

# Screenmilker: How to Milk Your Android Screen for Secrets

Chia-Chi Lin, Hongyang Li<sup>1</sup>,  
Xiaoyong Zhou<sup>2</sup>, and XiaoFeng Wang<sup>2</sup>

*<sup>1</sup>University of Illinois at Urbana-Champaign*

*<sup>2</sup>Indiana University at Bloomington*



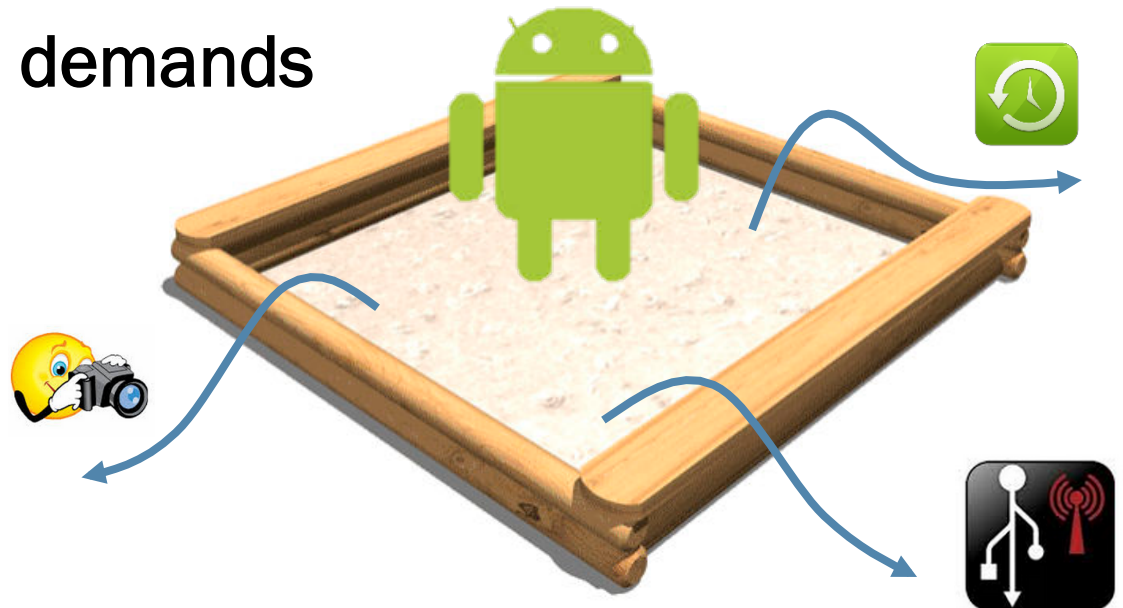
# Android Security VS. App Demands

## Android security design

- No Direct access system resources
- No Reading/Writing outside it's own directory
- No installing/uninstalling other apps

## User's/developer's demands

- Capture screen
- Backup
- USB Tethering



# One Solution: Root the phone

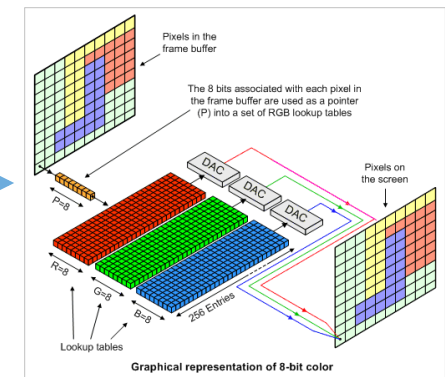
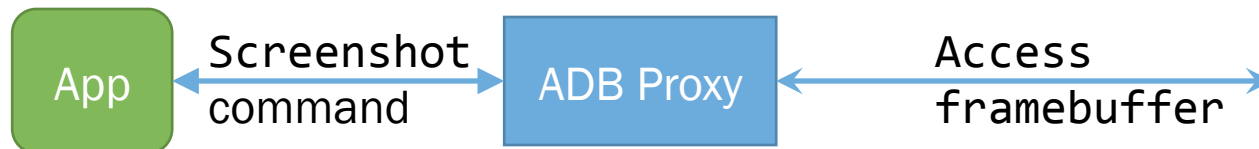


