Automated Synthesis of Semantic Malware Signatures using MaxSAT

Yu Feng

Osbert Bastani

Ruben Martins

Isil Dillig

Saswat Anand

NDSS'17, San Diego

Newly Found Malware can steal bank details on Android phones

by **Ali Raza**
9 months ago

https://www.hackread.com/malware-can-steal-bank-details-android-phones/

Hundreds Of Operations Canceled After Malware Hacks Hospitals Systems

Thursday, November 03, 2016 Mohit Kumar

G+1 94 Like 3.3K Share 1187 Tweet 529 Share 279 share 2068

http://thehackernews.com/2016/11/hospital-cyber-attack-virus.html

Android Malware Used to Hack and Steal a Tesla Car

By Catalin Cimpanu   November 25, 2016   06:05 AM   2

http://www.bleepingcomputer.com/news/security/android-malware-used-to-hack-and-steal-a-tesla-car/

Statistics

https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

http://www.mcafee.com/us/resources/reports/rp-mobile-threat-report-2016.pdf

Statistics

37M

Total count of malware detected over 6 months

https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

http://www.mcafee.com/us/resources/reports/rp-mobile-threat-report-2016.pdf

Statistics

37M
Total count of malware detected over 6 months

295
# of Android malware families by 2016

https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

http://www.mcafee.com/us/resources/reports/rp-mobile-threat-report-2016.pdf

Statistics

37M
Total count of malware detected over 6 months

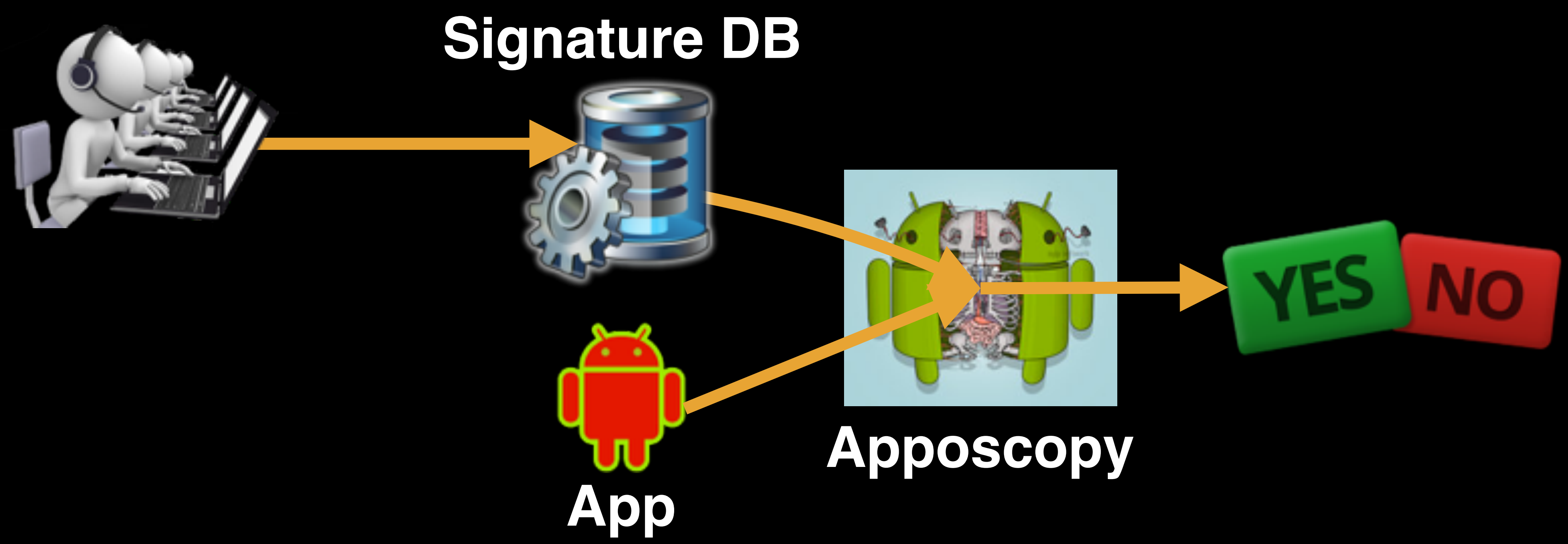
295
\# of Android malware families by 2016

Hundreds of signatures	≈	*Millions of samples*

https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

http://www.mcafee.com/us/resources/reports/rp-mobile-threat-report-2016.pdf

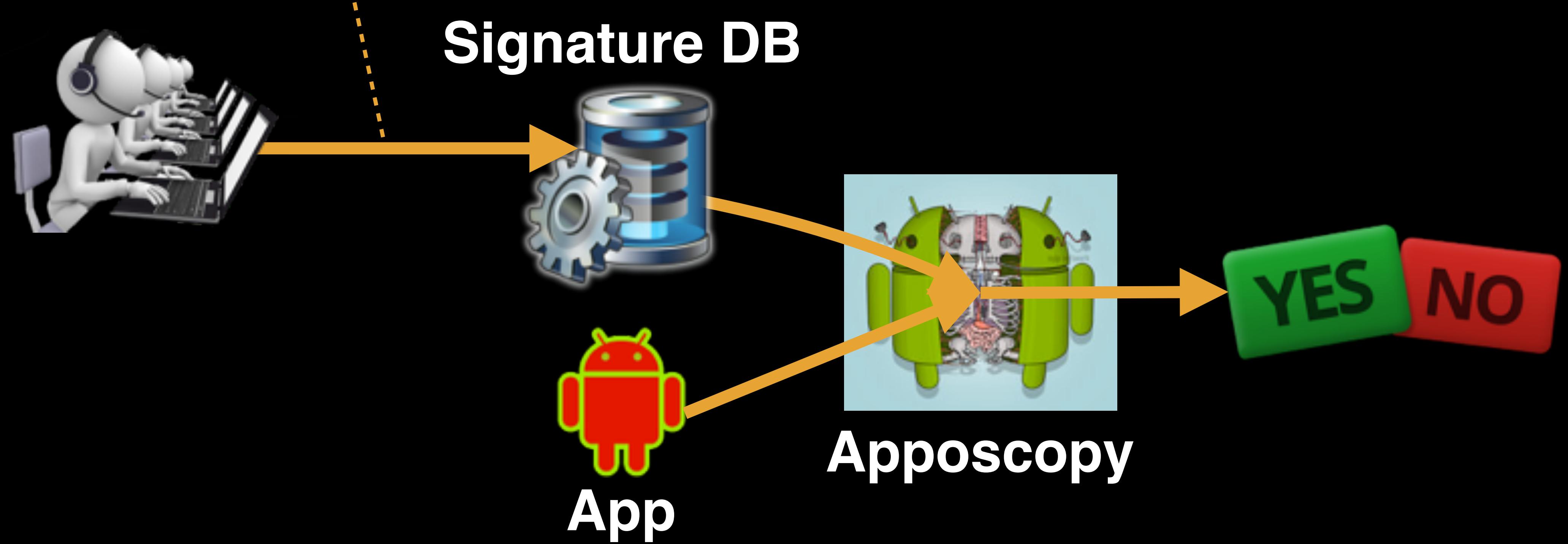
Apposcopy Overview



Feng, et al. FSE'14

Apposcopy Overview

A high-level language for describing semantic properties of malware

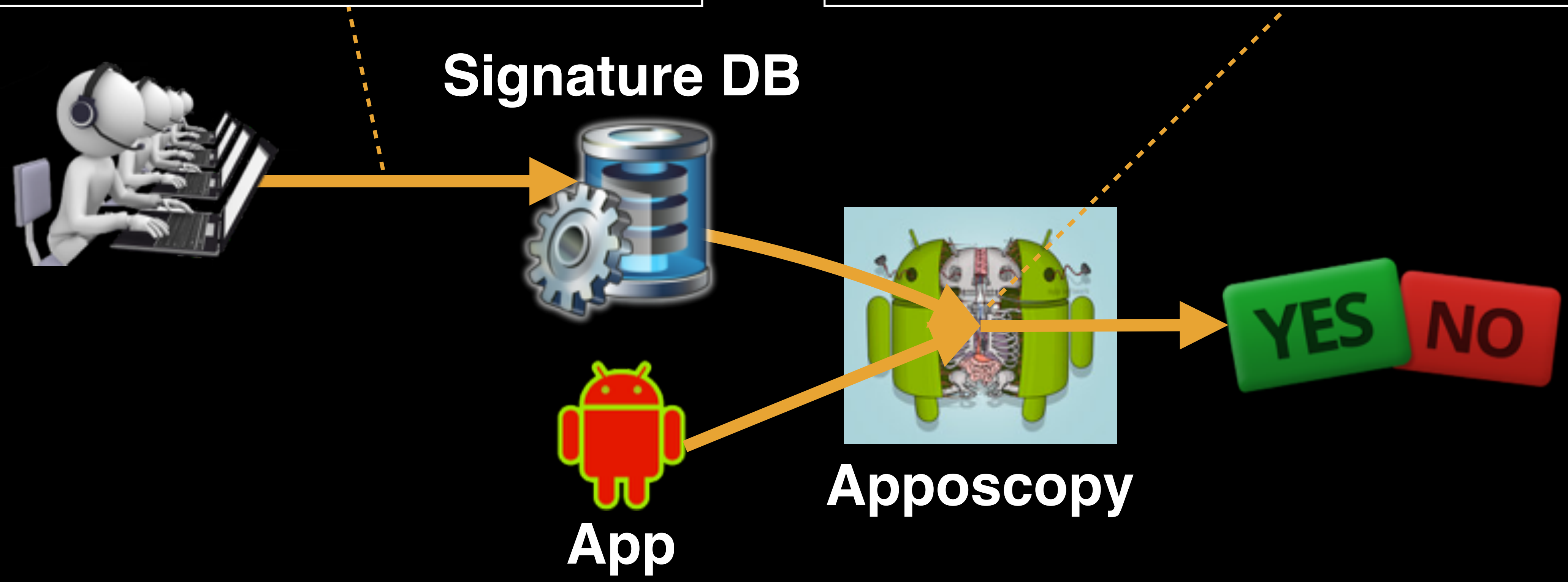


Feng, et al. FSE'14

Apposcopy Overview

A high-level language for describing semantic properties of malware

A novel static analysis for deciding if an app matches the signature of a family



Feng, et al. FSE'14

Caveats

Caveats

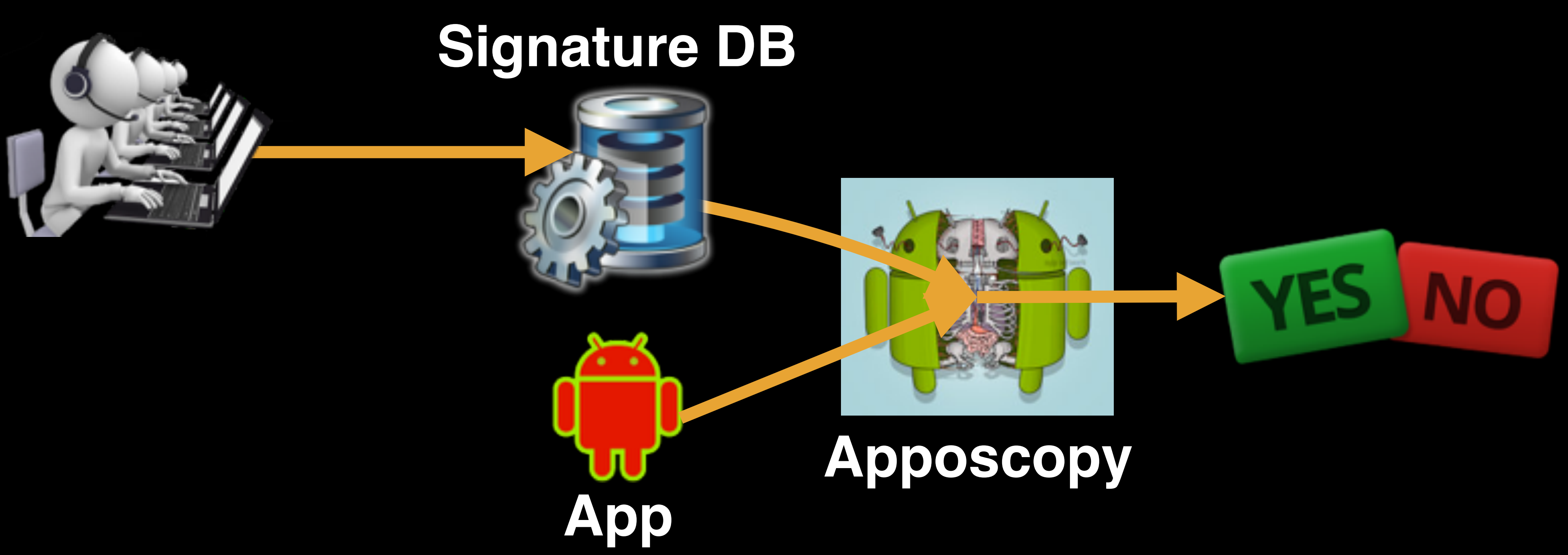
Writing signatures
is tedious

Caveats

Writing signatures
is tedious

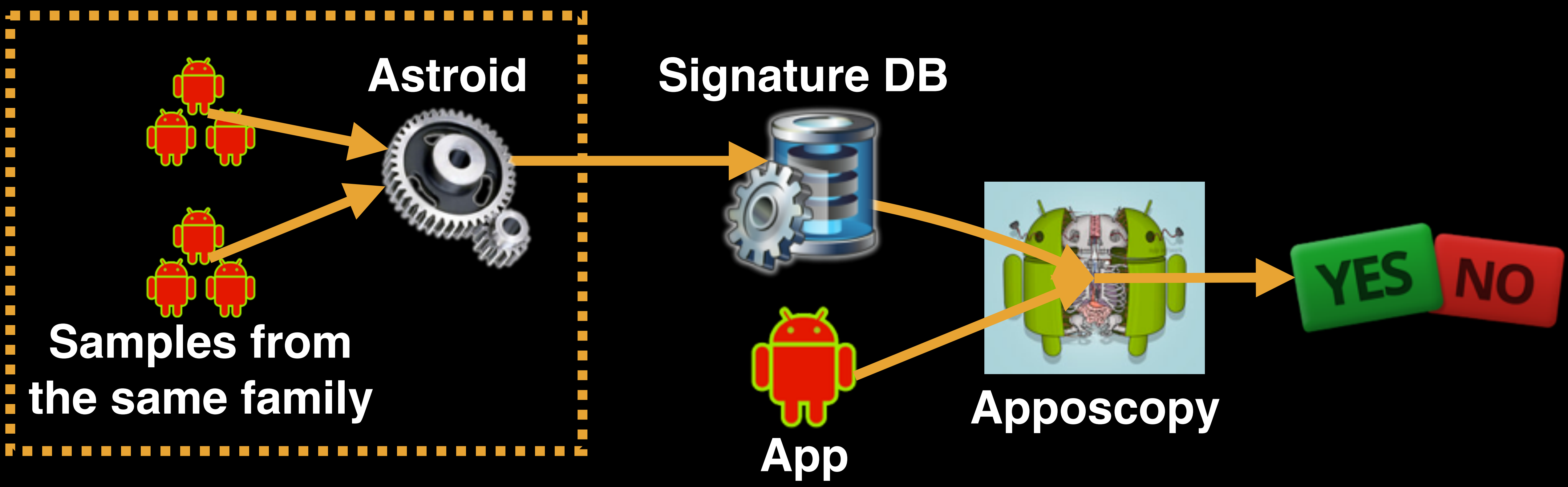
Vulnerable to
semantic obfuscation

Goal



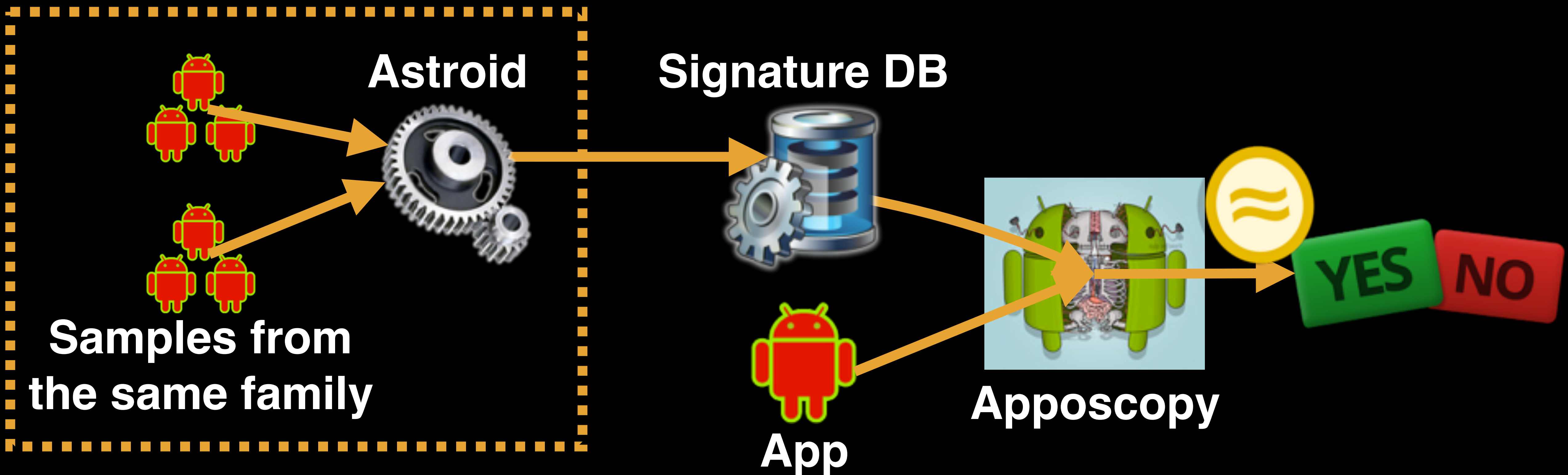
Goal

- Infer a signature from few samples of a malware family



Goal

- Infer a signature from few samples of a malware family
- Approximate matching algorithm that is resistant to semantic obfuscation



Our Signature in a Nutshell

Inter-Component Call Graph

Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell

Activity1

Inter-Component
Call Graph

Android

Feng, et al. FSE’14
Feng, et al. OOPSLA’15

Our Signature in a Nutshell

Activity1

Activity2

Inter-Component Call Graph

Android

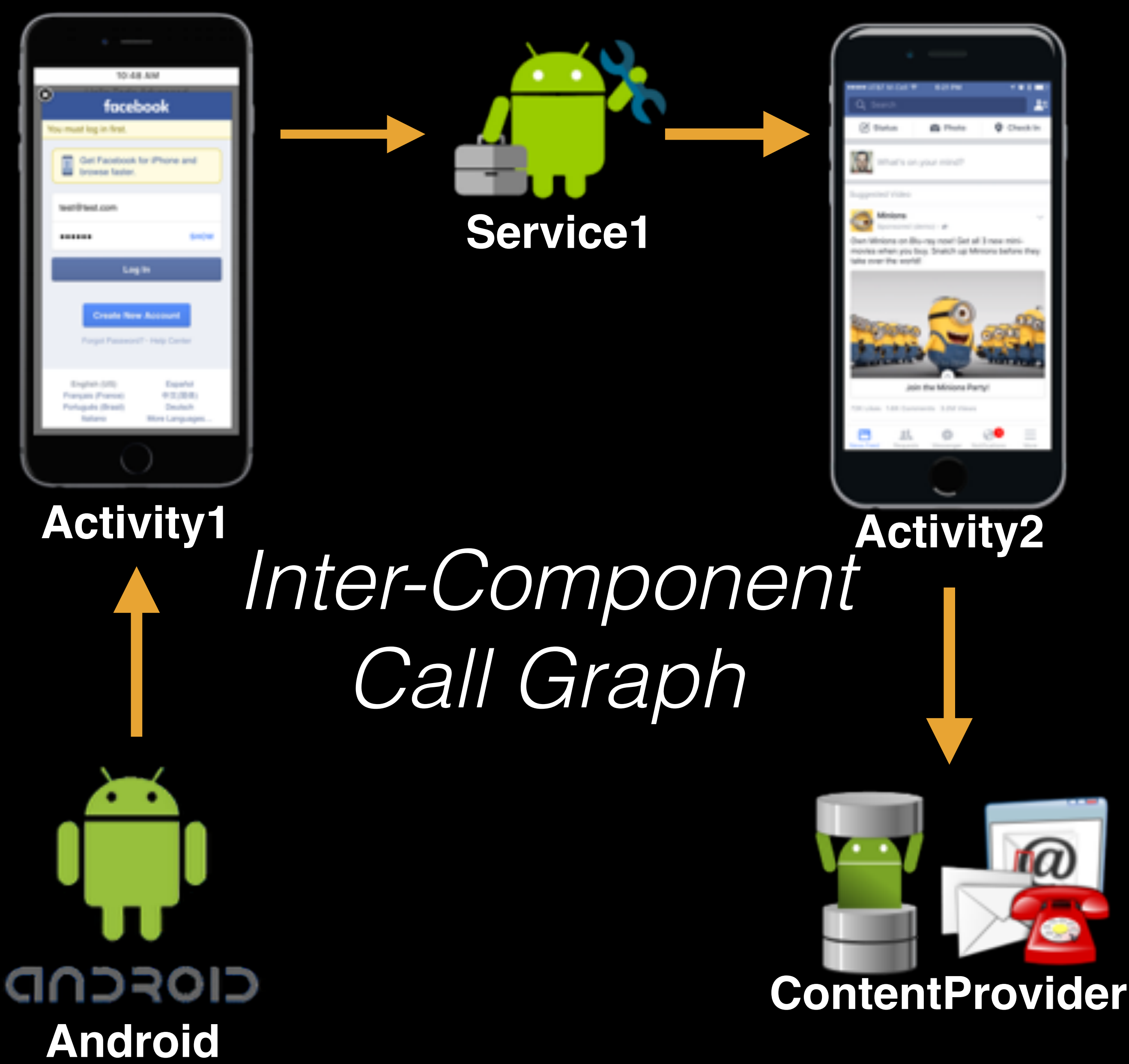
Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell

