



ANALYZING INTER-APPLICATION COMMUNICATION IN ANDROID

Erika Chin

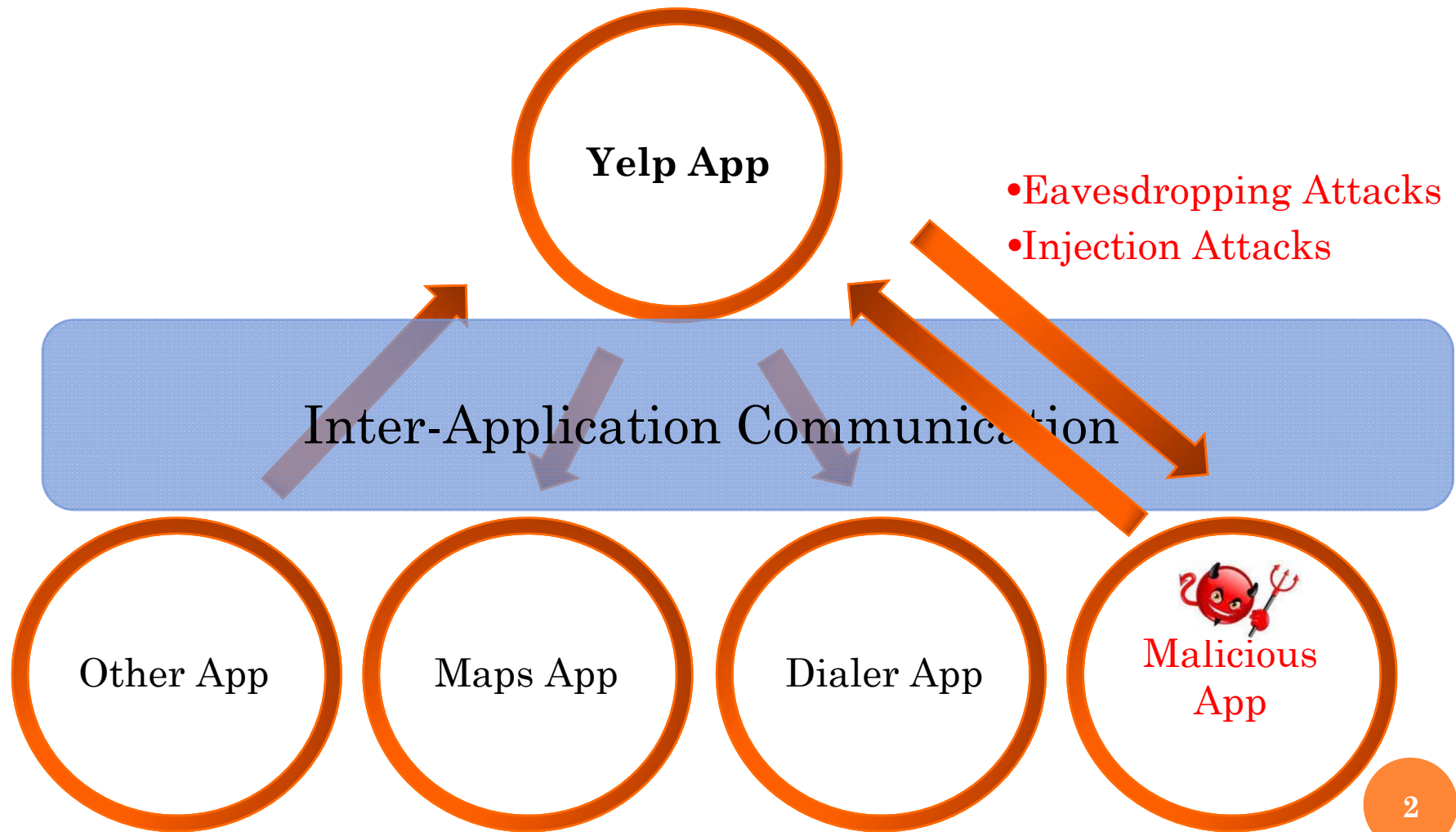
Adrienne Porter Felt

Kate Greenwood

David Wagner

UC Berkeley

INTER-APPLICATION COMMUNICATION

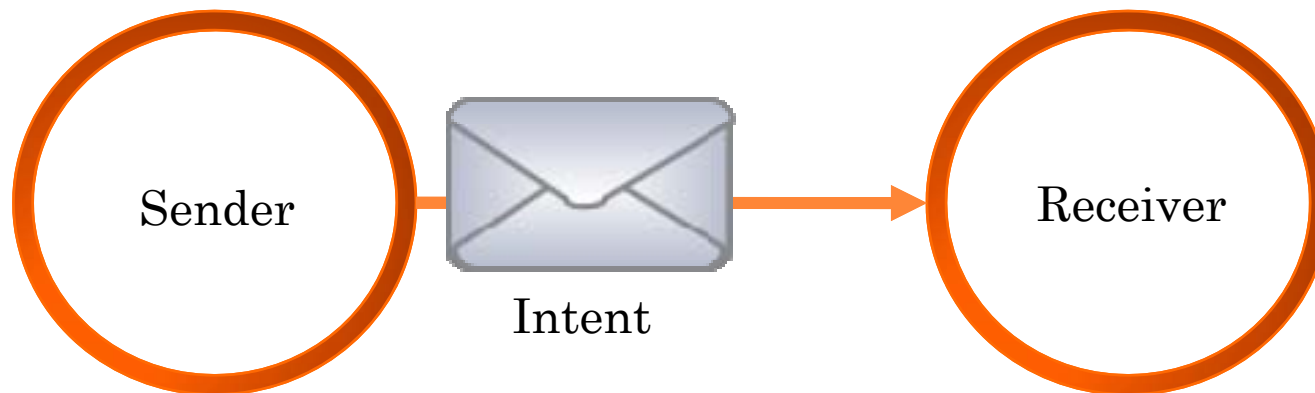


ORGANIZATION

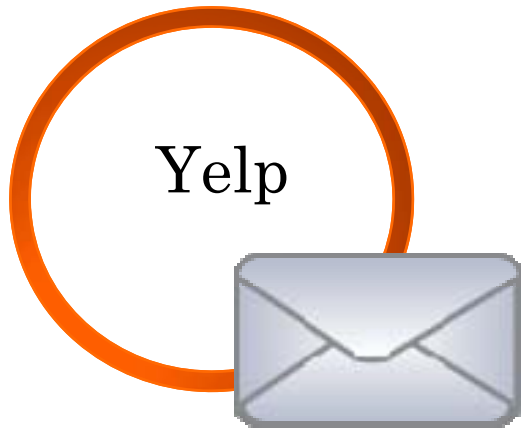
- Android communication model
- Security analysis of Android
- ComDroid
- Analysis of third-party applications
- Recommendations

ANDROID OVERVIEW

- **Intents** = Android IPC
- Applications are divided into **components**
- Intents can be sent between components
- Intents can be used for intra- and inter-application communication