# An Legitimate Alternative: ADB Proxy

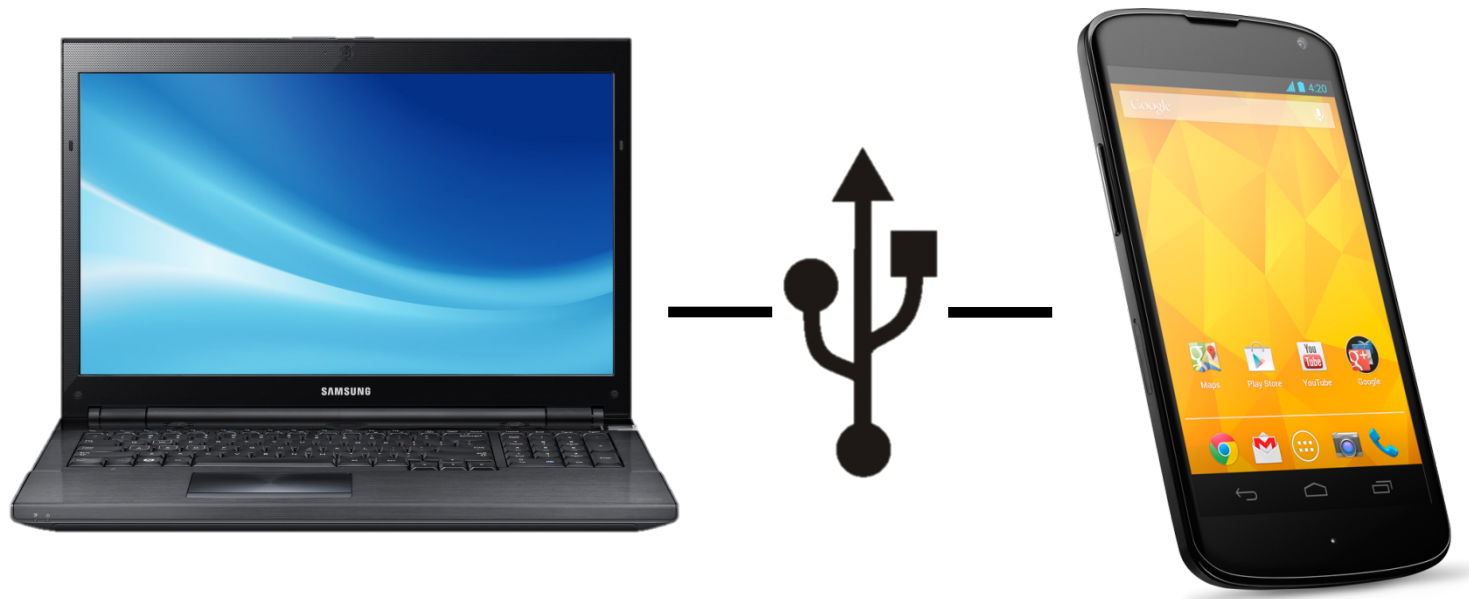
- Android Debug Bridge (ADB)
  - A versatile command line tool that lets user communicate with his device
  - A set of capabilities
    - Install/Uninstall
    - Pull/Push data
    - Take screenshots / Record screen
    - ...
- How app can use ADB? -- proxy

# ADB Proxy

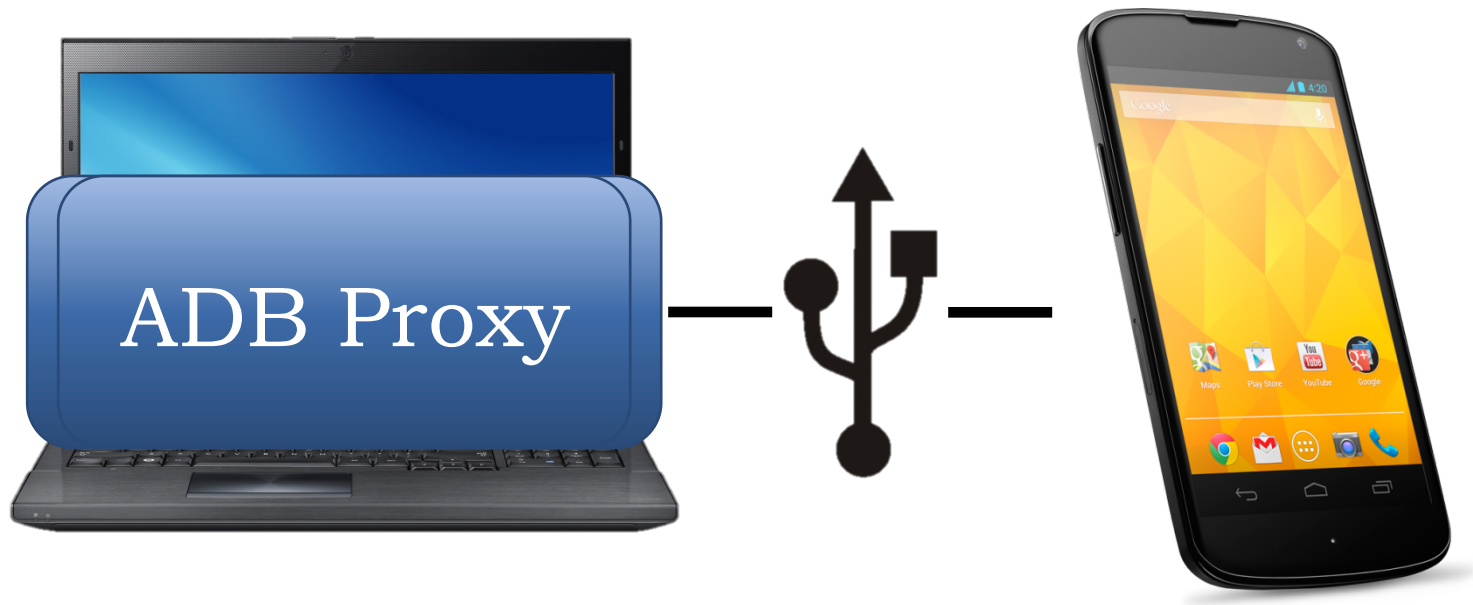
- An native executable implemented by developer
- Runs on the phone as shell user to provide privileged services to other apps
- ADB proxy is legitimate
  - Apps using this approach have tens of millions of downloads
  - No objections from Google