Inter-Component Call Graph

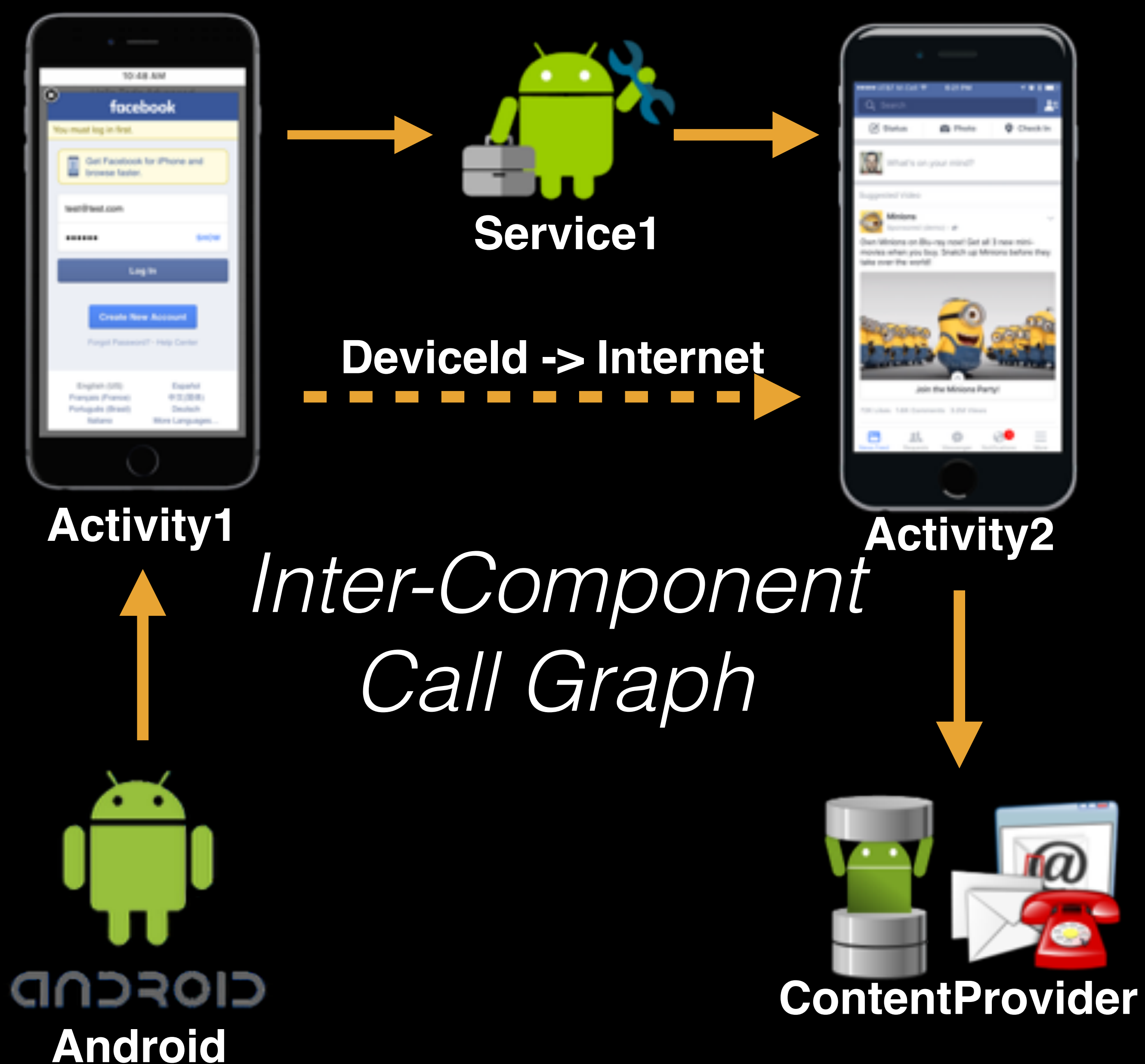
Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell



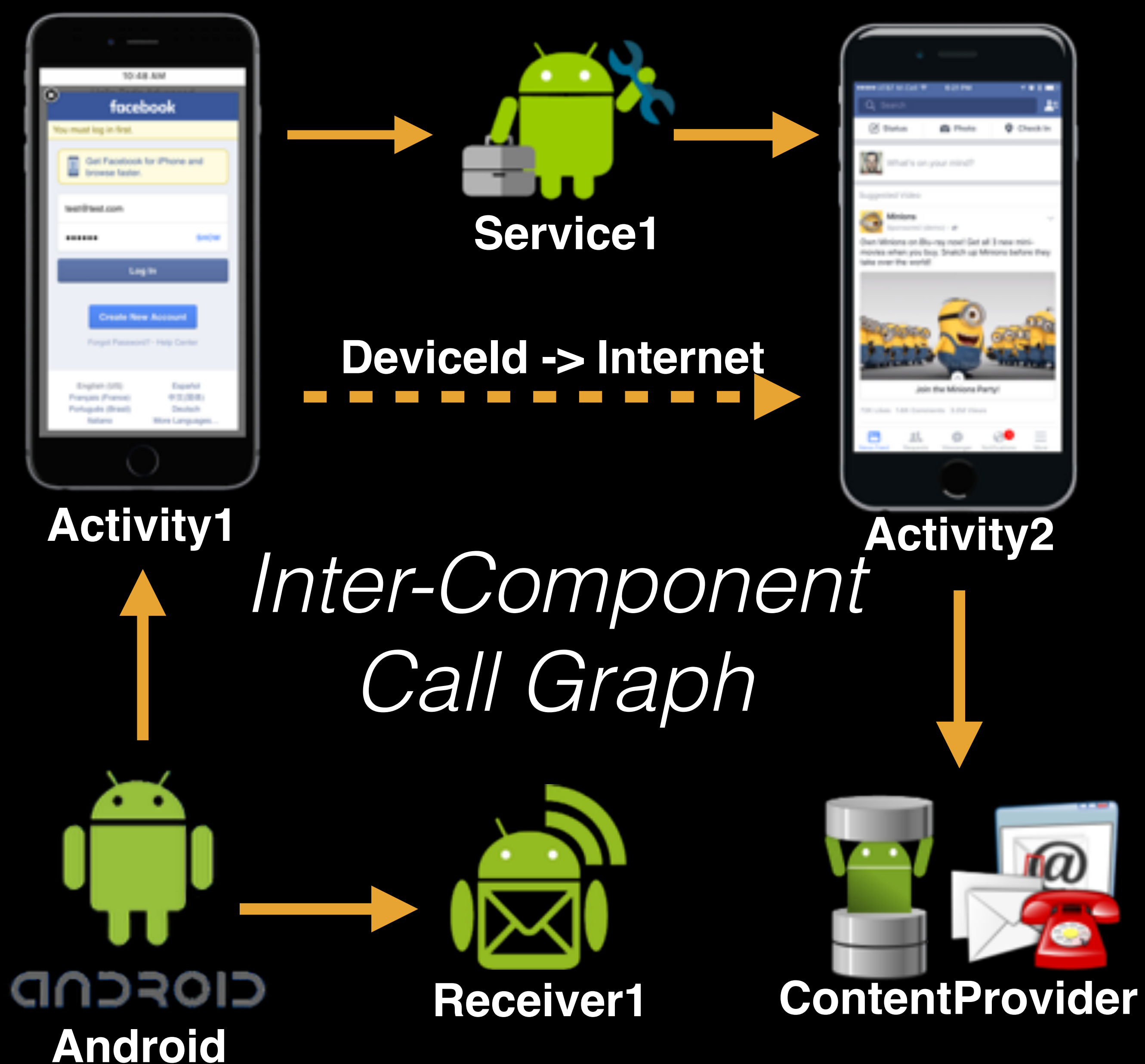
Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell



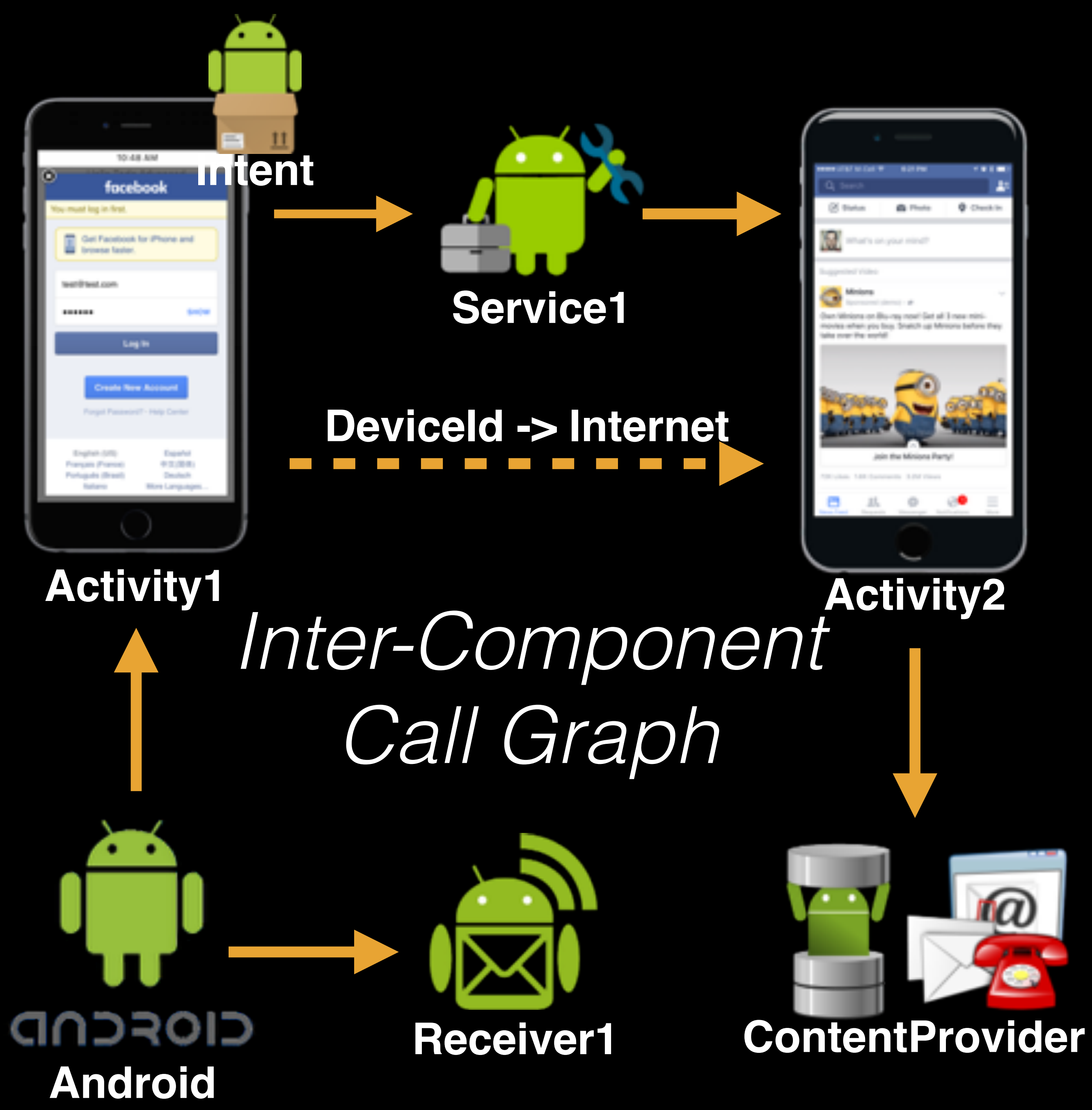
Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell



Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell

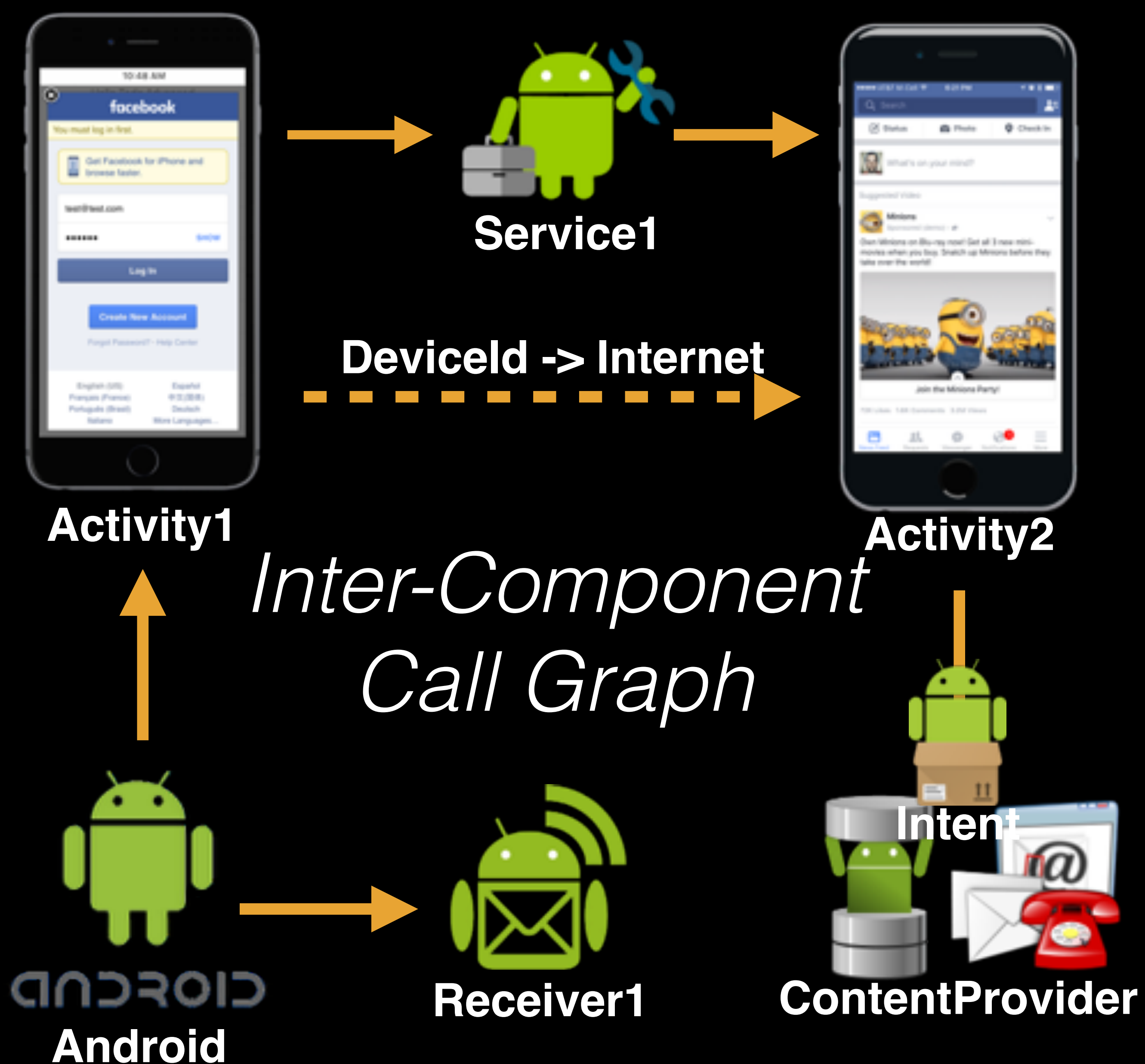


Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell

Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell



Solid line: control property
Dashed line: data property

Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell

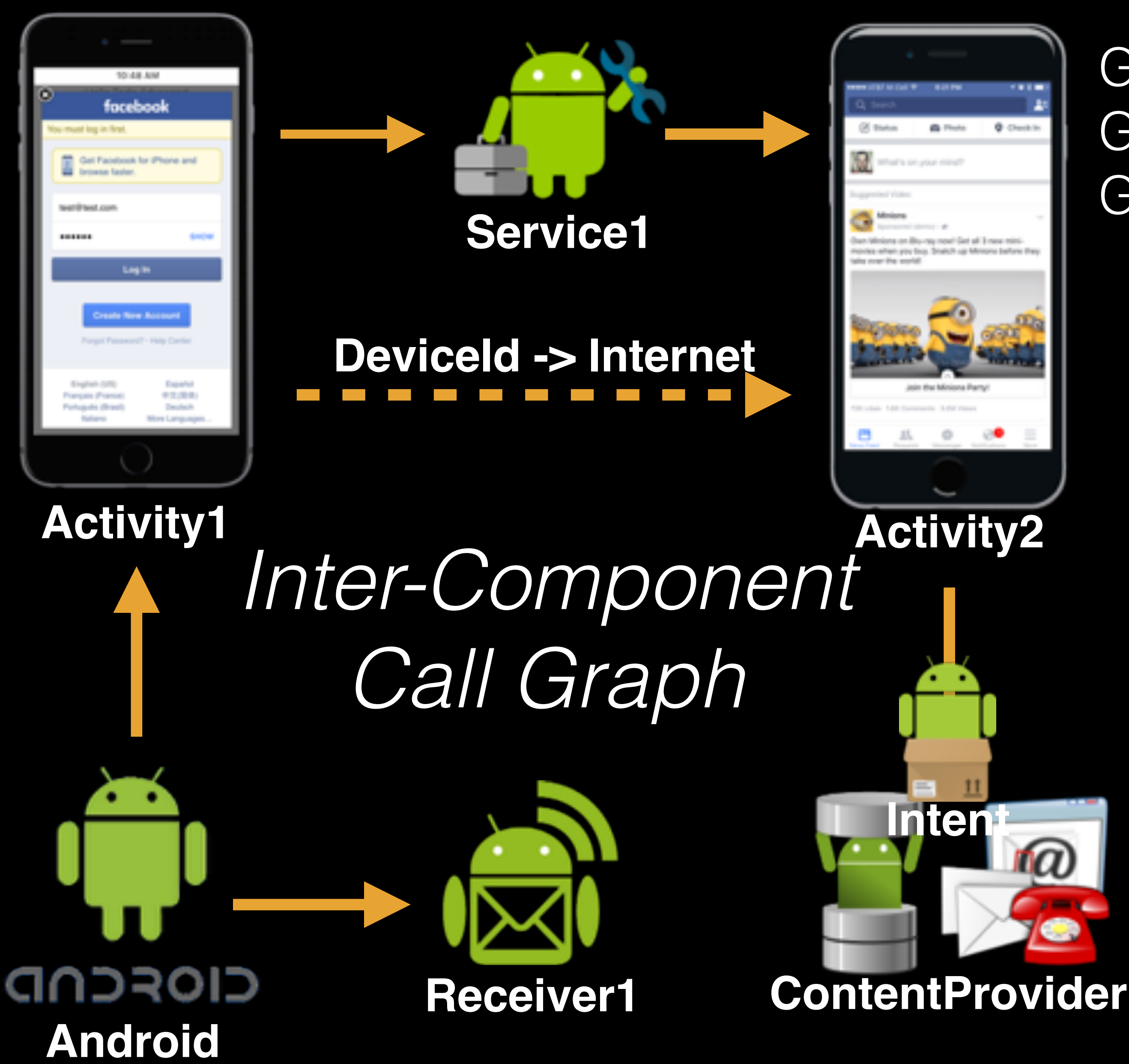
GDEvent(SMS_RECEIEVED).
GDEvent(NEW_OUTGOING_CALL).
GoldDream :- **receiver(r)**,
icc(SYSTEM, r, e, _), GDEvent(e),
service(s), **icc*(r, s)**,
flow(s, DeviceId, s, Internet),
flow(s, SubscriberId, s, Internet).