EXPLICIT INTENTS



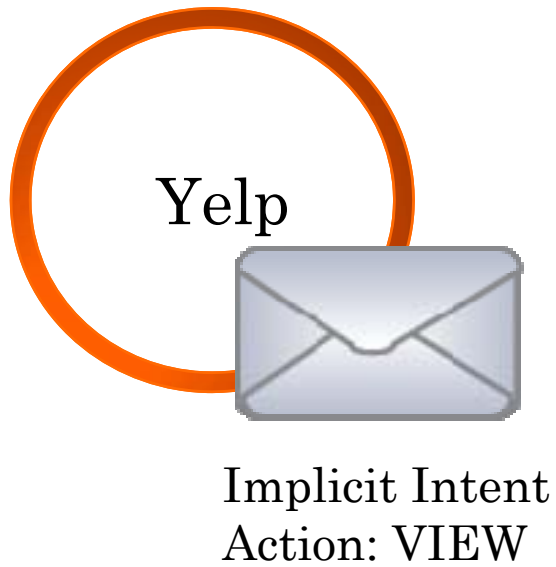
To: MapActivity

Name: MapActivity



Only the specified destination receives this message

IMPLICIT INTENTS



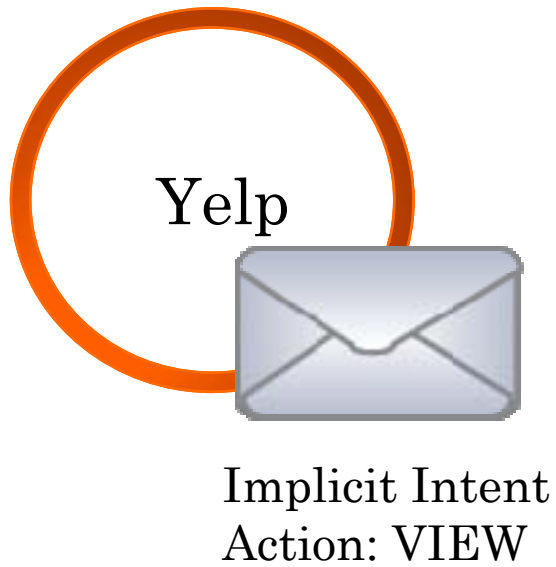
Handles Action: VIEW



Handles Action: DISPLAYTIME



IMPLICIT INTENTS



Handles Action: VIEW



Handles Action: VIEW



SECURITY ANALYSIS OF ANDROID

COMMON DEVELOPER PATTERN: UNIQUE ACTION STRINGS

IMDb App

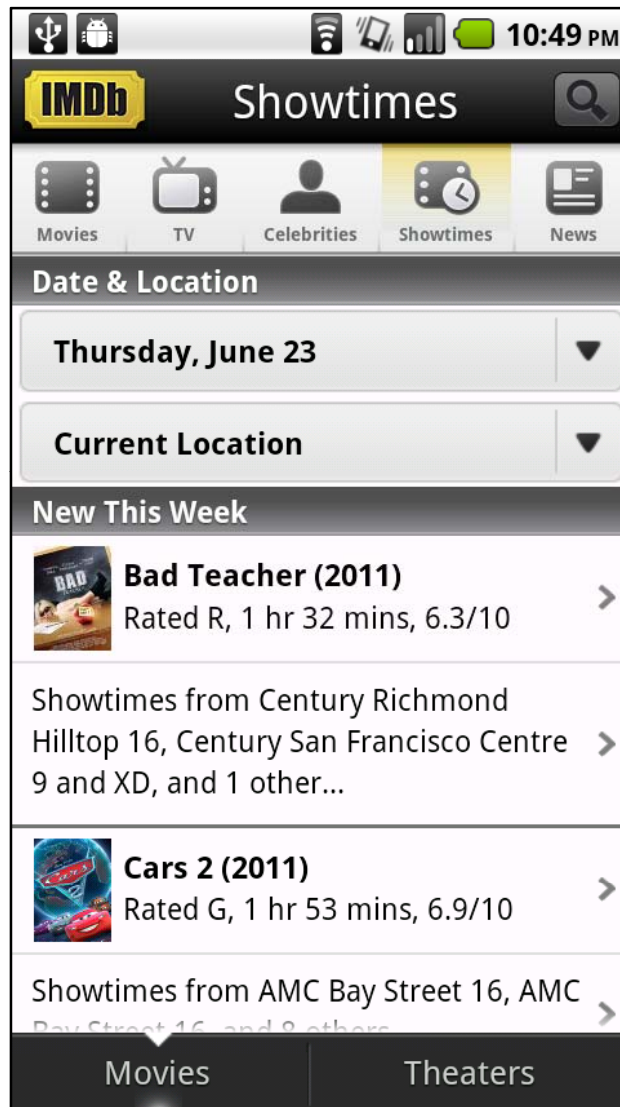
Handles Actions:
willUpdateShowtimes,
showtimesNoLocationError