# 1. Turn on USB Debugging and Connect Android to a PC

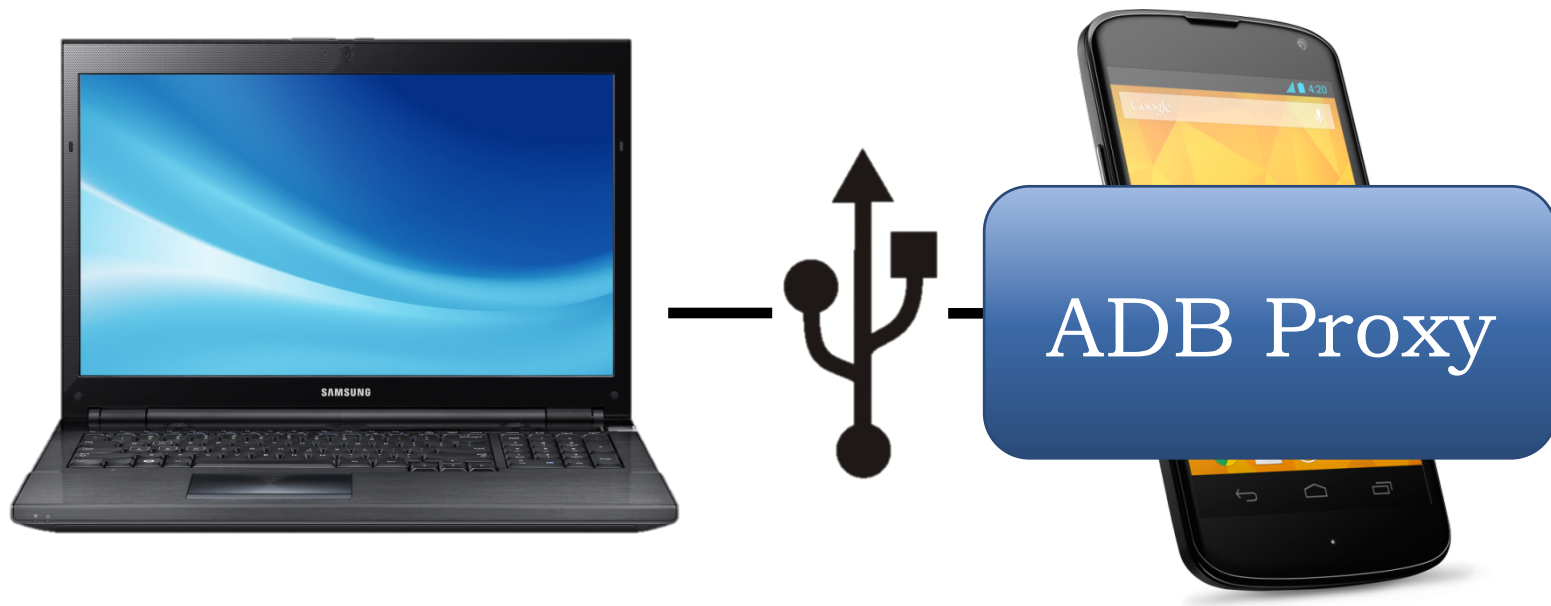


## 2. Run a Script on the PC to Install a ADB Proxy on Android



- ADB Proxy has the same capabilities as ADB

### 3. Disconnect Android from the PC





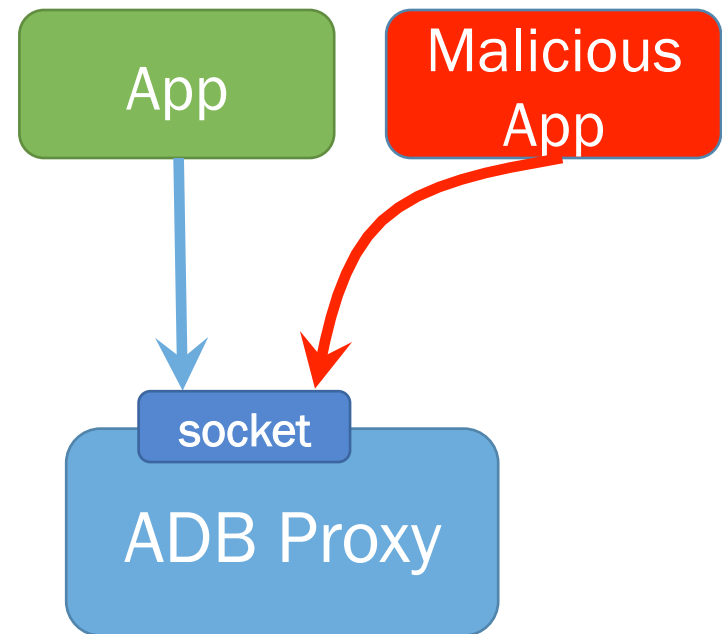
# Apps Using ADB proxy

- Screenshot apps
  - Very popular on Google Play
- USB Tethering Apps
- Sync and Backup Apps

App Name	Total Installs
Screen Capture – No Rooting 2.2	1,000,000 – 5,000,000
Screenshot Free	1,000,000 – 5,000,000