GoldDream Signature

Solid line: control property
Dashed line: data property

Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell



Component Predicate

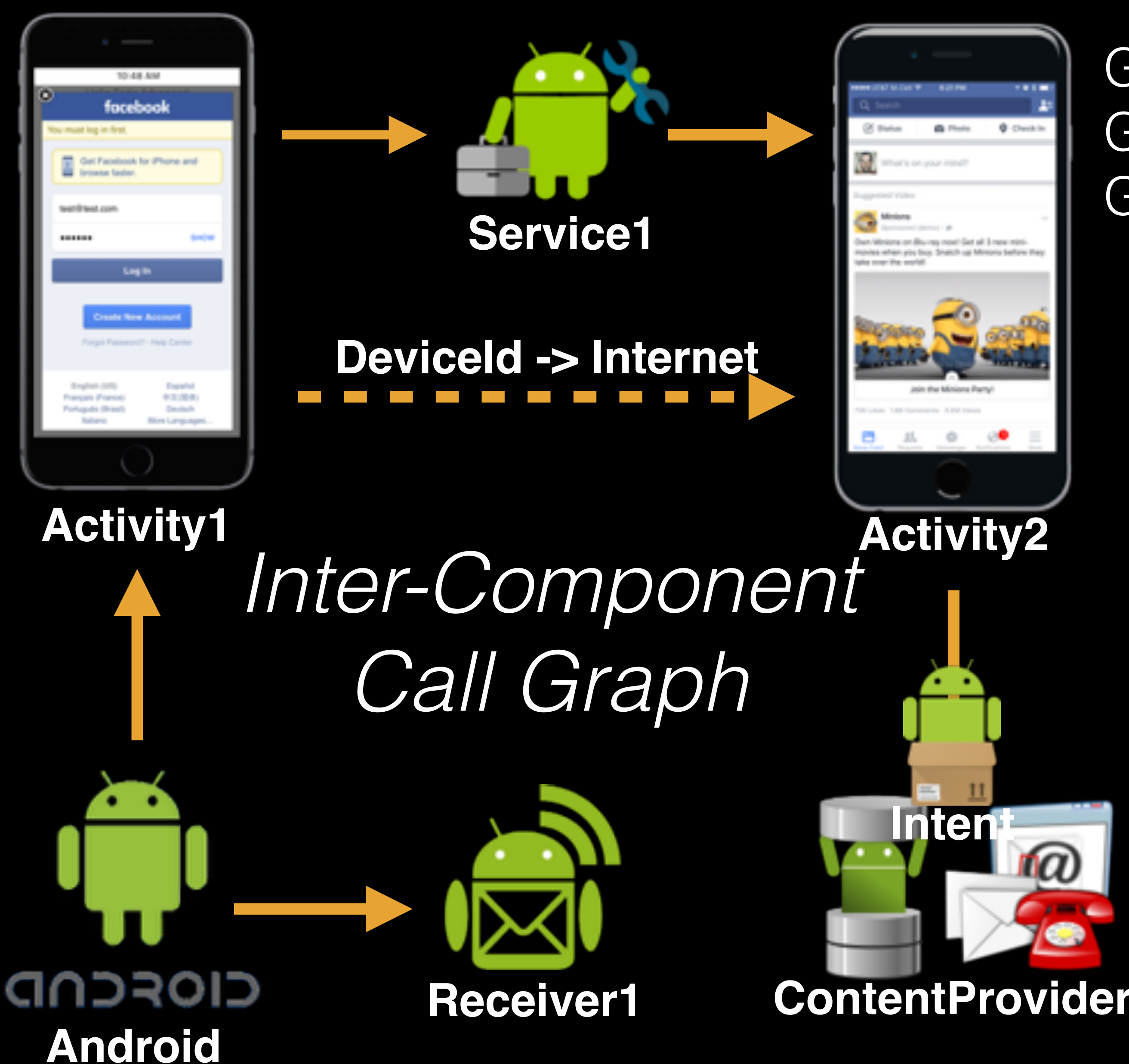
GDEvent(SMS_RECEIEVED).
GDEvent(NEW_OUTGOING_CALL).
GoldDream :- **receiver(r)**,
icc(SYSTEM, r, e, _), GDEvent(e),
service(s), **icc*(r, s)**,
flow(s, DeviceId, s, Internet),
flow(s, SubscriberId, s, Internet).

GoldDream Signature

Solid line: control property
Dashed line: data property

Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell



Inter-Component Call Graph

Component Predicate

Control Predicate

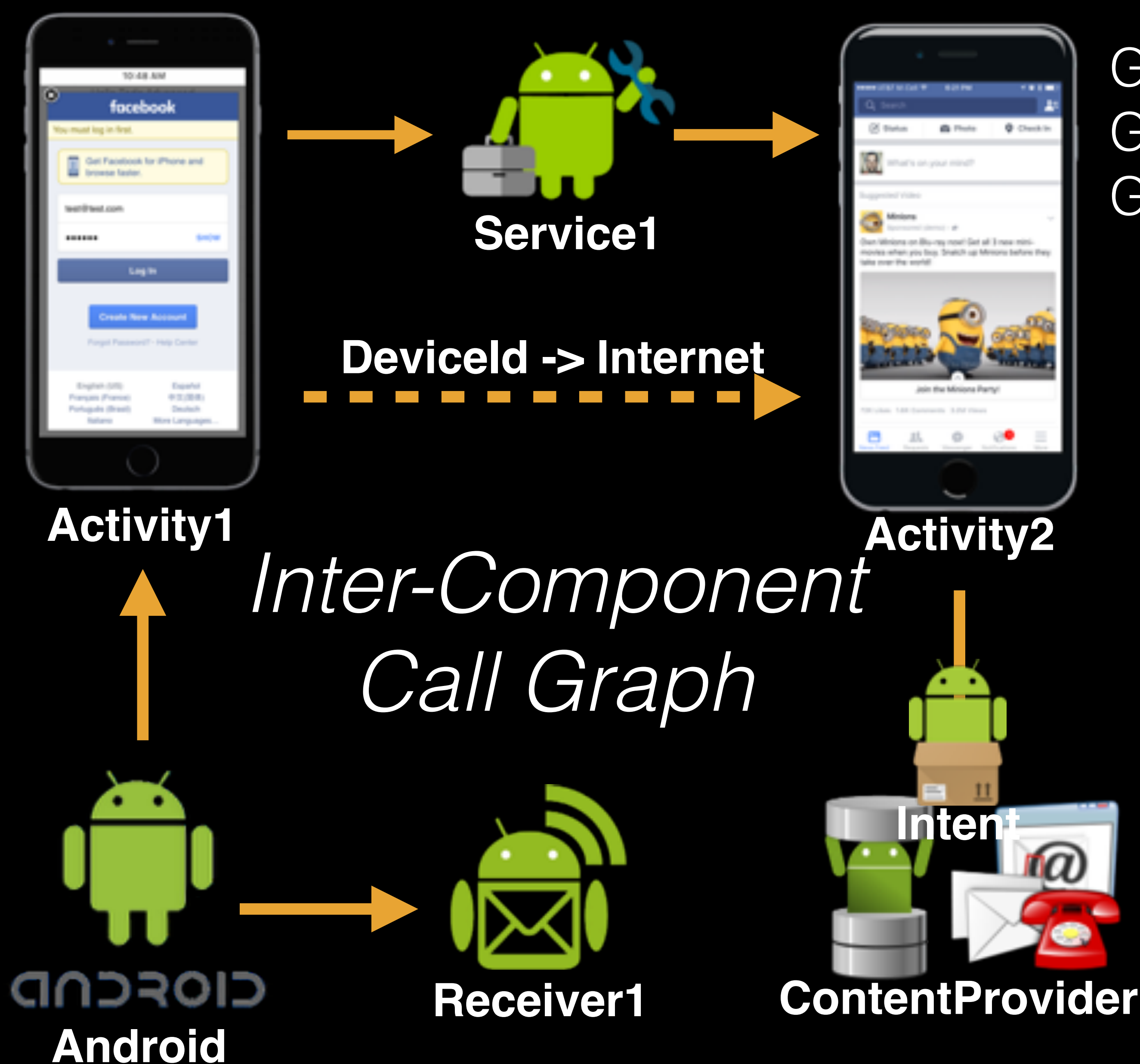
GDEvent(SMS_RECEIEVED).
GDEvent(NEW_OUTGOING_CALL).
GoldDream :- **receiver(r)**,
icc(SYSTEM, r, e, _), GDEvent(e),
service(s), **icc*(r, s)**,
flow(s, DeviceId, s, Internet),
flow(s, SubscriberId, s, Internet).

GoldDream Signature

Solid line: control property
Dashed line: data property

Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Our Signature in a Nutshell



Inter-Component Call Graph

Component Predicate

Control Predicate

Flow Predicate

GDEvent(SMS_RECEIEVED).
GDEvent(NEW_OUTGOING_CALL).
GoldDream :- **receiver(r)**,
icc(SYSTEM, r, e, _), GDEvent(e),
service(s), **icc*(r, s)**,
flow(s, DeviceId, s, Internet),
flow(s, SubscriberId, s, Internet).

GoldDream Signature

Solid line: control property
Dashed line: data property

Feng, et al. FSE'14
Feng, et al. OOPSLA'15

Signature Inference

Signature Inference

Given ***n*** malware samples from family ***F***, compute its signature ***S***

Signature Inference

Given ***n*** malware samples from family ***F***, compute its signature ***S***

Any signature that matches n samples

Signature Inference

Given ***n*** malware samples from family ***F***, compute its signature ***S***

Any signature that matches n samples

Empty signature could also be a solution!

Insight

Insight

Given ***n*** malware samples from family ***F***, compute its signature ***S***

Insight

Given ***n*** malware samples from family ***F***, compute its signature ***S***

- Our candidate ***S*** should be

Insight

Given ***n*** malware samples from family ***F***, compute its signature ***S***

- Our candidate ***S*** should be
  - A common subgraph to minimize <u>false negatives</u>

Insight

Given ***n*** malware samples from family ***F***, compute its signature ***S***

- Our candidate ***S*** should be
  - A common subgraph to minimize <u>false negatives</u>
  - Maximally suspicious to minimize <u>false positives</u>

Insight

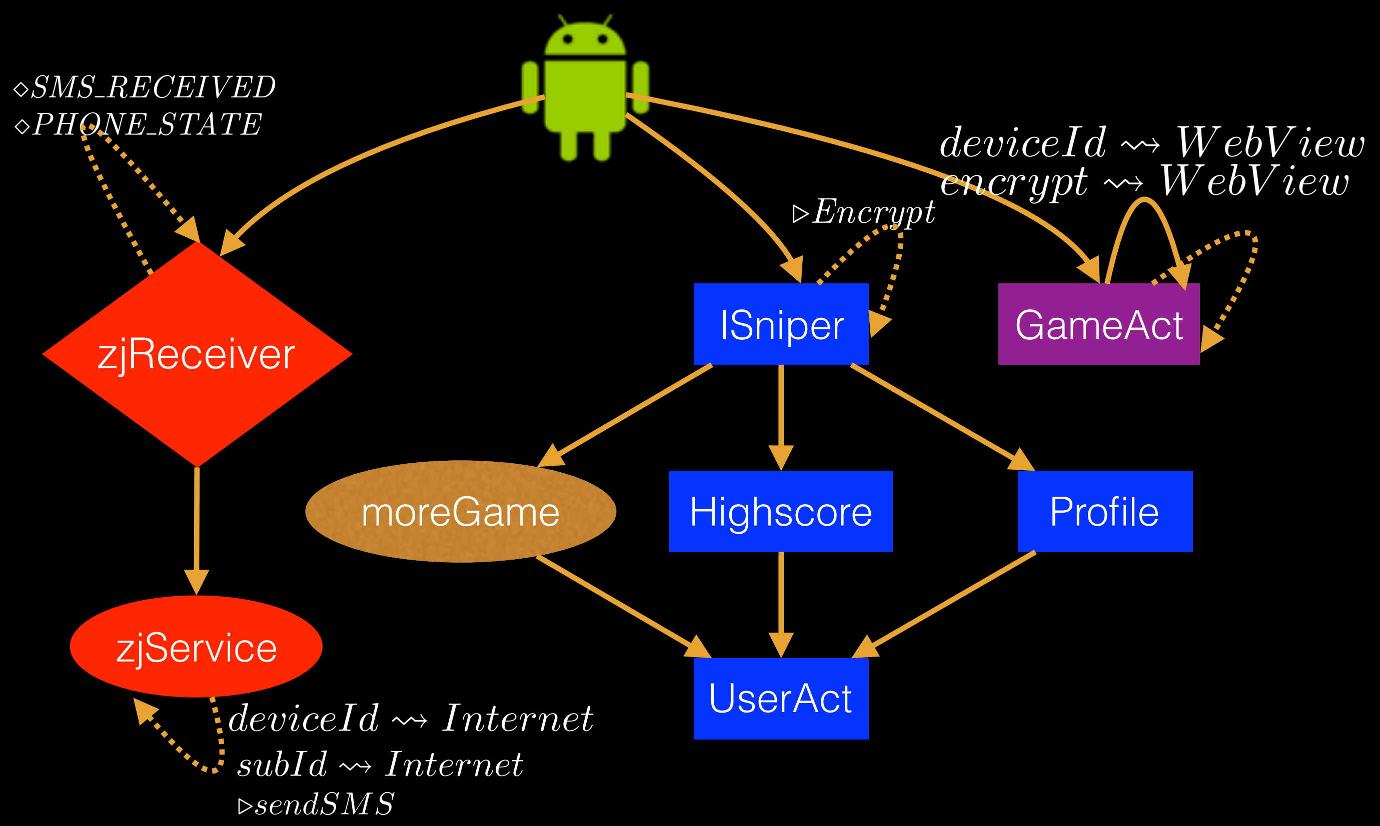
Given ***n*** malware samples from family ***F***, compute its signature ***S***

- Our candidate ***S*** should be
    - A common subgraph to minimize <u>false negatives</u>
    - Maximally suspicious to minimize <u>false positives</u>

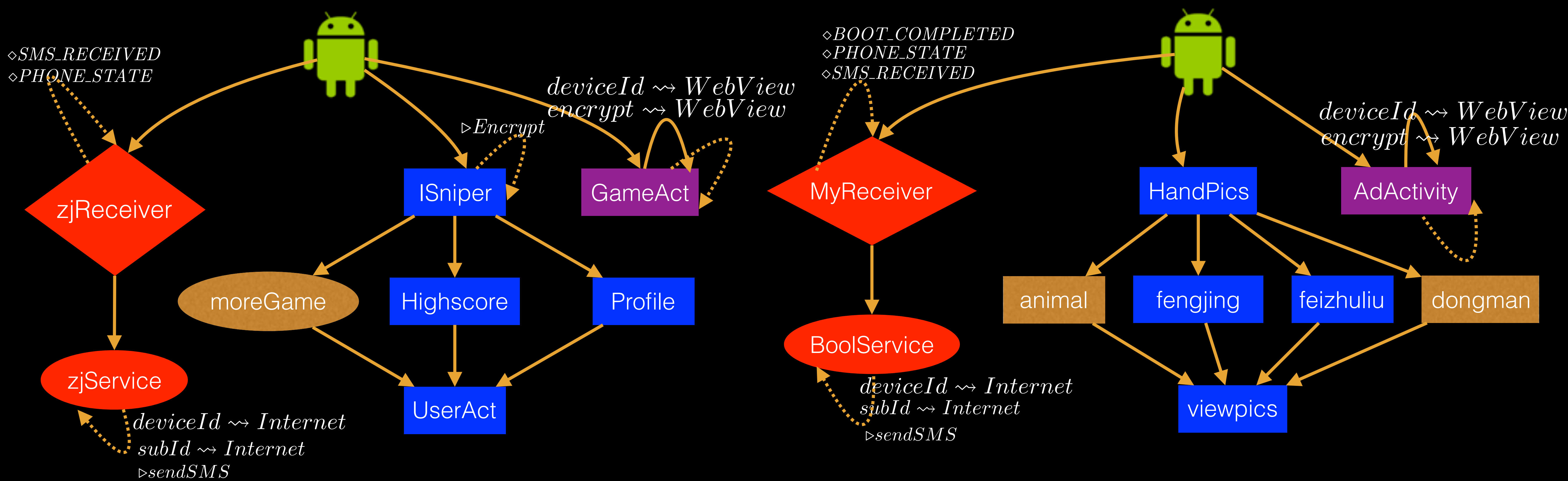
Infer signatures by finding a <u>Maximally Suspicious Common Subgraph</u> of n malware samples

Example

Example

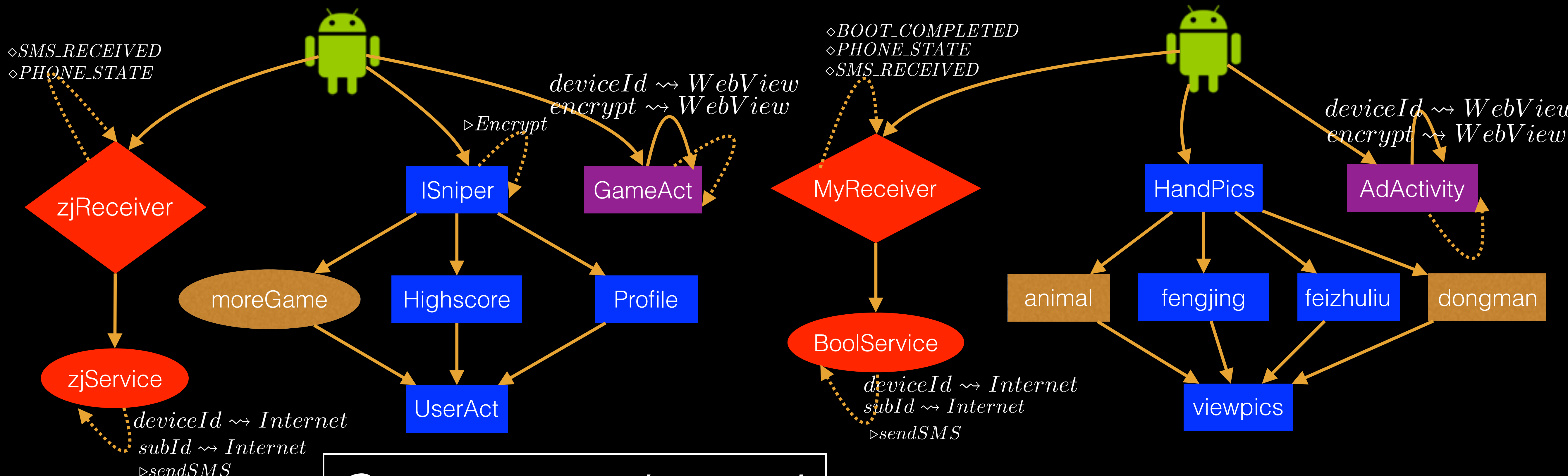

◇SMS_RECEIVED
◇PHONE_STATE
zjReceiver
zjService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
▷Encrypt
ISniper
deviceId ⇝ WebView
encrypt ⇝ WebView
GameAct
moreGame
Highscore
Profile
UserAct