Showtime
Search



Implicit Intent
Action:
willUpdateShowtimes

Results UI



COMMON DEVELOPER PATTERN: UNIQUE ACTION STRINGS

IMDb App

Handles Actions:
willUpdateShowtimes,
showtimesNoLocationError

Showtime
Search




Implicit Intent
Action:
willUpdateShowtimes

Results UI

ATTACK #1: EAVESDROPPING

IMDb App

Showtime Search



Implicit Intent Action:
willUpdateShowtimes



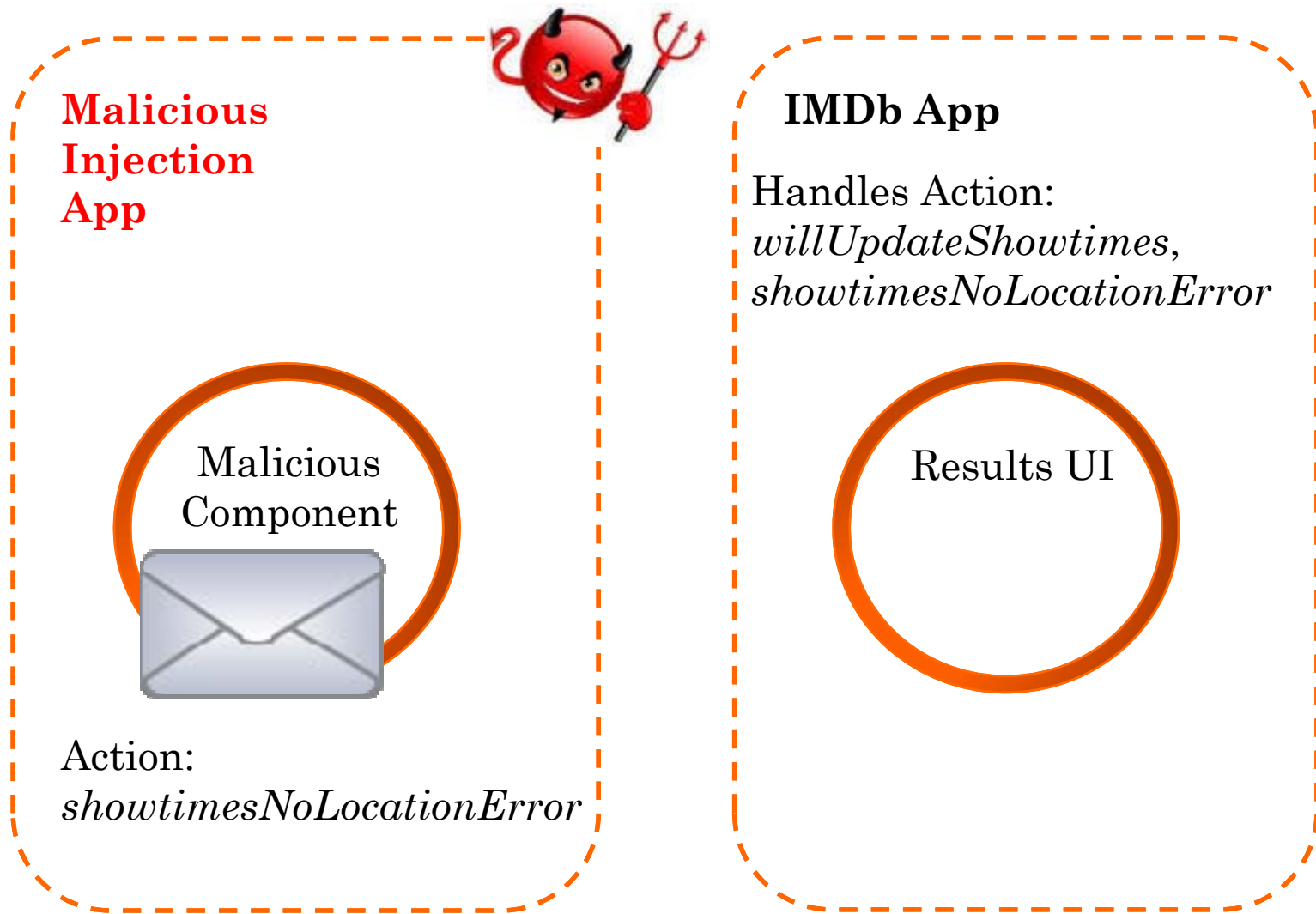
Eavesdropping App

Handles Action:
willUpdateShowtimes,
showtimesNoLocationError

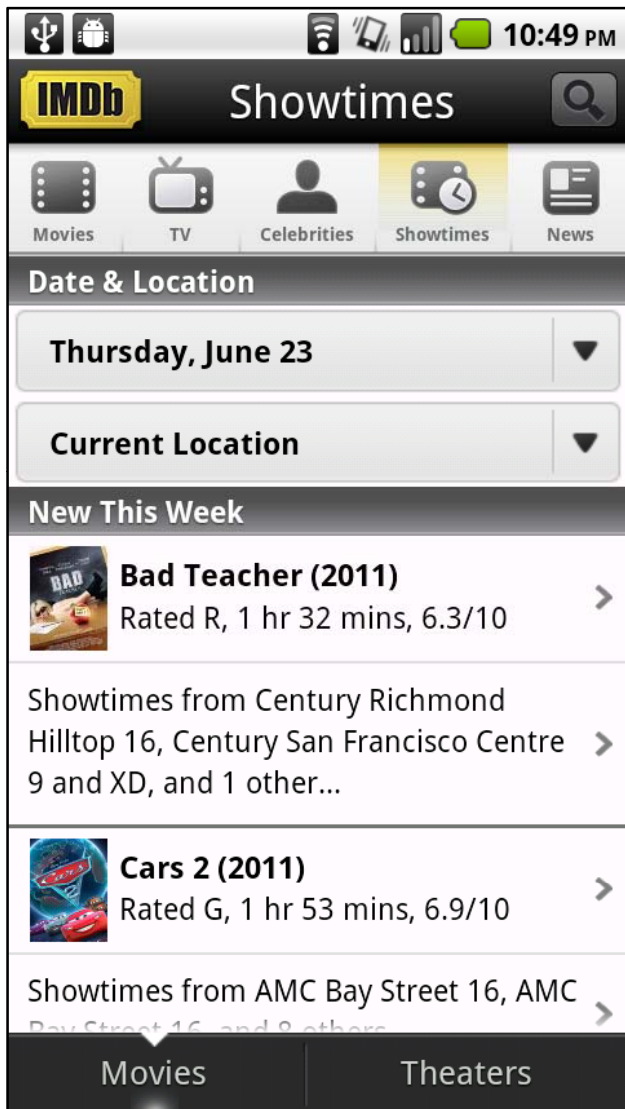
Malicious Receiver

Sending Implicit Intents makes communication public

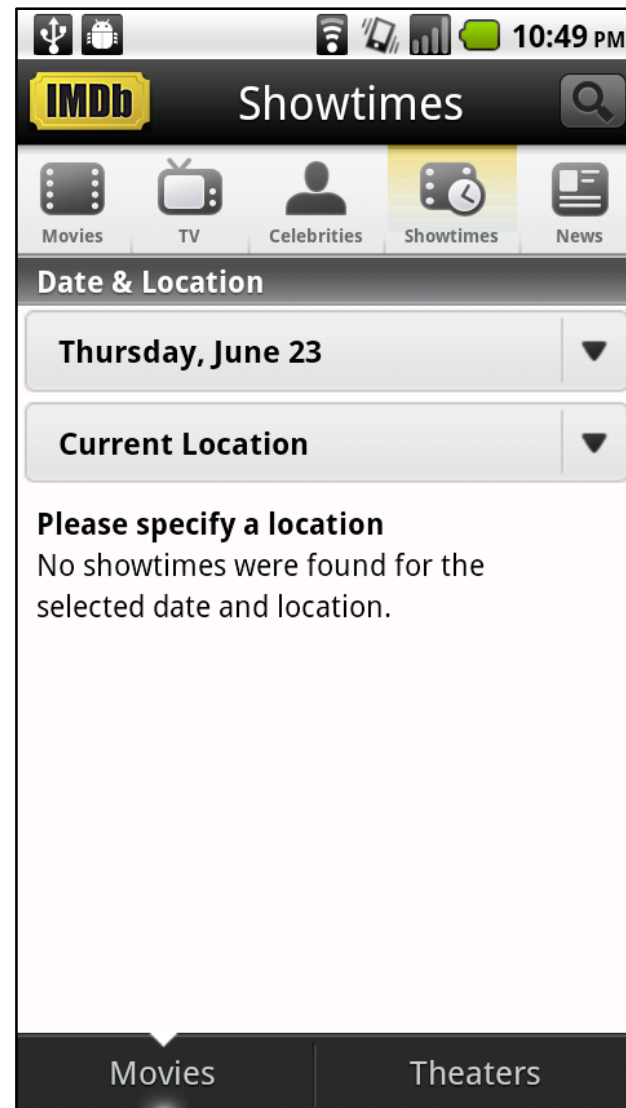
ATTACK #2: INTENT SPOOFING



Receiving Implicit Intents makes the component public



Typical case



Attack case

ATTACK #3: MAN IN THE MIDDLE



IMDb App

Handles Action:
willUpdateShowtimes,
showtimesNoLocation
Error

Showtime
Search



Action:
willUpdateShowtimes

Results UI

Man-in-the-Middle App

Handles Action:
willUpdateShowtimes,
showtimesNoLocationError

Malicious
Receiver

Action:
showtimesNoLocation
Error

ATTACK #4: SYSTEM INTENT SPOOFING

- Background – System Broadcast
