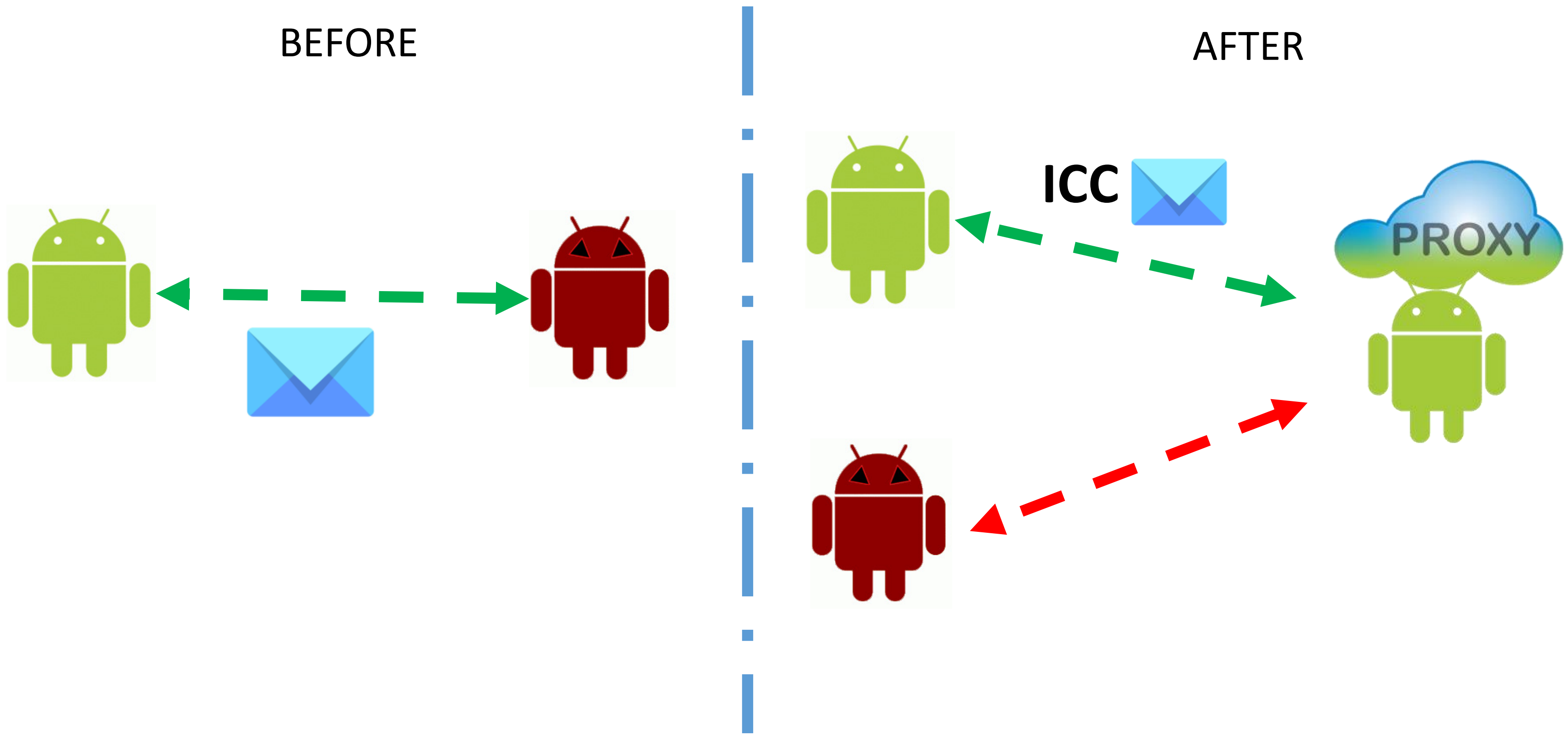
Our Advantages:

- Single app: data leak, privacy leak, detect malware.
- Inter-app Communication: data leak through communication channels

ICC (Inter-Component Communication)

- Communication mechanism in Android
- Pass data and information among Apps
- Can be used for data redirection

An example: detect ICC leakage (ICC Relay)



1. Identify sensitive data flows in an ICC (startActivity with implicit intent)
2. Redirect the intent into a secure proxy app
3. Intent is checking inside the proxy app
4. The communication is relayed from the proxy app

Evaluate rewriting performance

App Category	#of ICC Exits	Logging Success		ICC Relay Success	
ICCBench		Re.	In.	Re.	In.
icc_implicit_action	1	1	1	1	1
icc_implicit_category	1	1	1	1	1
icc_implicit_data	2	2	2	2	2
Icc_implicit_mix	3	3	3	3	3
icc_implicit_src_sink	2	2	2	2	2
icc_dynregister	2	2	2	2	2
DroidBench(IccTA)					
iac_startActivity	1	1	1	1	1
icc_startActivity	2	2	2	2	0
iac_startService	1	1	1	1	1
iac_broadCast	1	1	1	1	1
Summary	**16**	**16**	**16**	**16**	**14**