Screenshot UX Trail	1,000,000 – 5,000,000
No Root Screenshot It	100,000 – 500,000
Screenshot and Draw Trail	100,000 – 500,000
Screenshot Ultimate	100,000 – 500,000
ShakeShot Trail	100,000 – 500,000
NoRoot Screenshot Lite	50,000 – 100,000

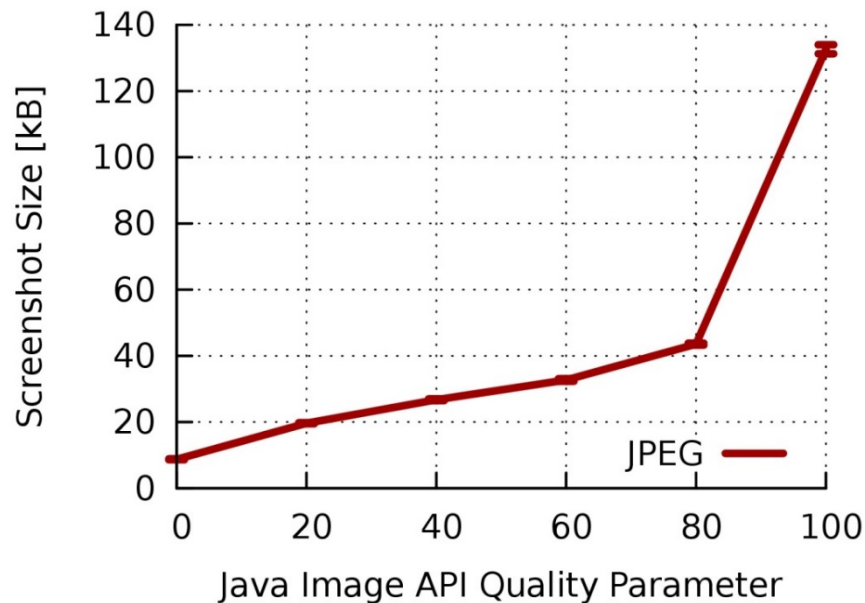
# Security Implications

- No Access Control
  - Local socket
  - Any apps with the INTERNET permission can connect to ADB proxy
- A malicious app could command ADB proxy to
  - Take screenshots
  - Install applications