Example

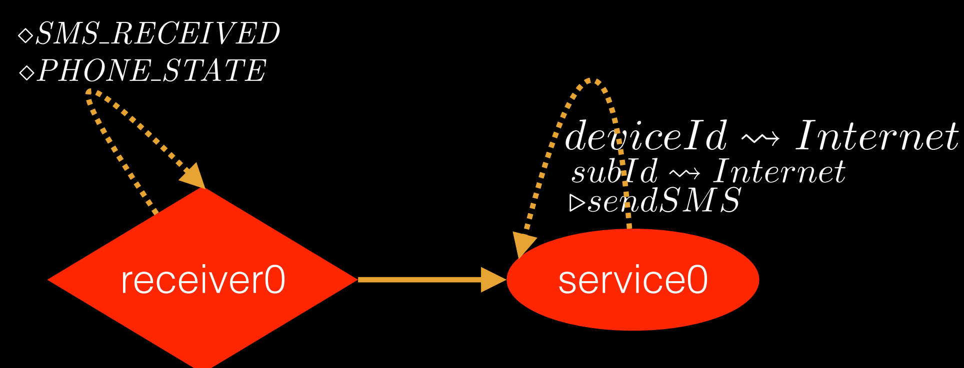


◇SMS_RECEIVED
◇PHONE_STATE
zjReceiver
zjService
deviceId ⇝ Internet
subId ⇝ Internet
▹sendSMS
▹Encrypt
ISniper
moreGame
Highscore
Profile
UserAct
deviceId ⇝ WebView
encrypt ⇝ WebView
GameAct
◇BOOT_COMPLETED
◇PHONE_STATE
◇SMS_RECEIVED
MyReceiver
BoolService
deviceId ⇝ Internet
subId ⇝ Internet
▹sendSMS
HandPics
deviceId ⇝ WebView
encrypt ⇝ WebView
AdActivity
animal
fengjing
feizhuliu
dongman
viewpics

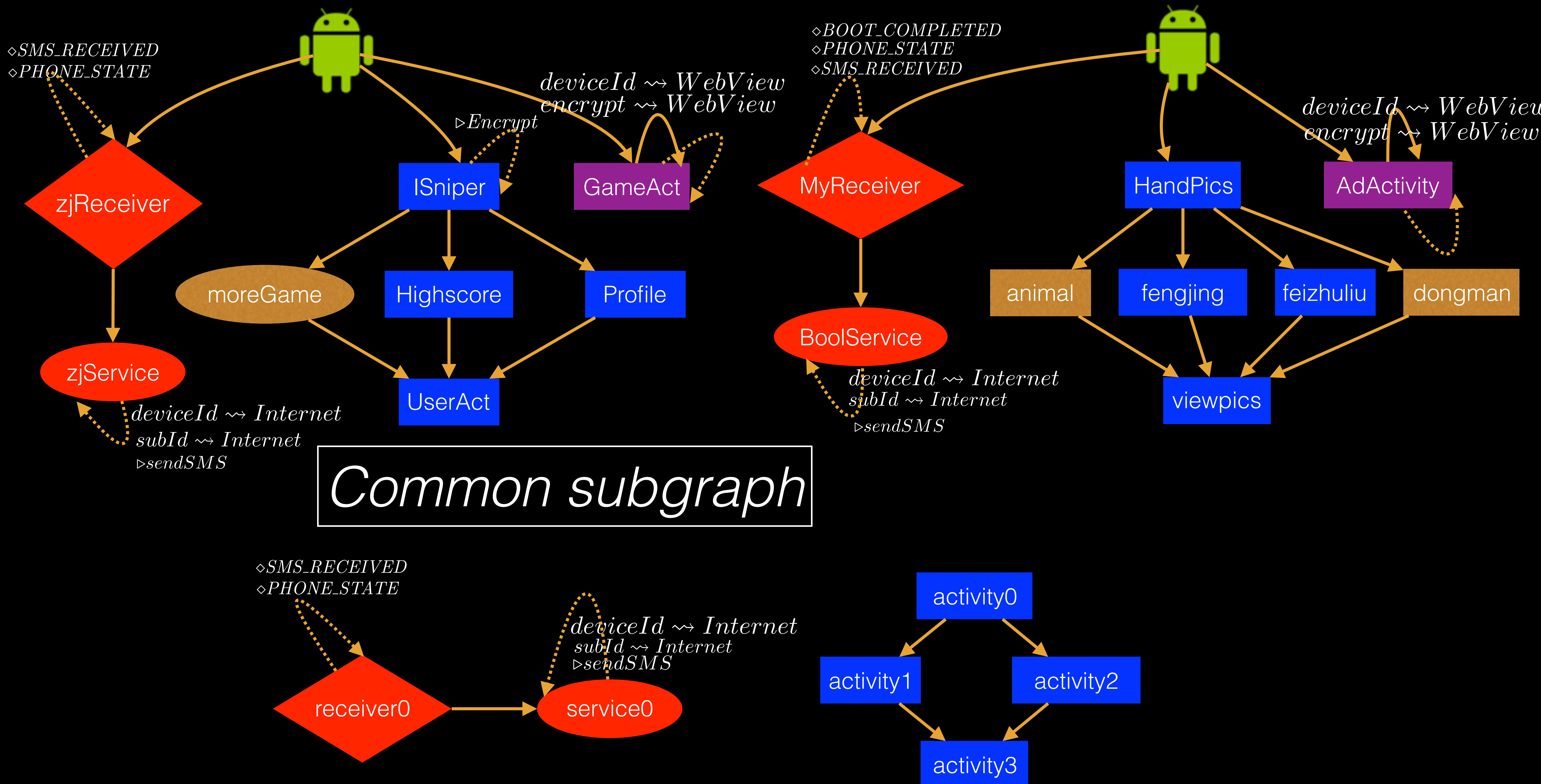
Example



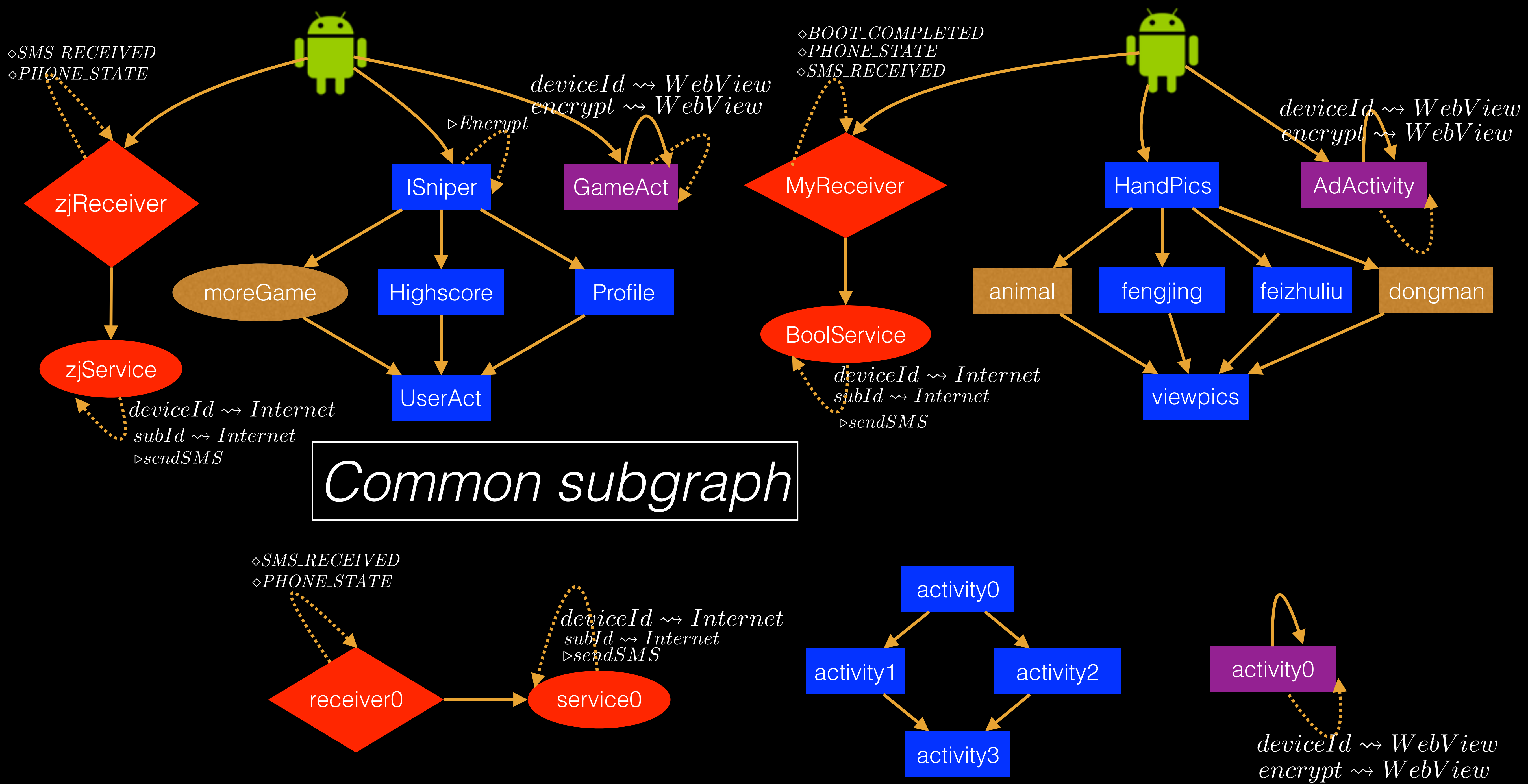
Common subgraph



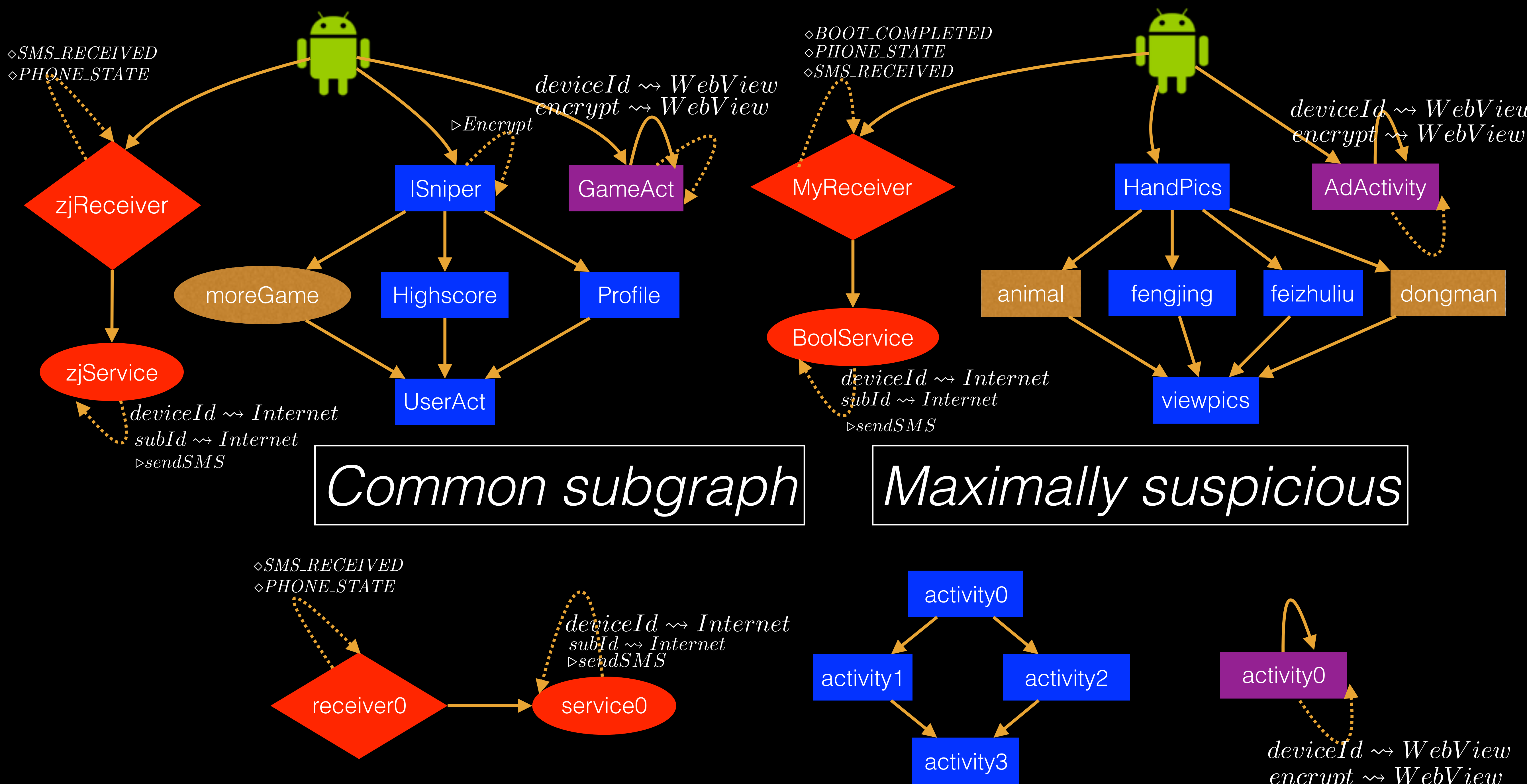
Example


◇SMS_RECEIVED
◇PHONE_STATE
zjReceiver
zjService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
▷Encrypt
ISniper
GameAct
deviceId ⇝ WebView
encrypt ⇝ WebView
moreGame
Highscore
Profile
UserAct
◇BOOT_COMPLETED
◇PHONE_STATE
◇SMS_RECEIVED
MyReceiver
BoolService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
HandPics
AdActivity
deviceId ⇝ WebView
encrypt ⇝ WebView
animal
fengjing
feizhuliu
dongman
viewpics
Common subgraph
◇SMS_RECEIVED
◇PHONE_STATE
receiver0
service0
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
activity0
activity1
activity2
activity3

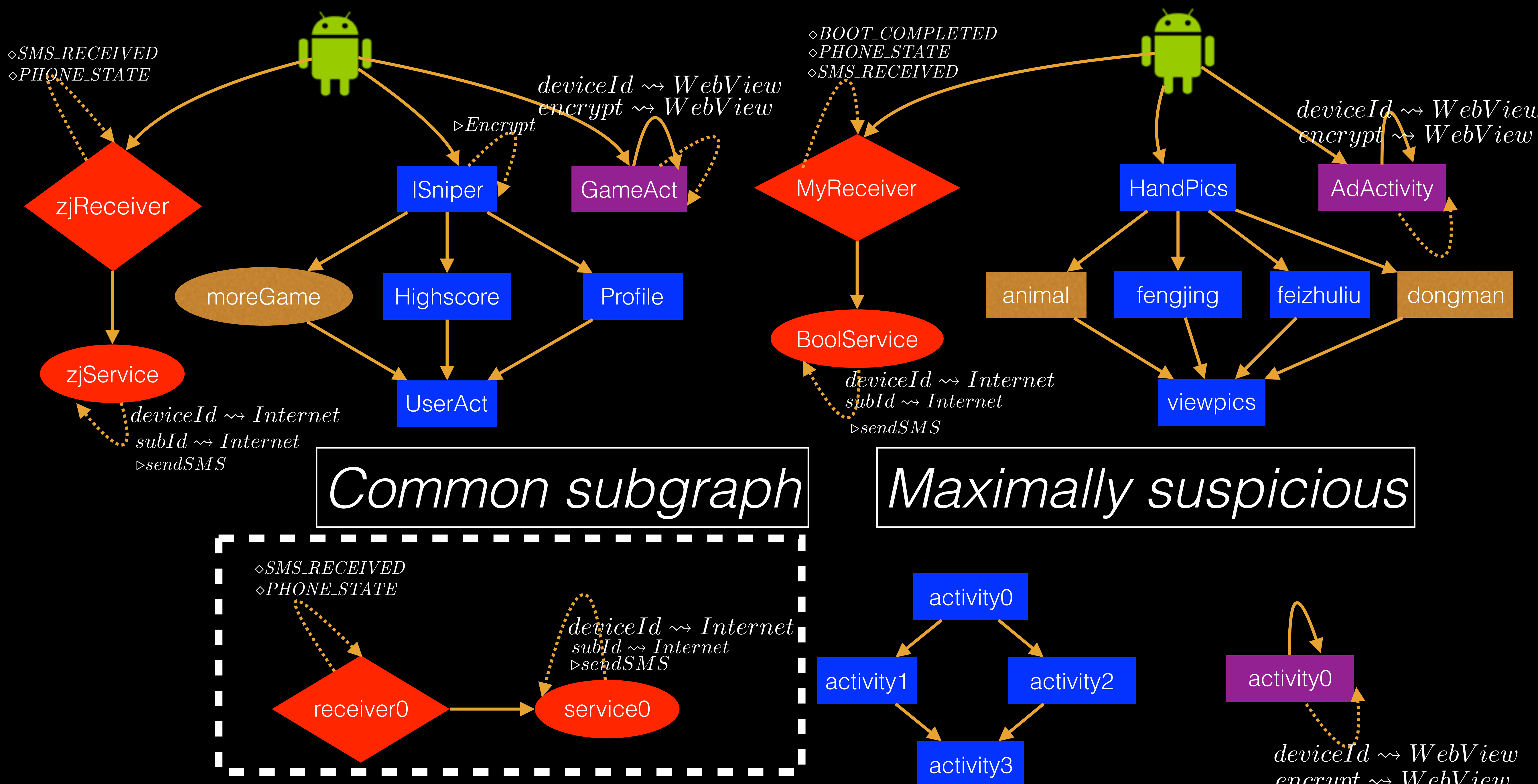
Example


◇SMS_RECEIVED
◇PHONE_STATE
zjReceiver
zjService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
▷Encrypt
ISniper
deviceId ⇝ WebView
encrypt ⇝ WebView
GameAct
moreGame
Highscore
Profile
UserAct
◇BOOT_COMPLETED
◇PHONE_STATE
◇SMS_RECEIVED
MyReceiver
BoolService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
deviceId ⇝ WebView
encrypt ⇝ WebView
HandPics
AdActivity
animal
fengjing
feizhuliu
dongman
viewpics
Common subgraph
◇SMS_RECEIVED
◇PHONE_STATE
receiver0
service0
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
activity0
activity1
activity2
activity3
activity0
deviceId ⇝ WebView
encrypt ⇝ WebView

Example


◇SMS_RECEIVED
◇PHONE_STATE
zjReceiver
zjService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
▷Encrypt
ISniper
deviceId ⇝ WebView
encrypt ⇝ WebView
GameAct
moreGame
Highscore
Profile
UserAct
◇BOOT_COMPLETED
◇PHONE_STATE
◇SMS_RECEIVED
MyReceiver
BoolService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
HandPics
deviceId ⇝ WebView
encrypt ⇝ WebView
AdActivity
animal
fengjing
feizhuliu
dongman
viewpics
Common subgraph
Maximally suspicious
◇SMS_RECEIVED
◇PHONE_STATE
receiver0
service0
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
activity0
activity1
activity2
activity3
activity0
deviceId ⇝ WebView
encrypt ⇝ WebView

Example



◇SMS_RECEIVED
◇PHONE_STATE
zjReceiver
zjService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
ISniper
▷Encrypt
GameAct
deviceId ⇝ WebView
encrypt ⇝ WebView
moreGame
Highscore
Profile
UserAct
◇BOOT_COMPLETED
◇PHONE_STATE
◇SMS_RECEIVED
MyReceiver
BoolService
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
HandPics
AdActivity
deviceId ⇝ WebView
encrypt ⇝ WebView
animal
fengjing
feizhuliu
dongman
viewpics
Common subgraph
Maximally suspicious
◇SMS_RECEIVED
◇PHONE_STATE
receiver0
service0
deviceId ⇝ Internet
subId ⇝ Internet
▷sendSMS
activity0
activity1
activity2
activity3
activity0
deviceId ⇝ WebView
encrypt ⇝ WebView

How to infer the signature

Infer signatures by finding a Maximally Suspicious Common Subgraph of n malware samples