Logging based rewriting achieve high accuracy and robustness

ICC relay based failed two cases because security protection. ICC is efficient in IAC-based protection

Transformation is efficient and valid

Conclusions:

- ReDroid can prioritize and harden Android apps
- Our ML based maximum likelihood mapping from permissions to risks is general
- Experiments confirmed the accuracy of our sink ranking algorithm
- We demonstrated multiple real-world rewriting scenarios

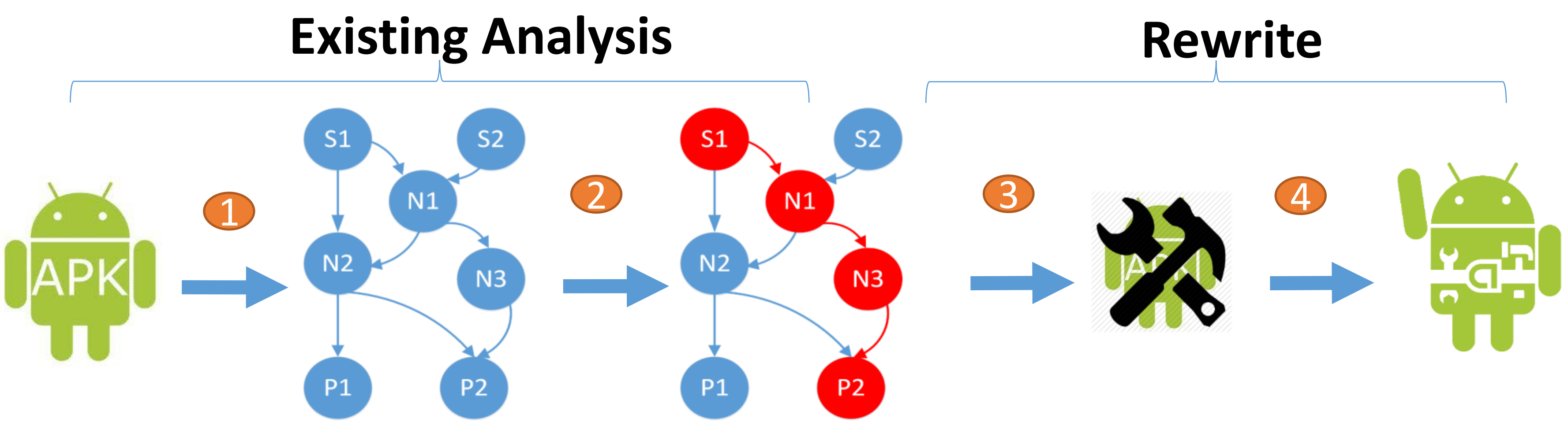
Check our tool:
https://github.com/ririhedou/AppRewriting

Thanks!

Android rewriting

- What?
  - Rewrite/Instrument Android apps (statically).
  - Edit app bytecode/IR (without source code)
  - Modify app behaviors according to security specifications (defined by analysts or users)
- Why?
  - Mitigate vulnerabilities (enhance security)
  - Easily used for security analysis (monitoring)

- How? A pipeline

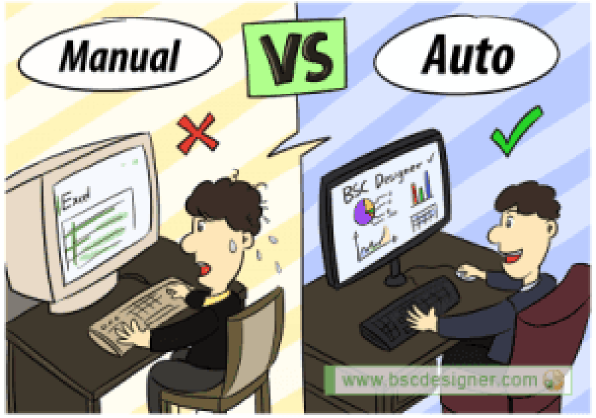


Technical Challenges

- How to identify which part needs to be transformed?

- How to keep the valid execution of apps after transformation?

- How to automate the transformation process?



http://mashable.com/2013/10/30/department-of-defense-app-store/#iJuBpfyLJaq4
https://thestack.com/security/2015/02/27

Our rewriting v.s. state-of-the-art in Android

Rewriting Granularity	RetroSkeleton and [10][19]	ReDroid **(Ours)**
Package-level (Repackage)	✓	✓
Class-level (Class Inject)	✓	✓
Method-level (Method Invoc.)	✓	✓
ICC-level (Intent Redirect)	–	✓
Prioritization-based Rewriting	–	✓

- Include Android specific components (inter-component communication)
- More fine-grained control for rewriting specifications
- Introduce flow-/sink- aware analysis to extend rewriting feasibility

- Motivations
  1. Security analysts need analyze (unknown) apps
  2. Apps are complex (# of sensitive flows & nodes)
  3. Manual analysis is time-consuming (no target)