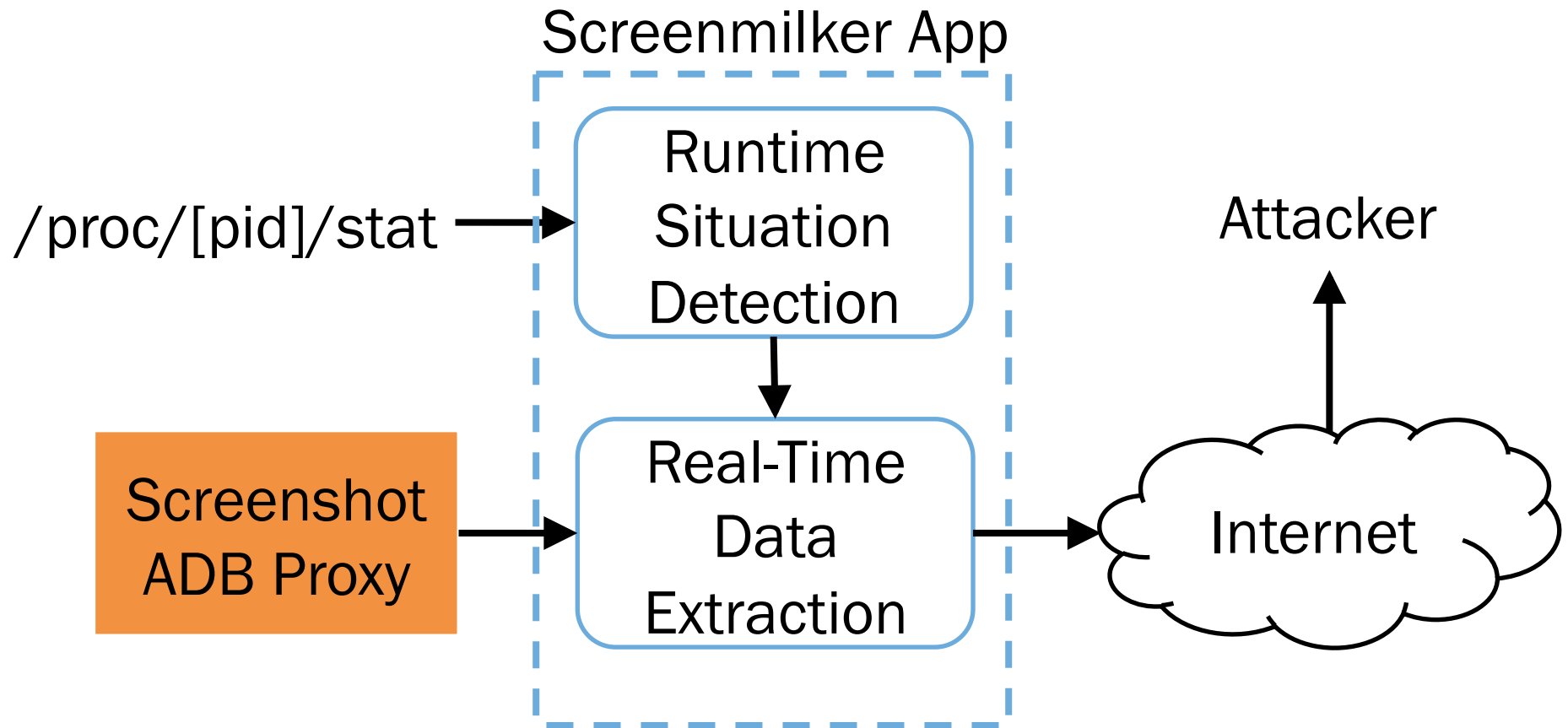
# Naïve attacks are not stealthy

- Streaming pictures to adversary consumes too much bandwidth
- Running OCR locally uses too much CPU and memory



For a 2-Mbps Upload  
Bandwidth, Only 2 Screenshots  
Can Be Sent Out Every Second

# Our Attack



# Detect Screenshot Proxy

- Build a database of screenshot apps
- Use call *PackageManager* to get the list of apps on the device
- Alternatively, scan TCP ports ADB proxies use



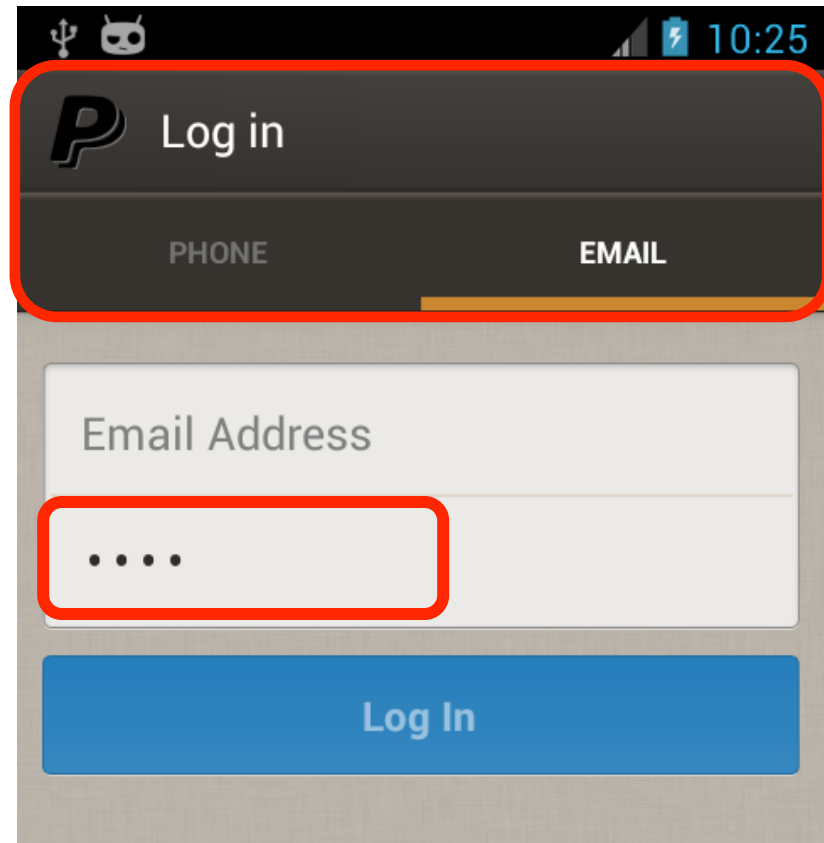
# Runtime Situation Detection

- Detect target apps (e.g., banking apps) through **PackageManager**
- Probe ***/proc/[pid]/stat*** to monitor apps' activities
  - Check the `cpu utime` change of target app
- Monitor the soft keyboard app to identify whether user is typing on the soft keyboard
  - `com.google.android.inputmethod.latin`