How to infer the signature

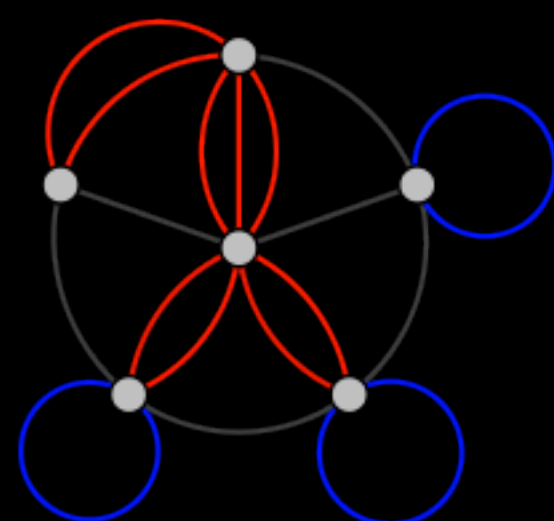
Infer signatures by finding a Maximally Suspicious Common Subgraph of n malware samples

Signature Inference

How to infer the signature

Infer signatures by finding a *<u>Maximally Suspicious Common Subgraph</u> of n malware samples*

Signature Inference

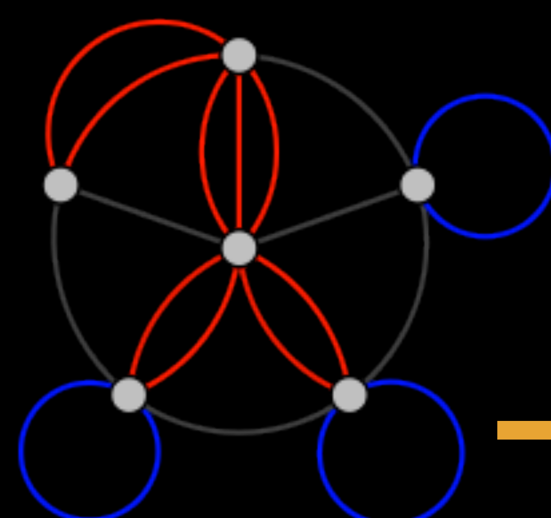


MSCS

How to infer the signature

Infer signatures by finding a *<u>Maximally Suspicious Common Subgraph</u> of n malware samples*

Signature Inference



MSCS

MaxSat

MaxSat in a nutshell

MaxSat in a nutshell

MaxSat: Given a UNSAT boolean formula in CNF, determine the maximum number of satisfied clauses

$$(x_0 \lor x_1) \land (\neg x_0 \lor x_1) \land (x_0 \lor \neg x_1) \land (\neg x_0 \lor \neg x_1)$$

MaxSat in a nutshell

MaxSat: Given a UNSAT boolean formula in CNF, determine the maximum number of satisfied clauses

$$(x_0 \vee x_1) \wedge (\neg x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) \wedge (\neg x_0 \vee \neg x_1)$$

Hard Clause: has to be satisfied

MaxSat in a nutshell

MaxSat: Given a UNSAT boolean formula in CNF, determine the maximum number of satisfied clauses

$$(x_0 \vee x_1) \wedge (\neg x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) \wedge (\neg x_0 \vee \neg x_1)$$

Hard Clause: has to be satisfied

Soft Clause: preferable to be satisfied but could be UNSAT. Each has different weight since some are more important than the others

MaxSat in a nutshell

MaxSat: Given a UNSAT boolean formula in CNF, determine the maximum number of satisfied clauses

$$(x_0 \vee x_1) \wedge (\neg x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) \wedge (\neg x_0 \vee \neg x_1)$$

Hard Clause: has to be satisfied

Soft Clause: preferable to be satisfied but could be UNSAT. Each has different weight since some are more important than the others

Find an assignment s.t. the total weight of satisfied clauses is maximized

$$\{x_0 \mapsto 0, x_1 \mapsto 0\}$$

Synthesis using MaxSat

Synthesis using MaxSat

- Hard Clause: common subgraph (control-flow property)

Synthesis using MaxSat

- Hard Clause: common subgraph (control-flow property)
- Soft Clause: maximally suspiciousness (data-flow property)

Synthesis using MaxSat

- Hard Clause: common subgraph (control-flow property)
- Soft Clause: maximally suspiciousness (data-flow property)
- Weight for each clause
  - Inverse frequency from benign samples
  - Higher weight to features that are commonly found in malware

$$\mathcal{O} = \underbrace{\sum_{v,v' \in V} x_0(v,v')}_{\textit{Hard}} + \underbrace{\sum_{v,v' \in V} \sum_{d \in \mathcal{D}} w_{(v,v',d)} y_0(v,v',d)}_{\textit{Soft}}.$$

Example, cont.

$$\mathcal{O} = \underbrace{\sum_{v,v' \in V} x_0(v,v')}_{\textit{Hard}} + \underbrace{\sum_{v,v' \in V} \sum_{d \in \mathcal{D}} w_{(v,v',d)} y_0(v,v',d)}_{\textit{Soft}}.$$

Example, cont.

Control properties

$$\mathcal{O} = \underbrace{\sum_{v,v' \in V} x_0(v,v')}_{\textit{Hard}} + \underbrace{\sum_{v,v' \in V} \sum_{d \in \mathcal{D}} w_{(v,v',d)} y_0(v,v',d).}_{\textit{Soft}}$$

Example, cont.

Control properties

Data properties

$$\mathcal{O} = \sum_{v,v' \in V} x_0(v,v') + \sum_{v,v' \in V} \sum_{d \in \mathcal{D}} w_{(v,v',d)} y_0(v,v',d).$$

Hard

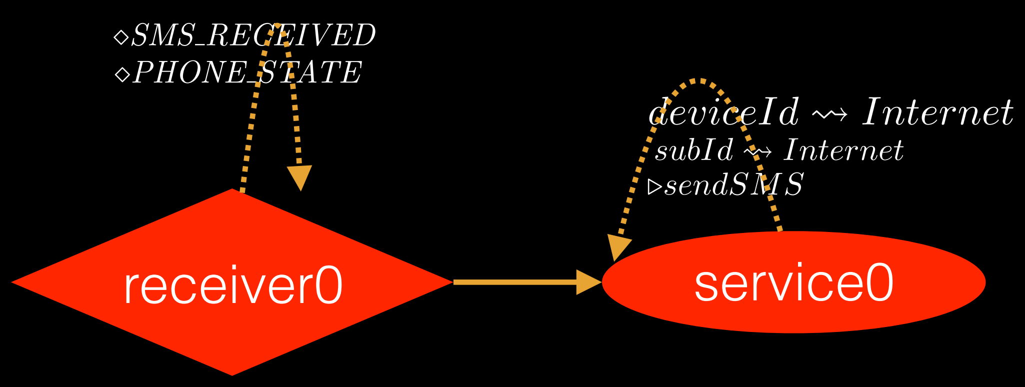
Soft

Example, cont.

Control properties **Data properties**

$$\mathcal{O} = \sum_{v,v' \in V} x_0(v,v') + \sum_{v,v' \in V} \sum_{d \in \mathcal{D}} w_{(v,v',d)} y_0(v,v',d).$$

Hard *Soft*



$$\mathcal{O}_1 = 6$$

Example, cont.

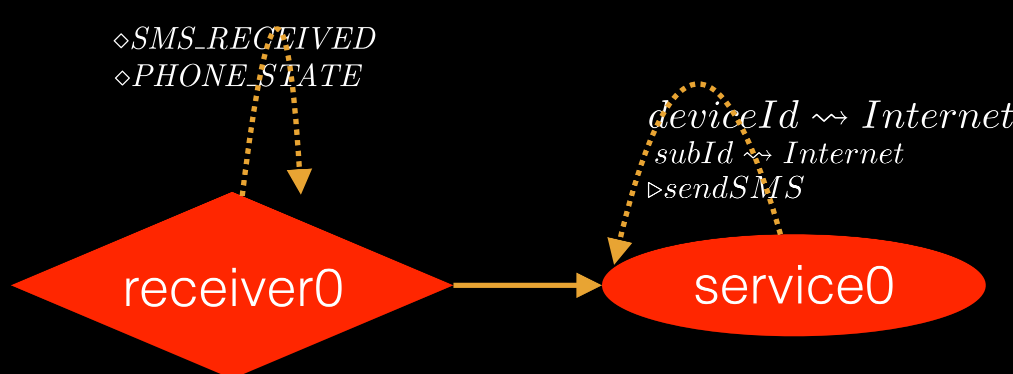
Control properties

Data properties

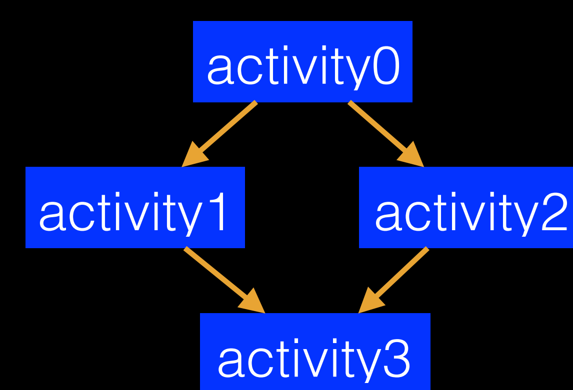
$$\mathcal{O} = \sum_{v,v' \in V} x_0(v, v') + \sum_{v,v' \in V} \sum_{d \in \mathcal{D}} w_{(v,v',d)} y_0(v, v', d).$$

Hard

Soft



$\mathcal{O}_1 = 6$



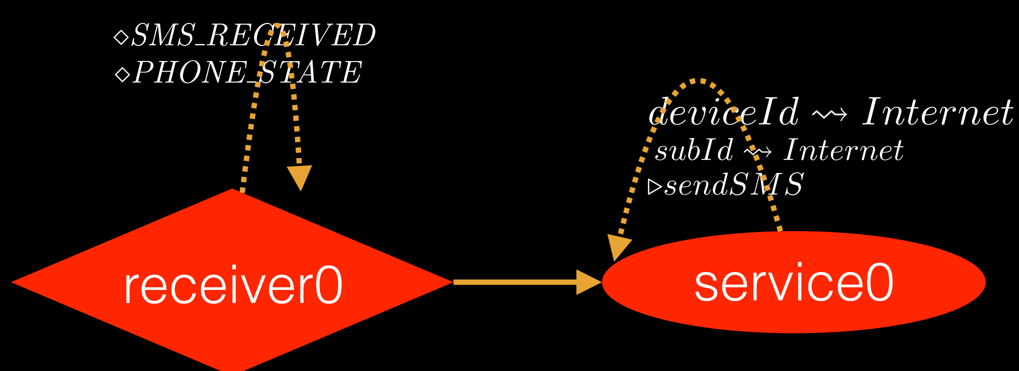
$\mathcal{O}_2 = 4$

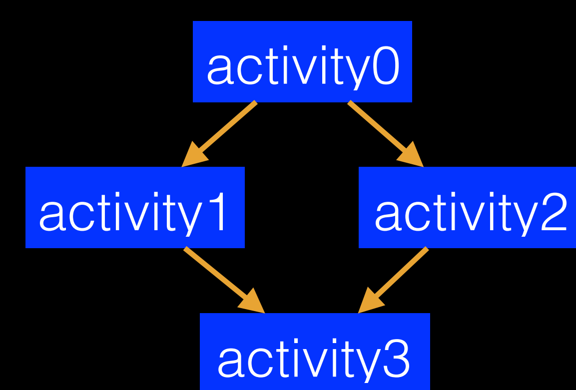
Example, cont.

Control properties

Data properties

$$\mathcal{O} = \underset{\text{Hard}}{\sum_{v,v'\in V} x_0(v,v')} + \underset{\text{Soft}}{\sum_{v,v'\in V}\sum_{d\in\mathcal{D}} w_{(v,v',d)}\, y_0(v,v',d)}.$$





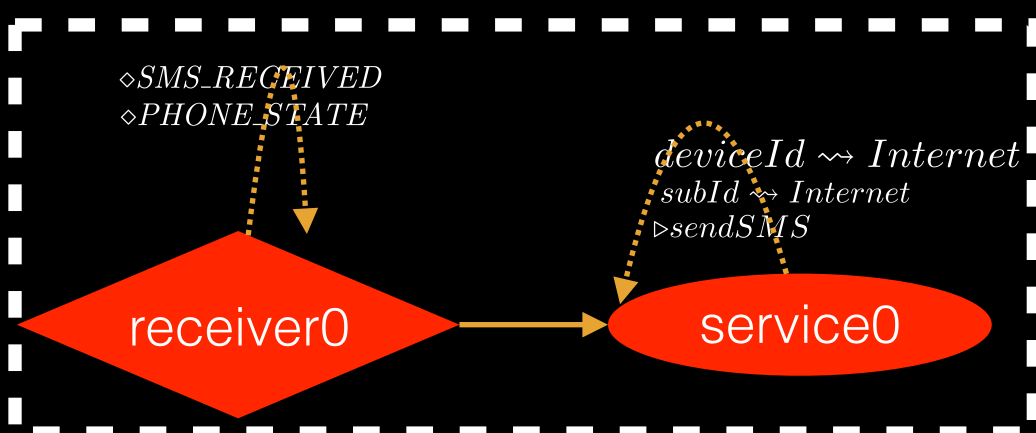
$\mathcal{O}_1 = 6$ $\mathcal{O}_2 = 4$ $\mathcal{O}_3 = 3$

Example, cont.

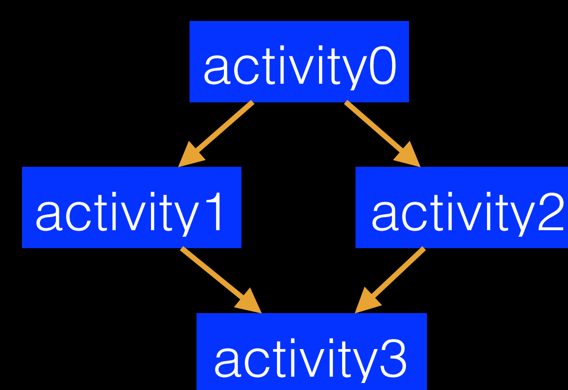
Control properties **Data properties**

$$\mathcal{O} = \sum_{v,v' \in V} x_0(v,v') + \sum_{v,v' \in V} \sum_{d \in \mathcal{D}} w_{(v,v',d)} y_0(v,v',d).$$

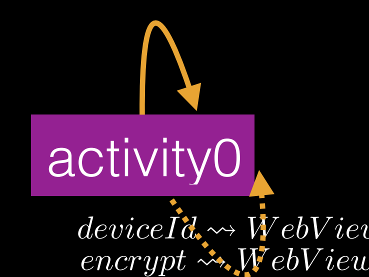
Hard *Soft*



$\mathcal{O}_1 = 6$



$\mathcal{O}_2 = 4$



$\mathcal{O}_3 = 3$

Approximate matching

Now that we have the signature…

Approximate matching

Now that we have the signature…

Utilize existing signature inference algorithm to decide if a sample A belongs to a family F:

Approximate matching

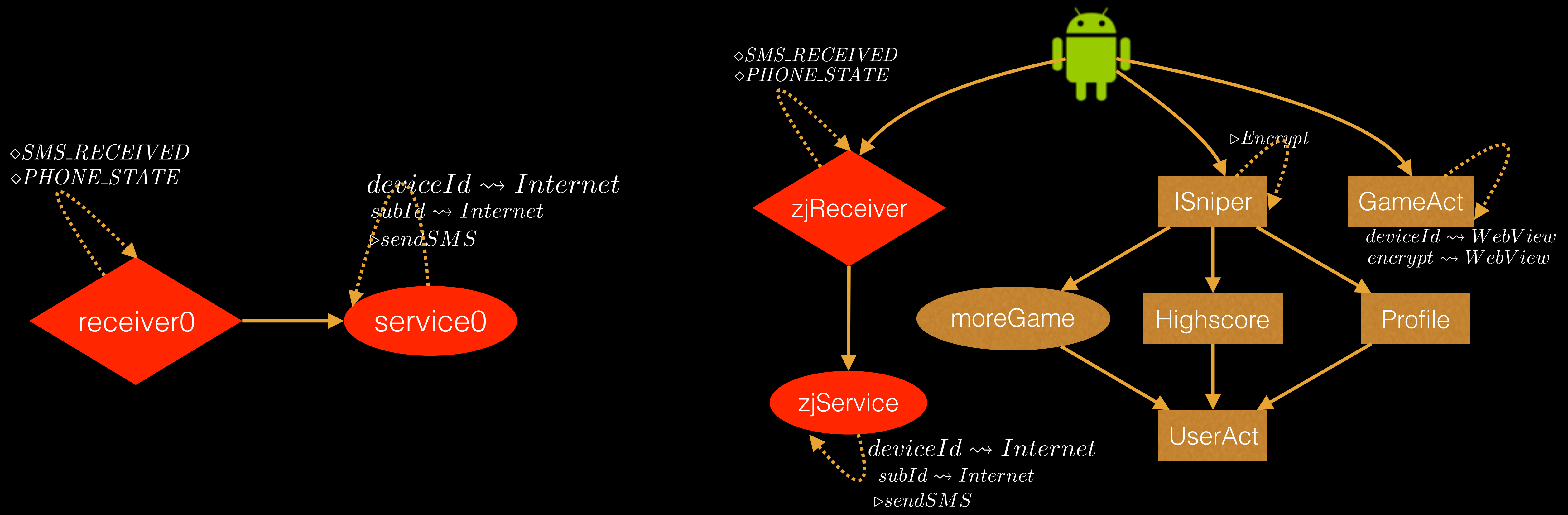
Now that we have the signature…

Utilize existing signature inference algorithm to decide if a sample A belongs to a family F:

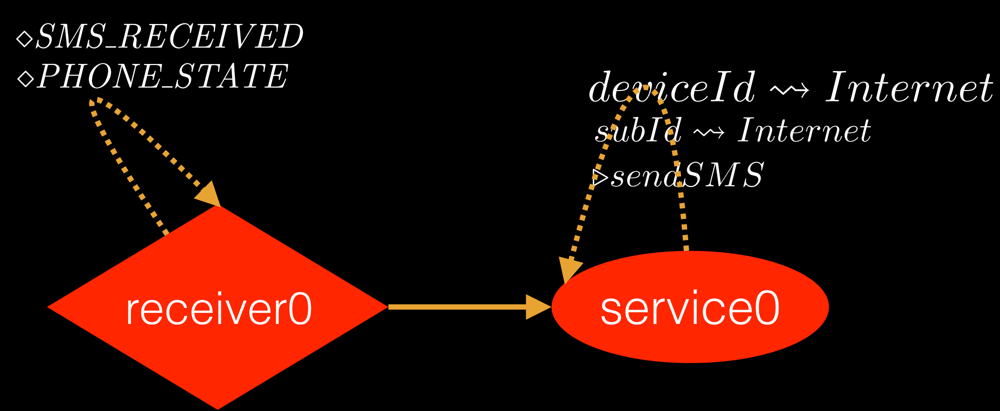
$$\delta(\mathcal{A}, \mathcal{F}) = \frac{f(\textsc{InferSignature}(\mathcal{A}, \mathcal{S}_{\mathcal{F}}))}{f(\mathcal{S}_{\mathcal{F}})}$$

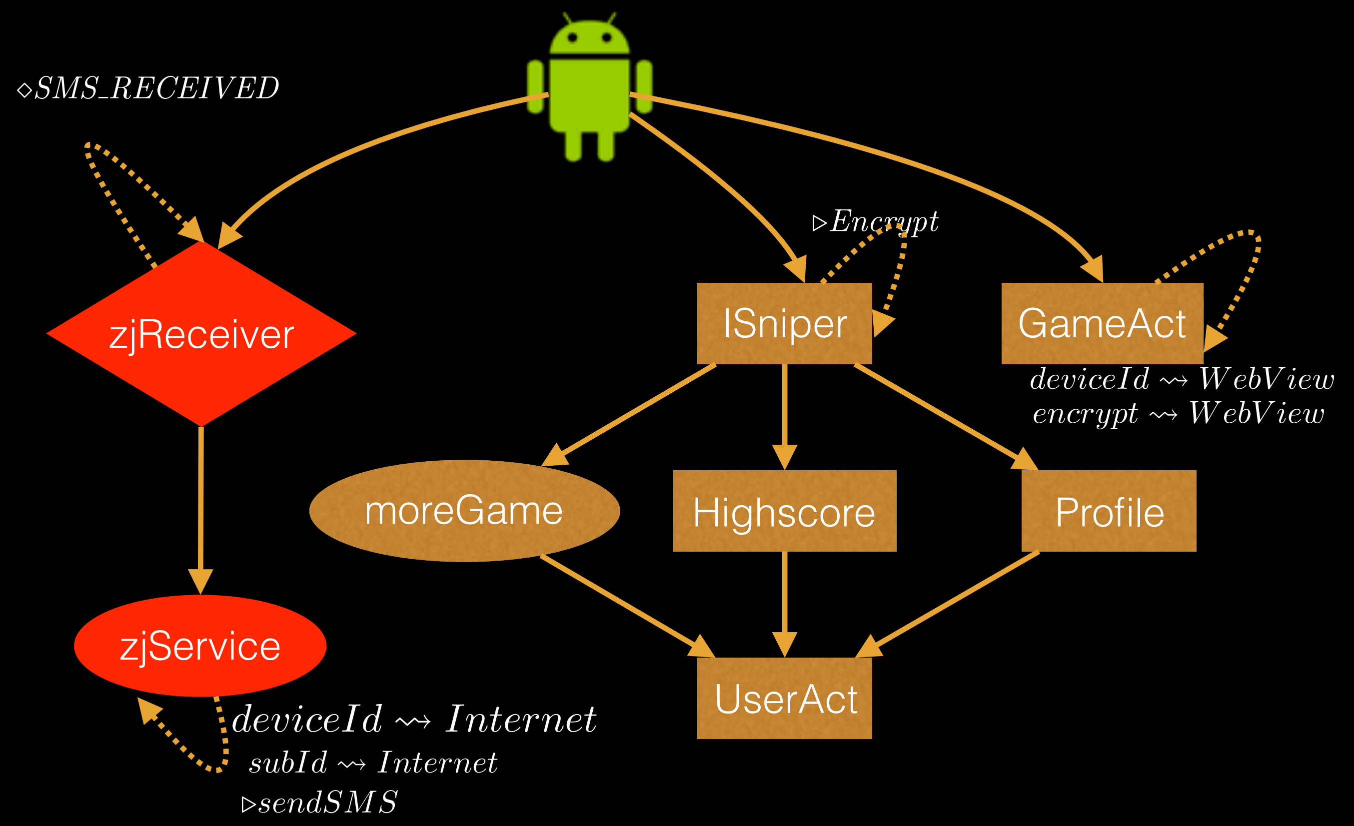
f(S): Weighted sum of the number of nodes and edges in S

Example, cont.