 - Event notifications sent by the system
 - Some can only be sent by the system
- Receivers become accessible to all applications when listening for system broadcast

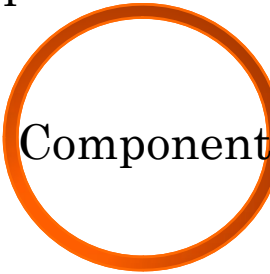
SYSTEM BROADCAST

System
Notifier



Action:
BootCompleted

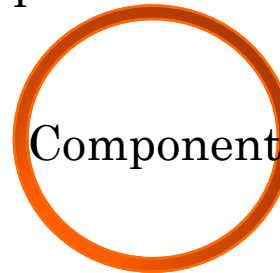
App 1



Component

Handles Action: *BootCompleted*

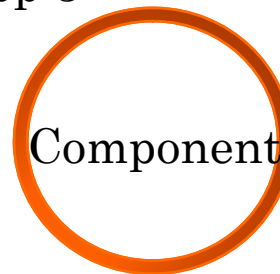
App 2



Component

Handles Action: *BootCompleted*

App 3



Component

Handles Action: *BootCompleted*

SYSTEM INTENT SPOOFING: FAILED ATTACK



**Malicious
App**

Malicious
Component



Action:
BootCompleted

App 1

Handles Action: *BootCompleted*

Component



SYSTEM INTENT SPOOFING: SUCCESSFUL ATTACK



**Malicious
App**

Malicious
Component



To: App1.Component

App 1

Handles Action: *BootCompleted*

Component

REAL WORLD EXAMPLE: ICE APP

- ICE App: Allows doctors access to medical information on phones
- Contains a component that listens for the *BootCompleted* system broadcast
- On receipt of the Intent, it exits the application and locks the screen

REAL WORLD EXAMPLE: ICE



COMDROID



ComDroid analyzes applications to detect Intent-based attack surfaces

EVALUATION

- Manually verified ComDroid's warnings for 20 applications
- **60%** of applications examined have at least 1 exploitable IPC vulnerability

Type	# of Warnings	# of Apps
Severe Vulnerability	34	12
Bad Practice	16	6
Spurious Warning	6	6

RECOMMENDATIONS

- Treat inter- and intra-application communication as different cases
- Prevent public internal communication
 - 21% of severe vulnerabilities
 - 63% of bugs due to bad practice
- Verify system broadcasts
 - 6% of severe vulnerabilities
 - 13% of bugs due to bad practice
- Can be fixed by either developers or platform

RELATED WORK

- Enck et al. – introduces information leakage through Broadcast Intents and information injection into Receivers
- Burns – discusses other common developers' errors

CONCLUSION

- Applications may be vulnerable to other applications through Android Intent communication
- Many developers misuse Intents or do not realize the consequences of their program design
- 60% of applications examined had at least 1 vulnerability
- ComDroid tool to be publically accessible soon at www.comdroid.org

Thank you!

Any questions?