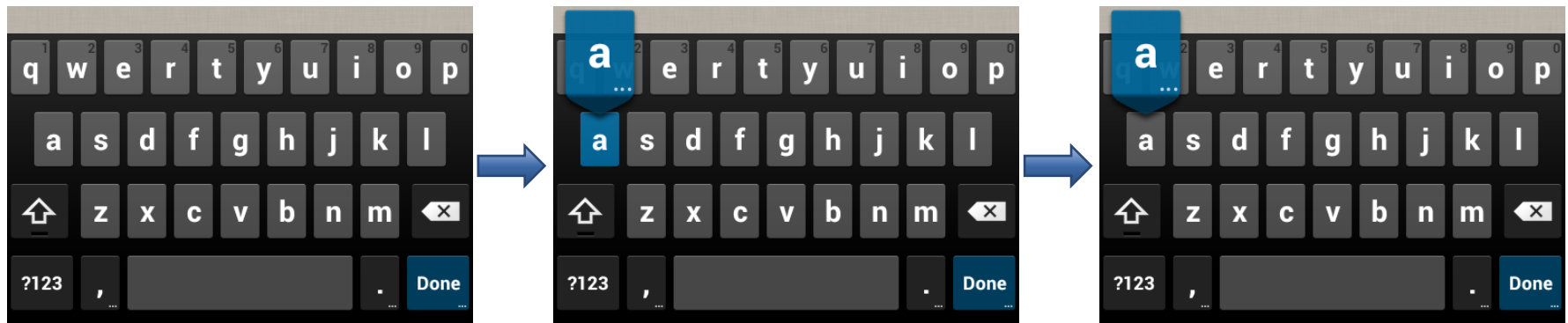


# Detecting Application States

1. Get Screen Orientation
2. Take screenshots
3. Extract title bar
4. Match the title bar against app state database

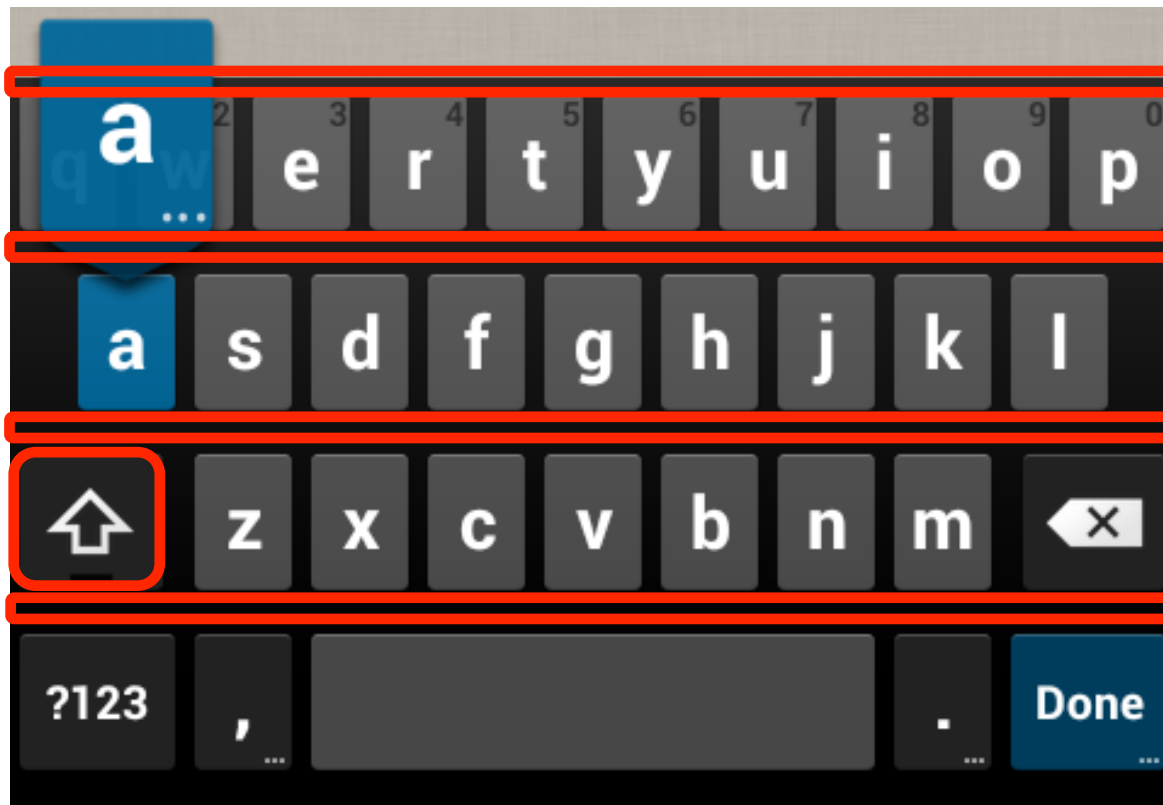


# Real-Time Keystroke Analysis

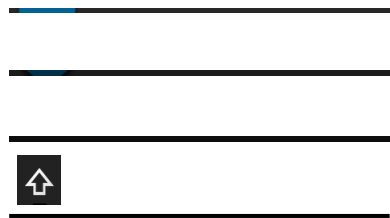




# Fingerprinting the Soft Keyboard



# Determining the Keystroke

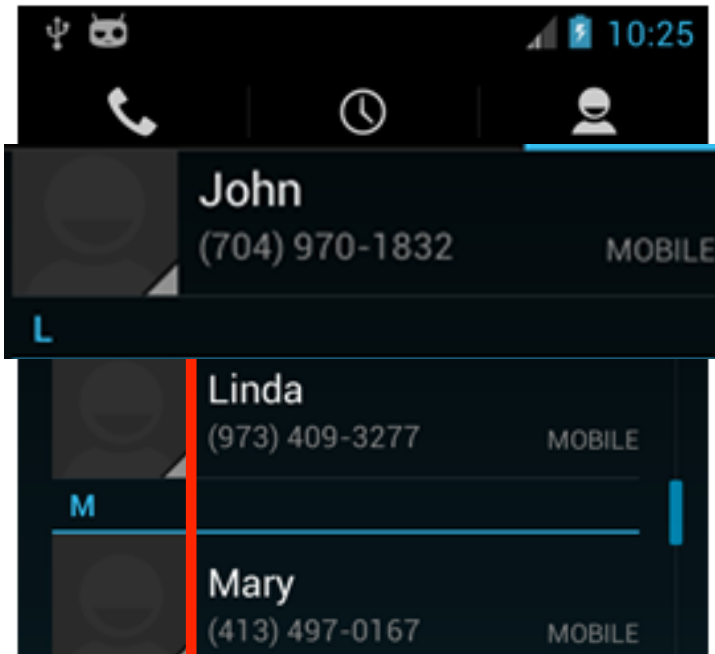


222054093



CRC32 Value	Keystroke
222054093	a
8599545	b
4181574192	c
...	...