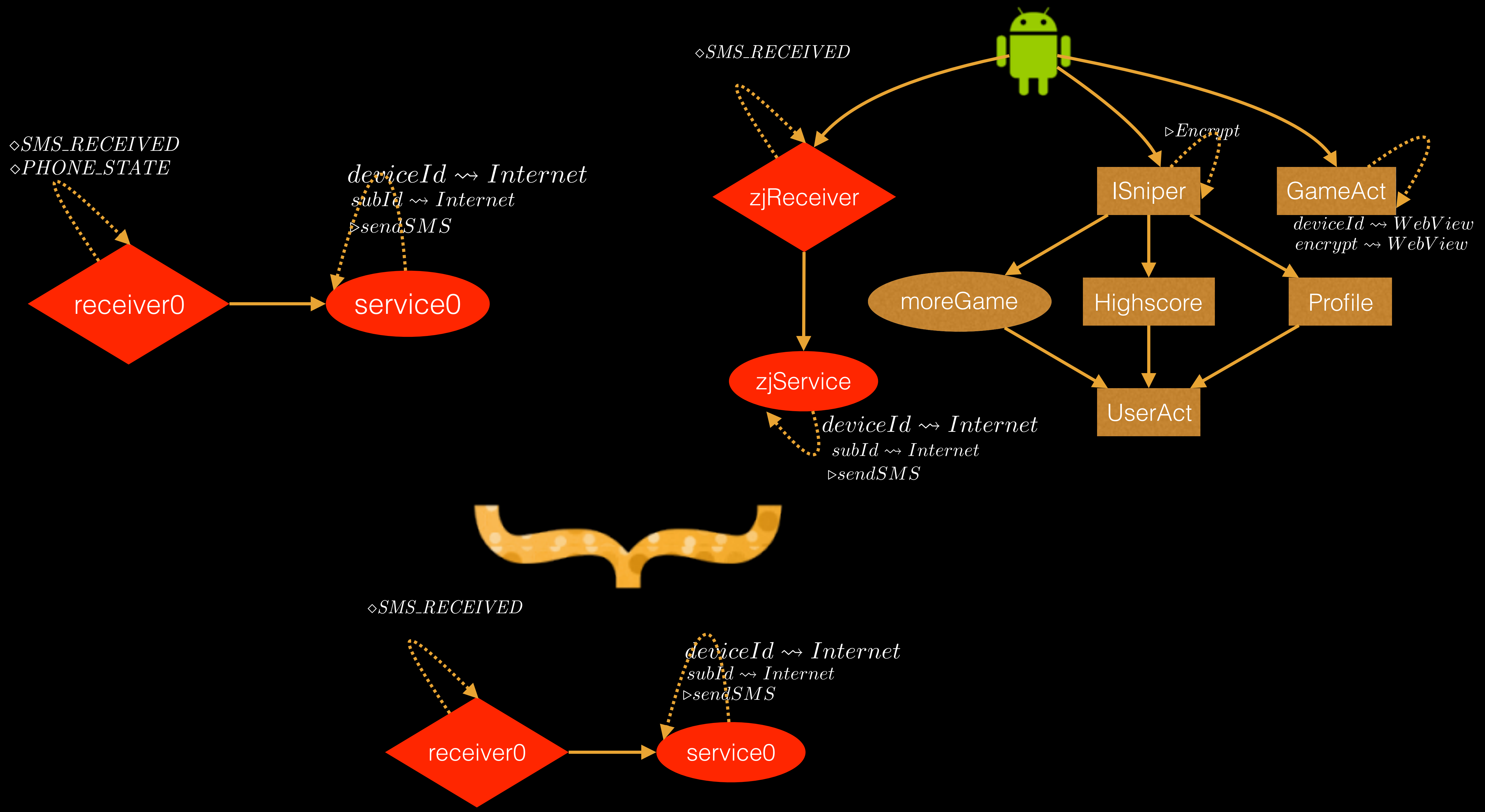


Example, cont.



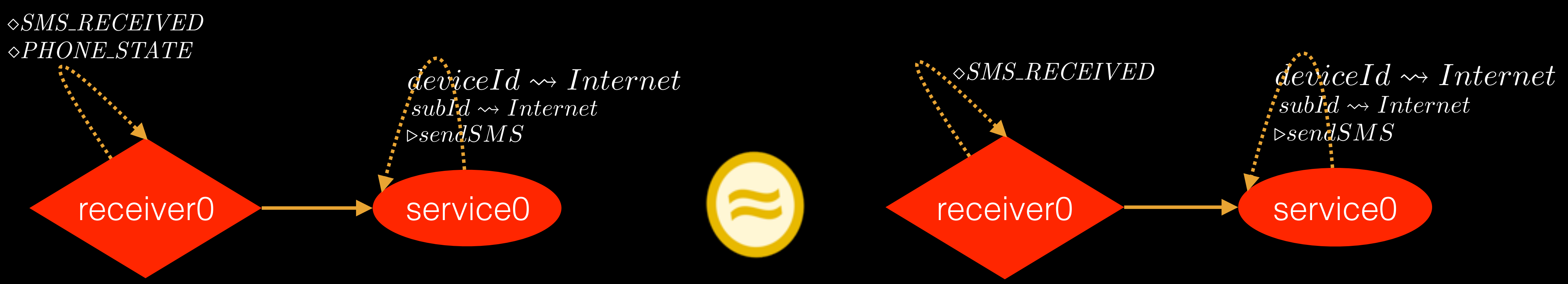


Example, cont.

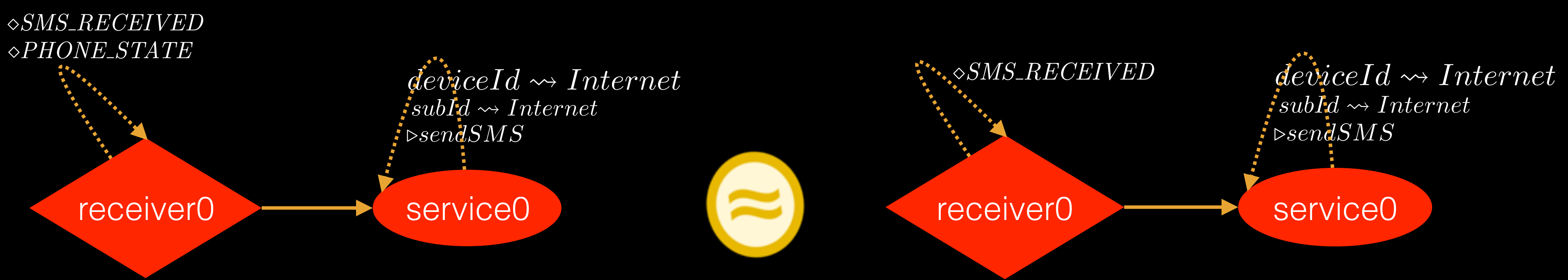


Example, cont.

Example, cont.



Example, cont.



Resistant to semantic obfuscation!

Evaluation

Evaluation

- RQ1: How do the signatures synthesized by Astroid compare with manual version?

Evaluation

- RQ1: How do the signatures synthesized by Astroid compare with manual version?

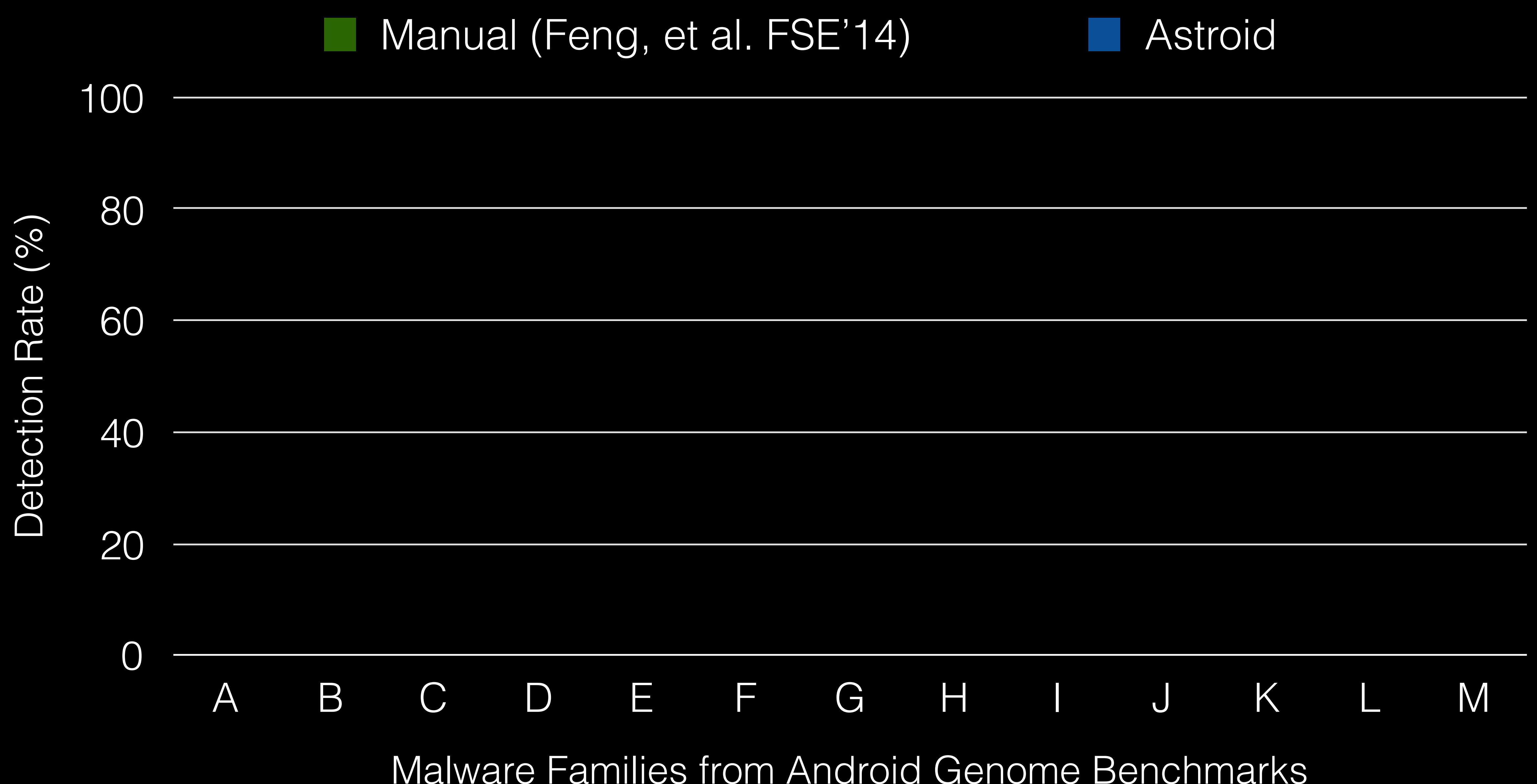
- RQ2: How effective is Astroid at detecting zero-day malware?

Evaluation

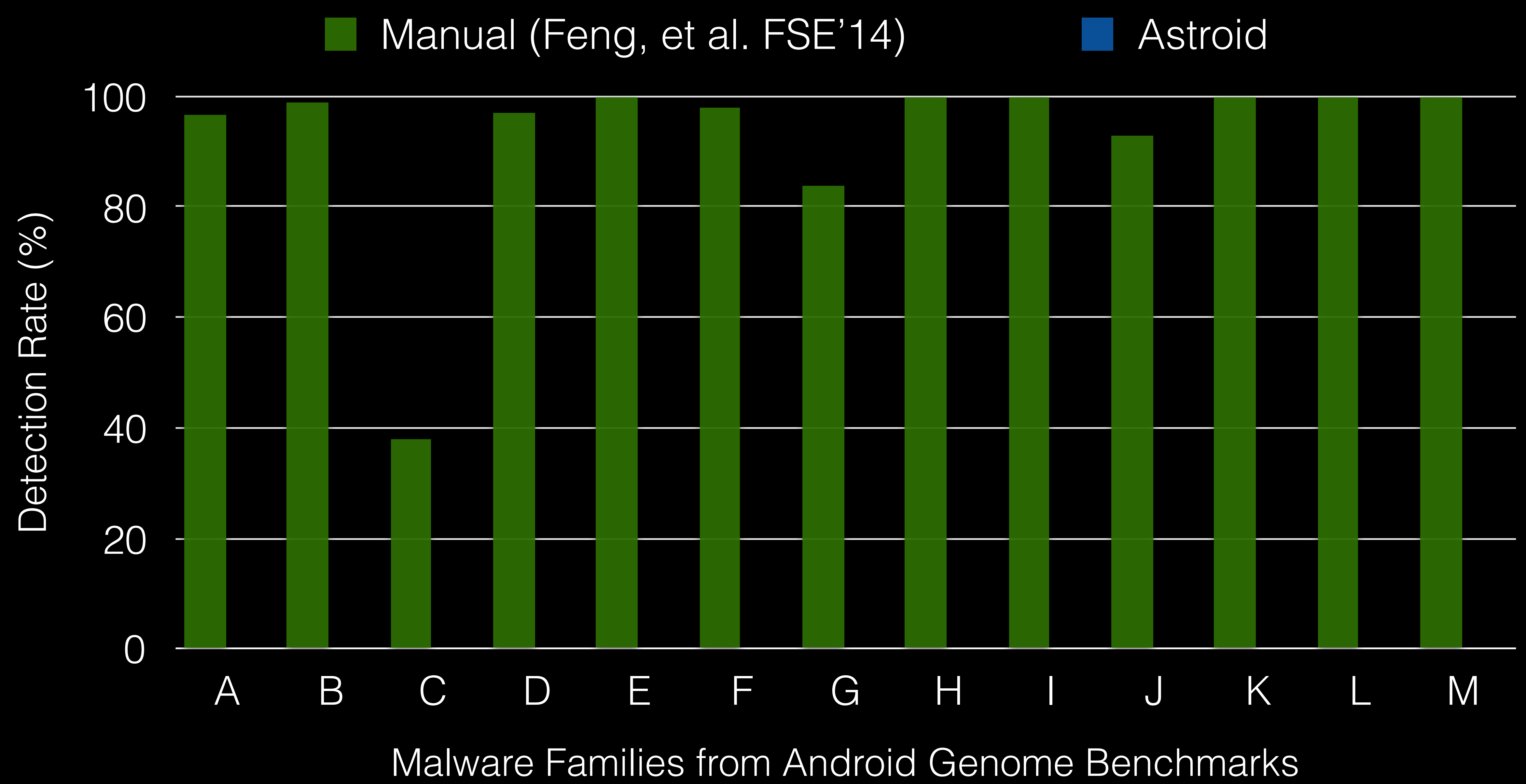
- RQ1: How do the signatures synthesized by Astroid compare with manual version?
- RQ2: How effective is Astroid at detecting zero-day malware?
- RQ3: How does Astroid compare against state-of-the-art malware detectors?