# Real-Time Contact Collection

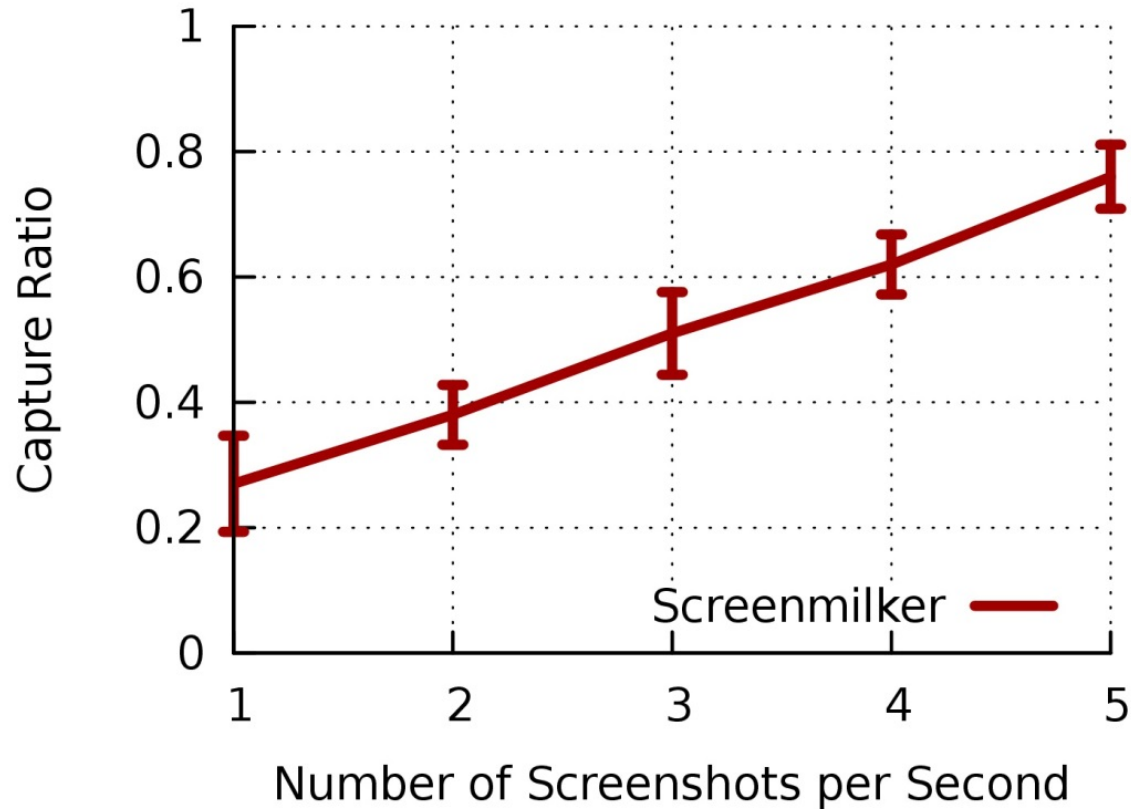


CRC

# Evaluations

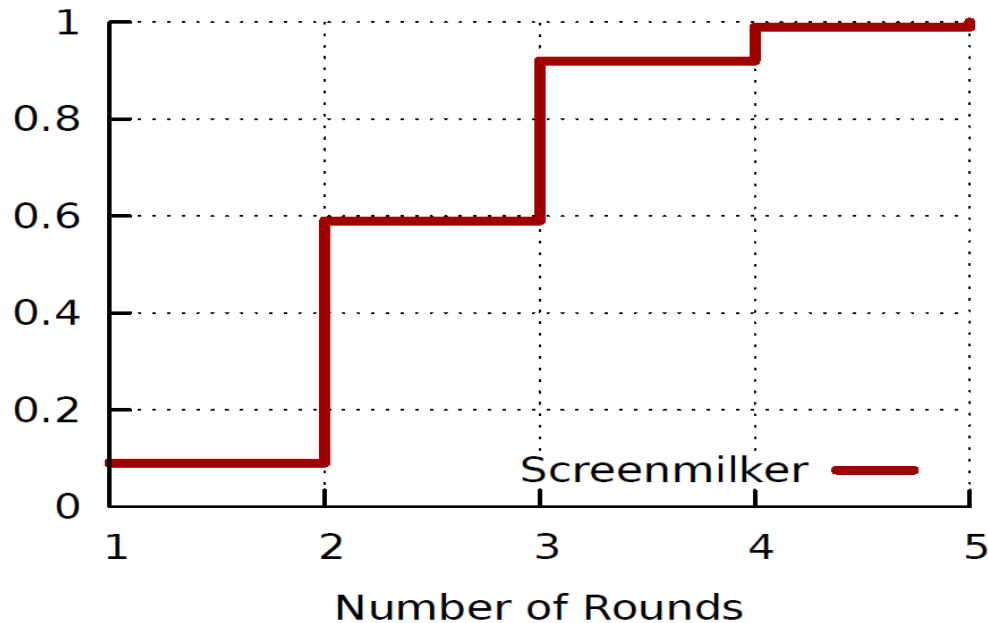
# Effectiveness: Single Key Stroke Capture Ratio

Capture Ratio Increases From 27% to 76% as the Screenshot Rate Increases



# Password Extraction

- Experiment setup
  - 10-character passwords
  - 5 banking apps [American Express, Chase, Citi, PayPal and Wells Fargo]
  - 40 password entering for each app
- How many rounds to recover a password?
  - Screenmilker may miss the moment for some keystrokes



## Rounds to Extract Entire Password

App	Average Number of Rounds
American Express US	2.625
Citi Mobile	2.525
Chase Mobile	2.325
PayPal	2.75
Wells Fargo Mobile	2.45

# CPU run time

	<b>Extraction Function</b>	<b>Time [ms]</b>
General	<b>Initialize Hash Table [one time]</b>	1.389
	Take a Screenshot [not controllable by Screenmilker]	161.314
Keystroke Extraction	<b>Fingerprint the Image Features</b>	0.388
	<b>Lookup Hash Table</b>	0.220
Contact Collection	<b>Obtain Position of Text</b>	3.018
	<b>Segment and Map Text</b>	2.916



# Memory Consumption

App	Memory [Kbytes]
<b>Screenmilker [situation detection]</b>	286.308
Clock	294.072
<b>Screenmilker [contact collection]</b>	295.279
<b>Screenmilker [keystroke extraction]</b>	295.364
Calculator	295.464
Google Talk	310.844
Instagram	326.244
Pandora Internet Radio	356.332
Facebook	365.384
Browser	391.912
Temple Run 2	436.712

# Power Consumption

App	Power [mW]
<b>Screenmilker [situation detection]</b>	4.1
<b>Screenmilker [contact collection]</b>	8.3
Google Talk	47.8
Clock	52.1
Calculator	91.8
<b>Screenmilker [keystroke extraction]</b>	101.6
Instagram	155.8
Pandora Internet Radio	213.5
Facebook	252.1
Browser	374.8
Temple Run 2	529.2

# Mitigations: Access Control on ADB Proxy

- Utilize *iptables* to control local-socket communication
- Users need to explicitly grant apps permission to communicate with local servers
- We build a service to add *iptables* rules accordingly

# Conclusions

- ADB proxy is a popular workarounds that grant privileged capabilities to 3<sup>rd</sup> party apps
- Without proper protection, ADB proxy could be exploited by malicious apps to extract sensitive information from the phone as demonstrated by Screenmilker
- From our evaluation, we show that malicious app can effectively and stealthily extract information from screenshots

**Thank You! Questions?**