Manual v.s. Automated

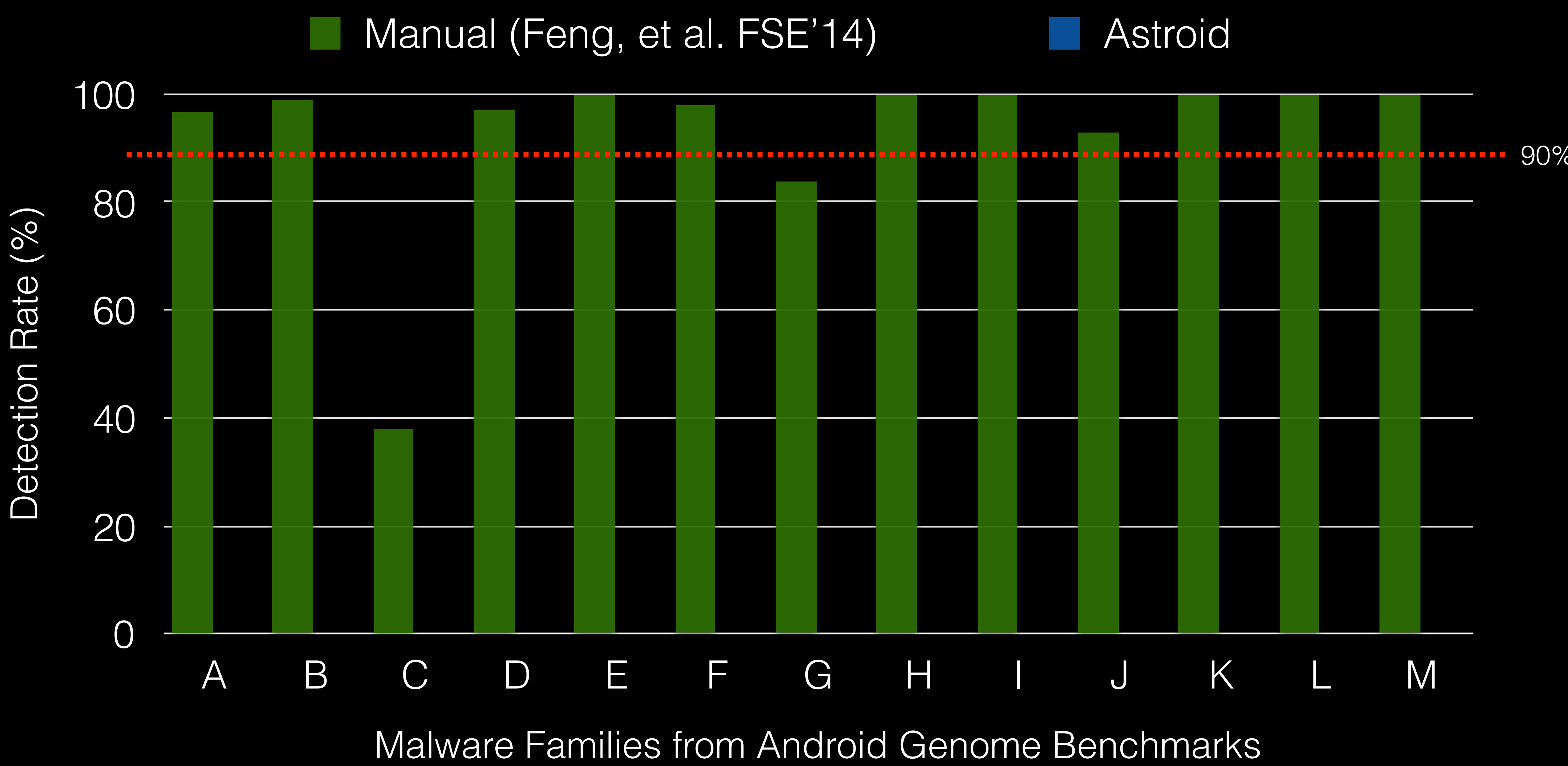
Manual v.s. Automated



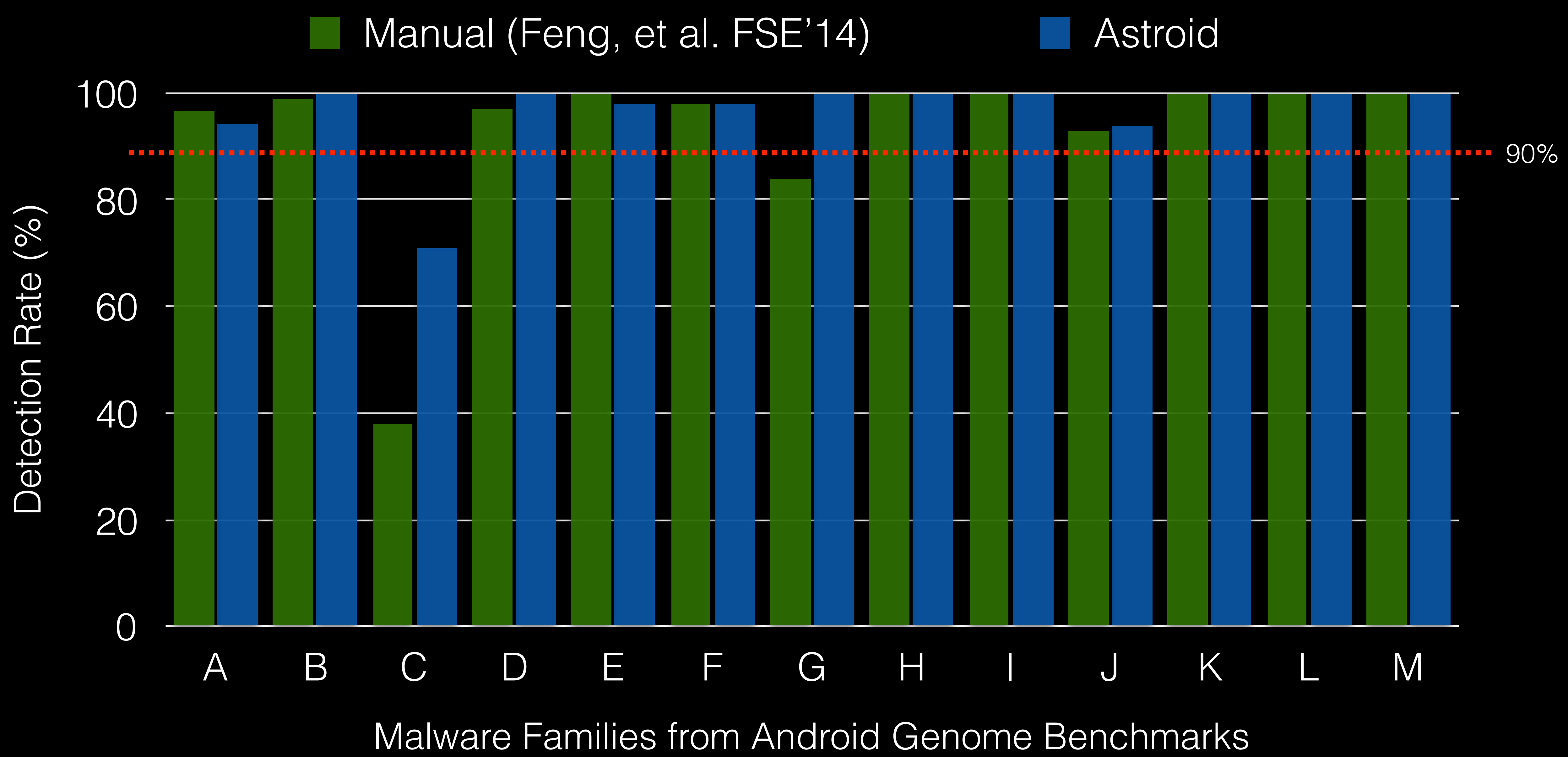
Manual v.s. Automated



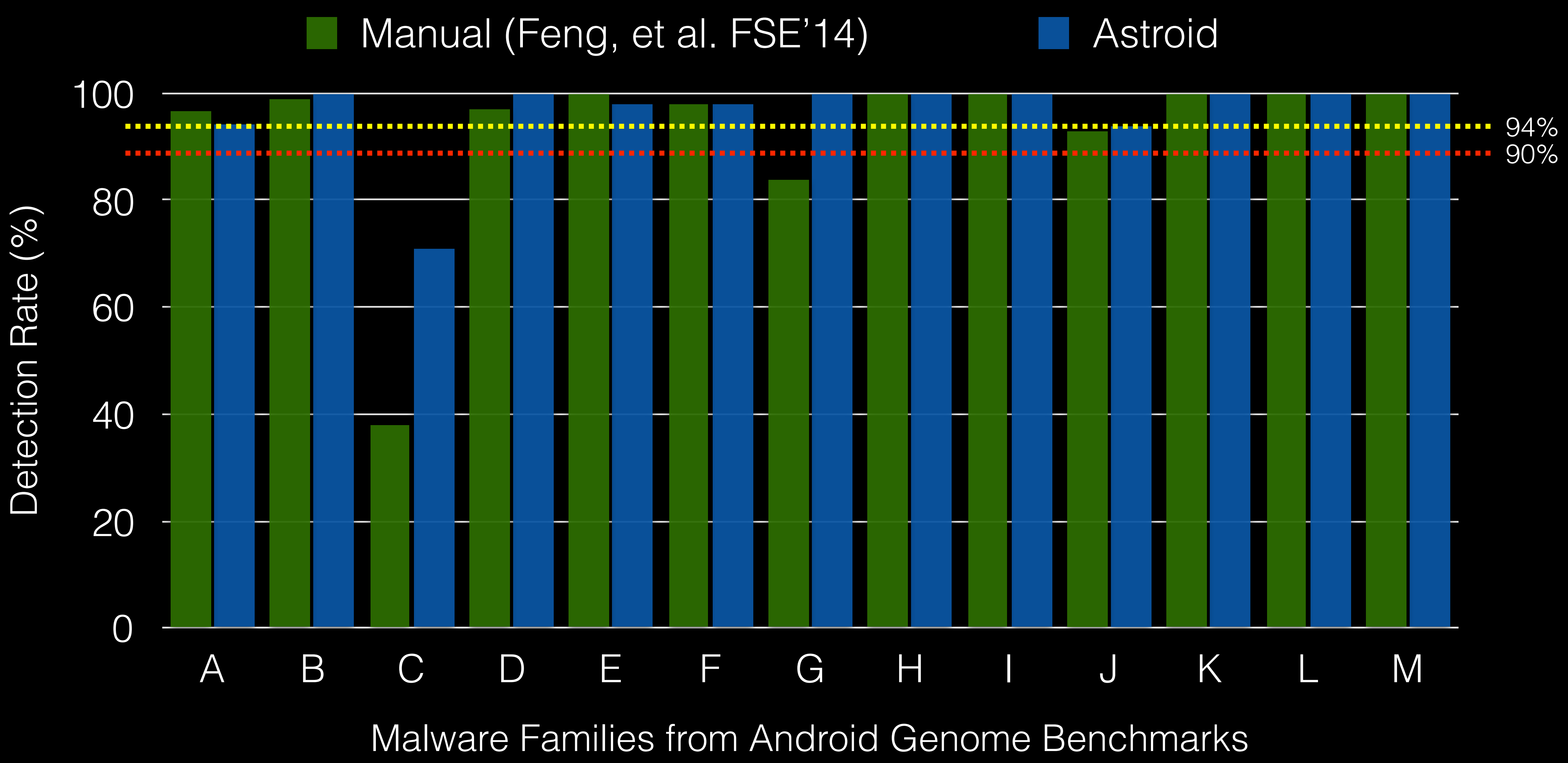
Manual v.s. Automated



Manual v.s. Automated

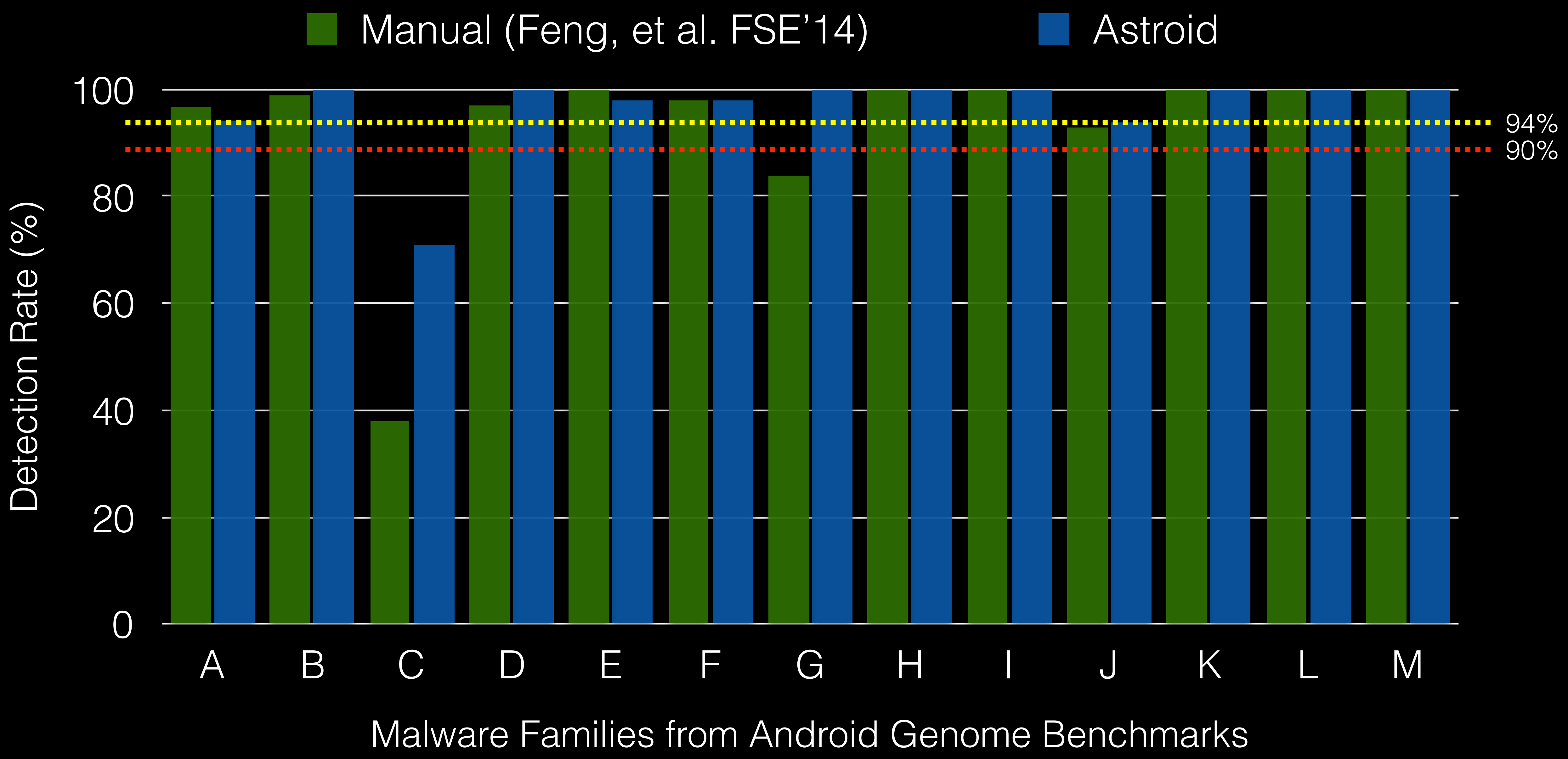


Manual v.s. Automated



Manual v.s. Automated

Outperform manual version!



Zero-day malware

Zero-day malware

- 160 malware samples from Symantec and McAfee of which we have no signature
  - Astroid: 92%, MassVet (Security'15): 81%

Zero-day malware

- 160 malware samples from Symantec and McAfee of which we have no signature
  - Astroid: 92%, MassVet (Security'15): 81%
- Identify 22 Google Play apps that can't be detected by AV tools but are actually malicious after manual inspection

Zero-day malware

- 160 malware samples from Symantec and McAfee of which we have no signature
  - Astroid: 92%, MassVet (Security'15): 81%
- Identify 22 Google Play apps that can't be detected by AV tools but are actually malicious after manual inspection

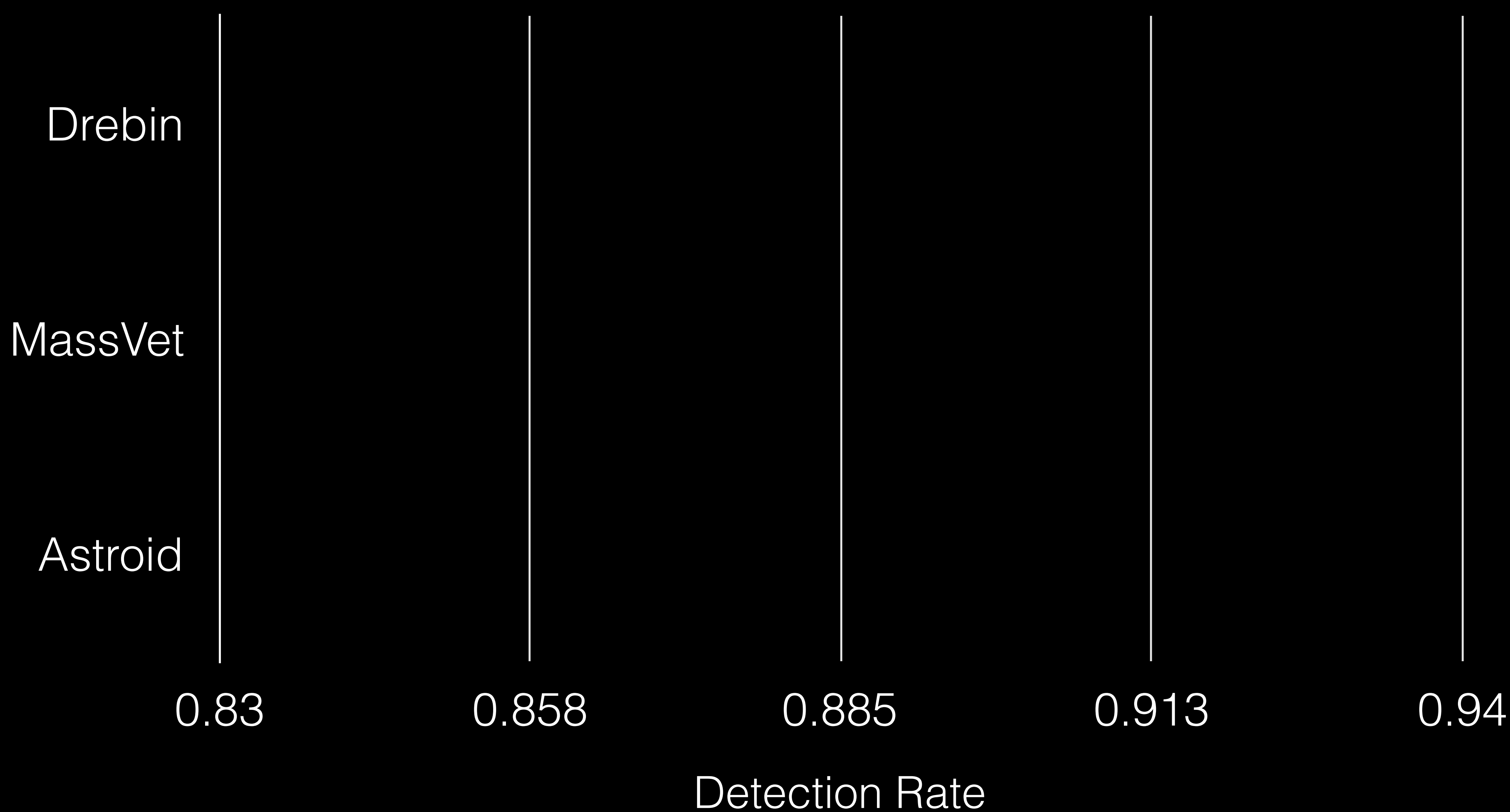
Our approximate matching is effective!

Comparison with other tools

False positive rate: Drebin(NDSS'14): 1%, MassVet (Security'15): 175/503, Astroid: 0.04%

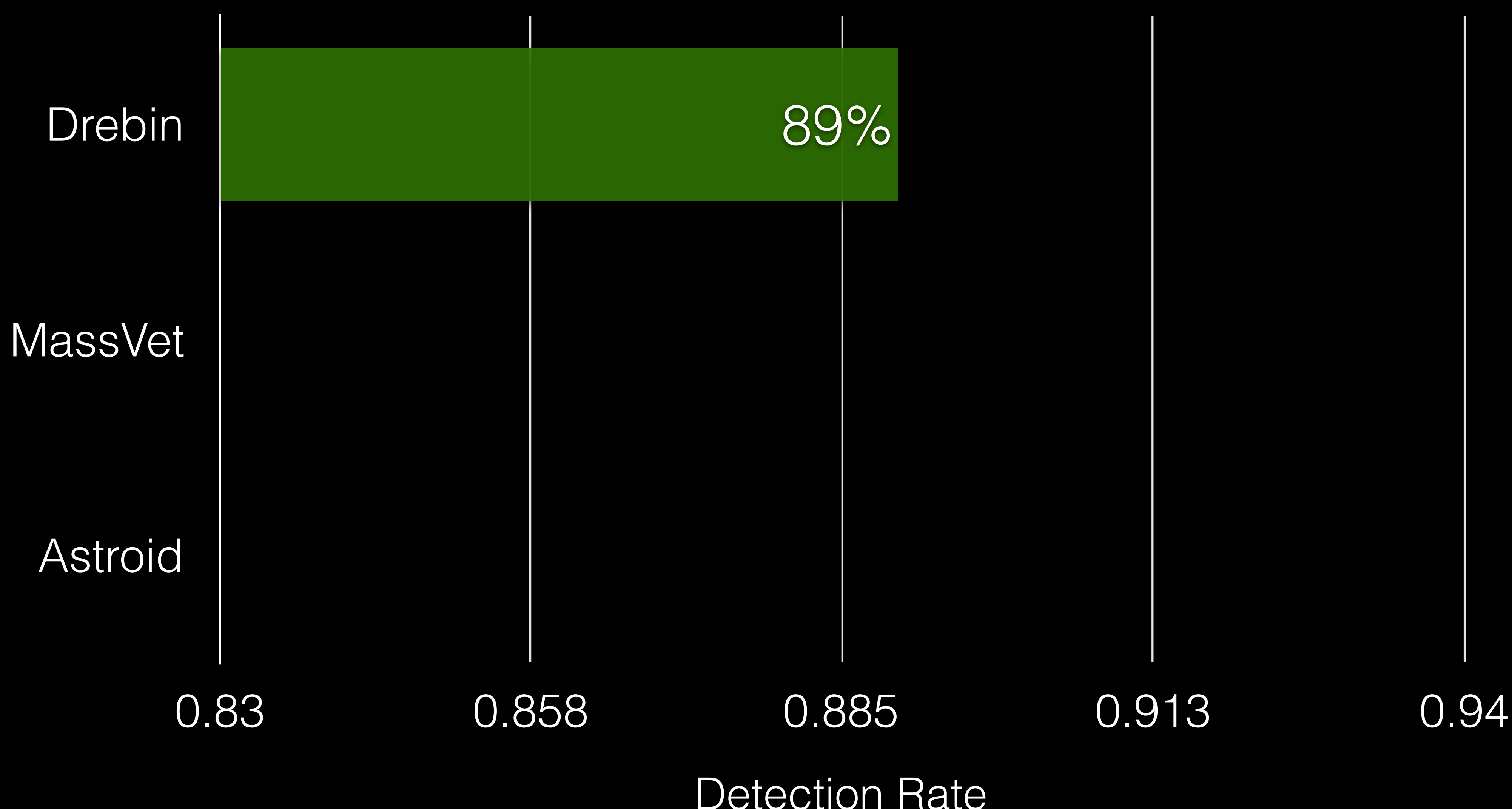
Comparison with other tools

False positive rate: Drebin(NDSS'14): 1%, MassVet (Security'15): 175/503, Astroid: 0.04%



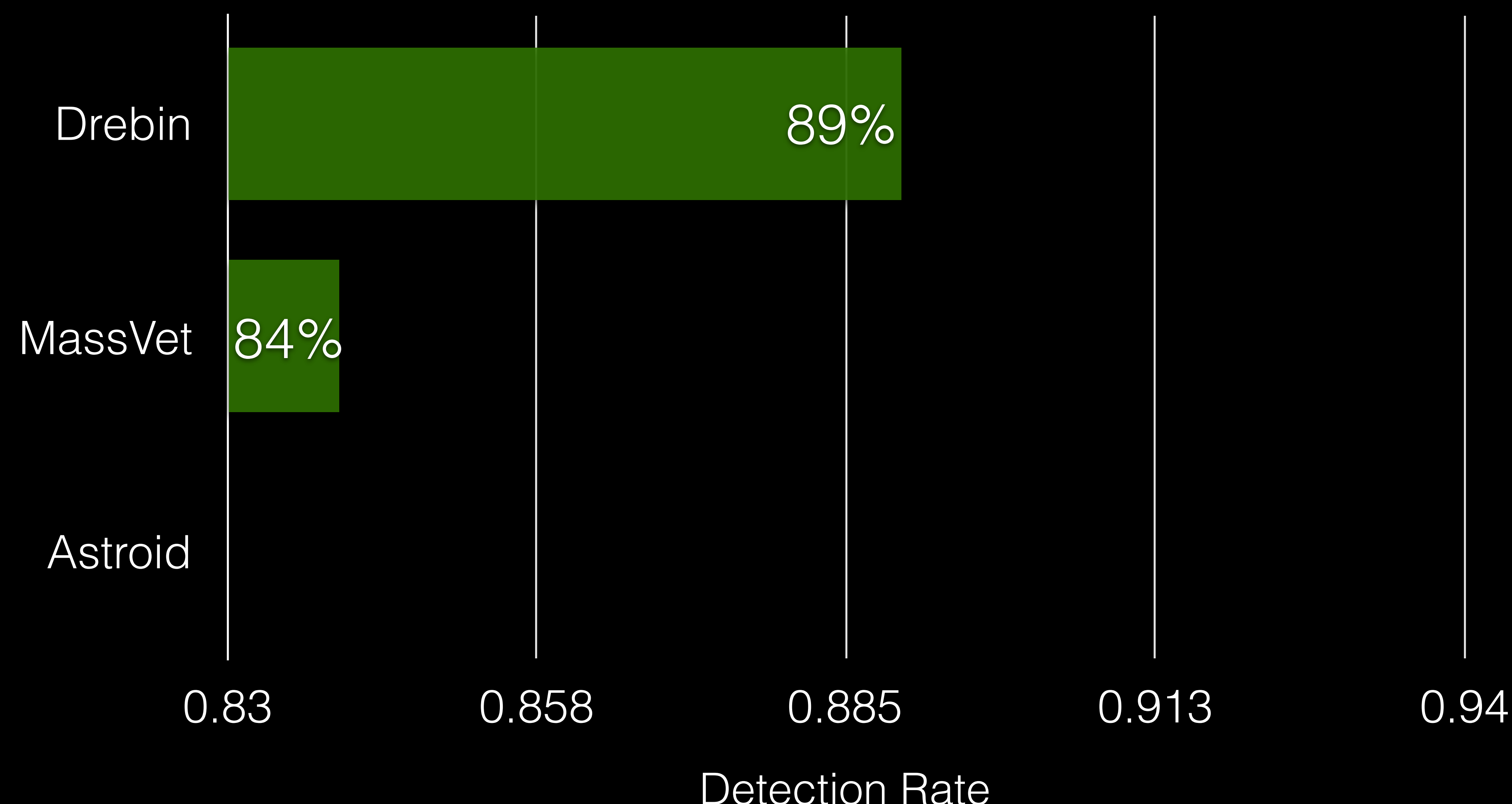
Comparison with other tools

False positive rate: Drebin(NDSS'14): 1%, MassVet (Security'15): 175/503, Astroid: 0.04%



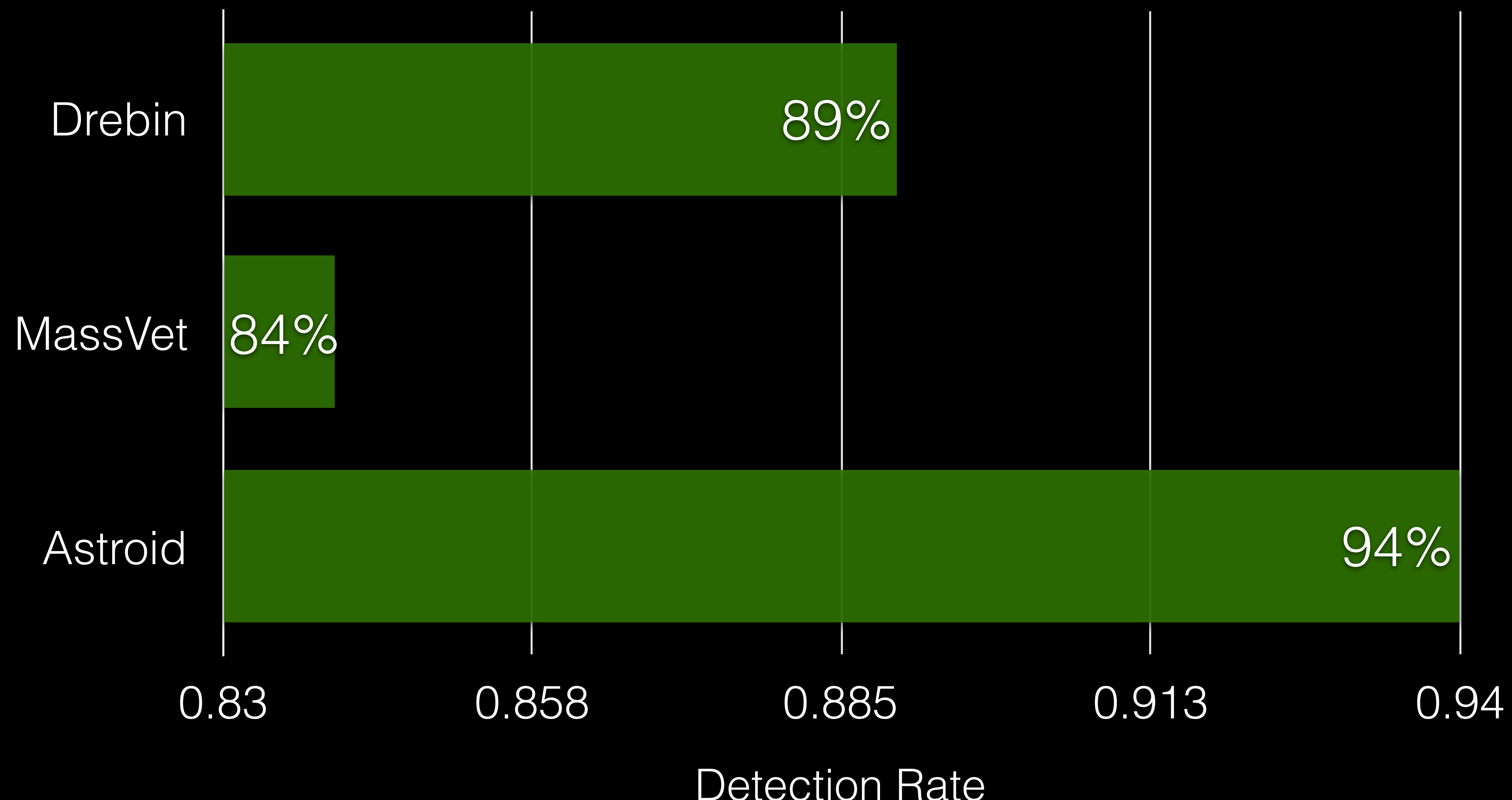
Comparison with other tools

False positive rate: Drebin(NDSS’14): 1%, MassVet (Security’15): 175/503, Astroid: 0.04%



Comparison with other tools

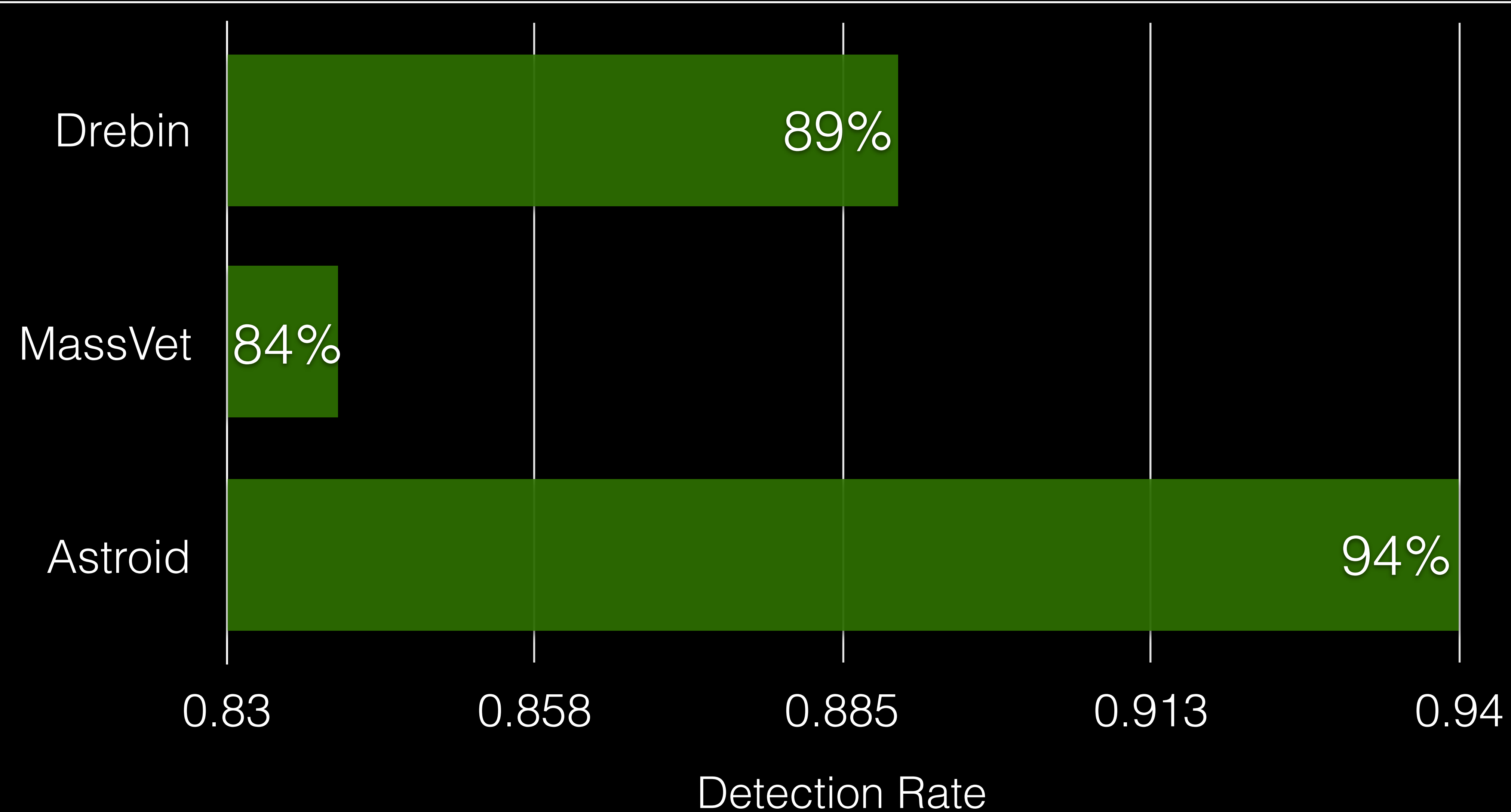
False positive rate: Drebin(NDSS'14): 1%, MassVet (Security'15): 175/503, Astroid: 0.04%



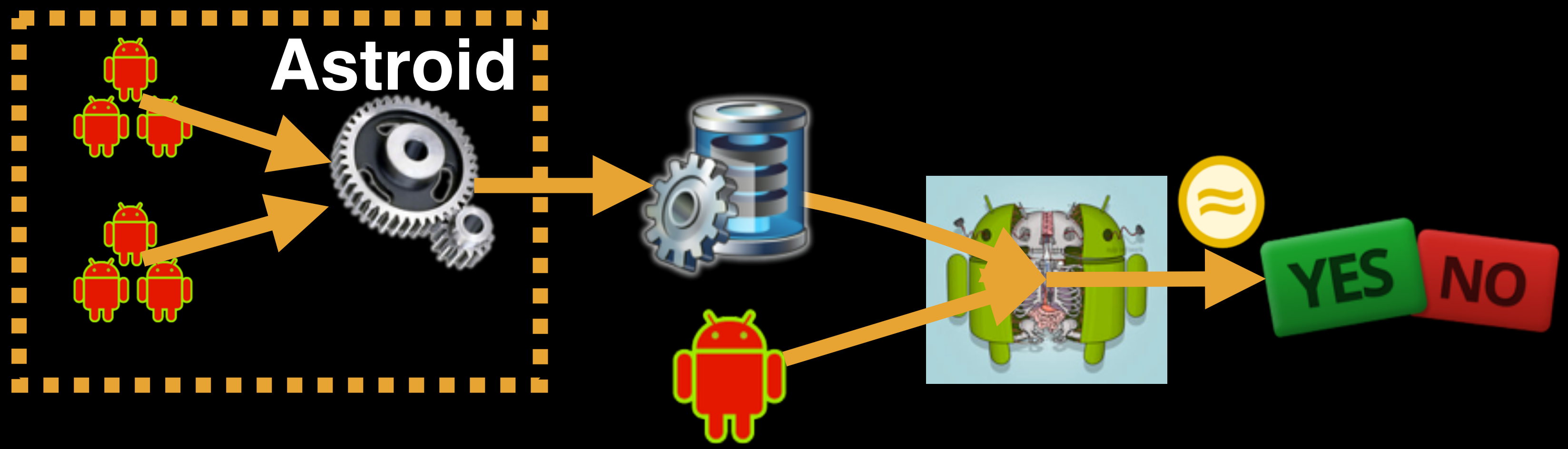
Comparison with other tools

False positive rate: Drebin(NDSS'14): 1%, MassVet (Security'15): 175/503, Astroid: 0.04%

Astroid achieves high detection rate with low FP!

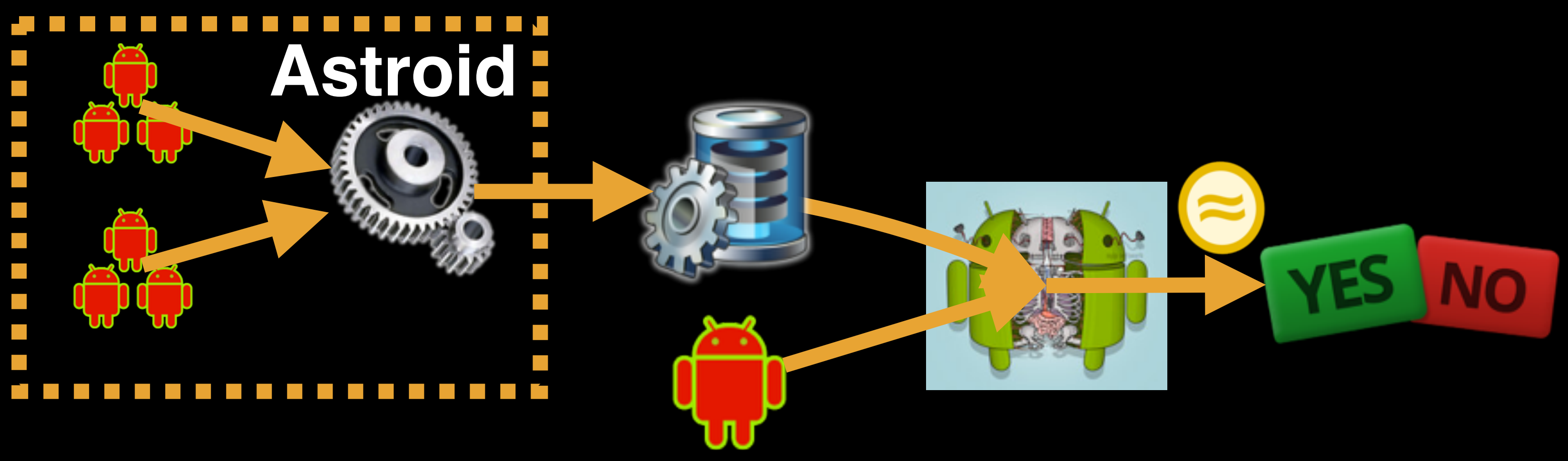


Conclusion



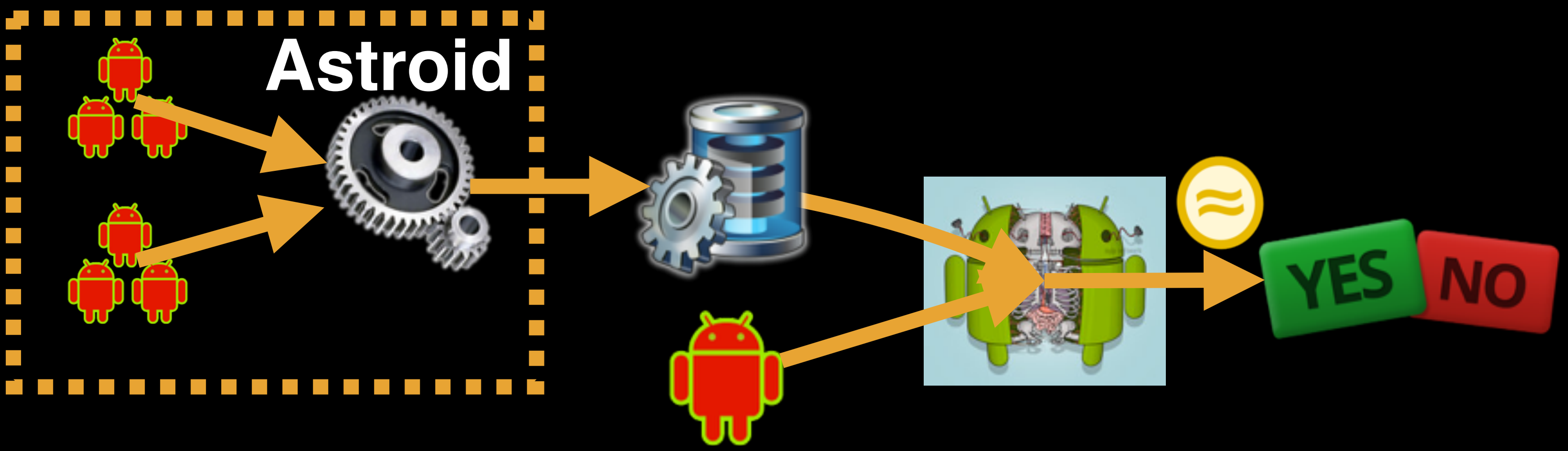
Conclusion

- Automatically infer semantic malware signature from very few samples



Conclusion

- Automatically infer semantic malware signature from very few samples
- Our approximate matching is resilient to semantic obfuscations



Thank you!

Automated Synthesis of Semantic Malware Signatures using Maximum Satisfiability.
Yu Feng, Osbert Bastani, Ruben Martins, Isil Dillig, Saswat Anand. NDSS 2017.

EXPLORER: Query- and Demand-Driven Exploration of Interprocedural Control Flow Properties. *Yu Feng, Xinyu Wang, Isil Dillig, Calvin Lin. OOPSLA 2015.*

Apposcopy: Semantics-Based Detection of Android Malware through Static Analysis.
Yu Feng, Saswat Anand, Isil Dillig, Alex Aiken. FSE 2014.