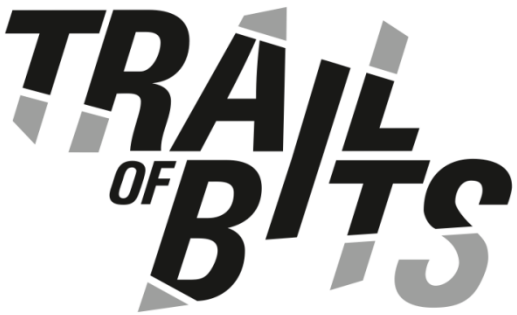
Mobile Exploit Intelligence Project

Dan Guido, Trail of Bits
Mike Arpaia, iSEC Partners

Blackberry, 06/2013



Mobile Device Security Thesis

- Mobile devices are loading up with data
  - E-mail, line of business apps, login credentials…
- Lots of possibilities to compromise mobile devices
  - Insecure data storage, app-to-app, NFC, web browser, …
- Very few vectors explored in *actual* attacks
  - Why is that? What motivates attackers? Isn't it easy?
- What attacks do I need to defend against *now*?
  - Actual vs Probable vs Possible
  - How will things change (or not) tomorrow?

Millions of Mobile Attacks

Attack Vector

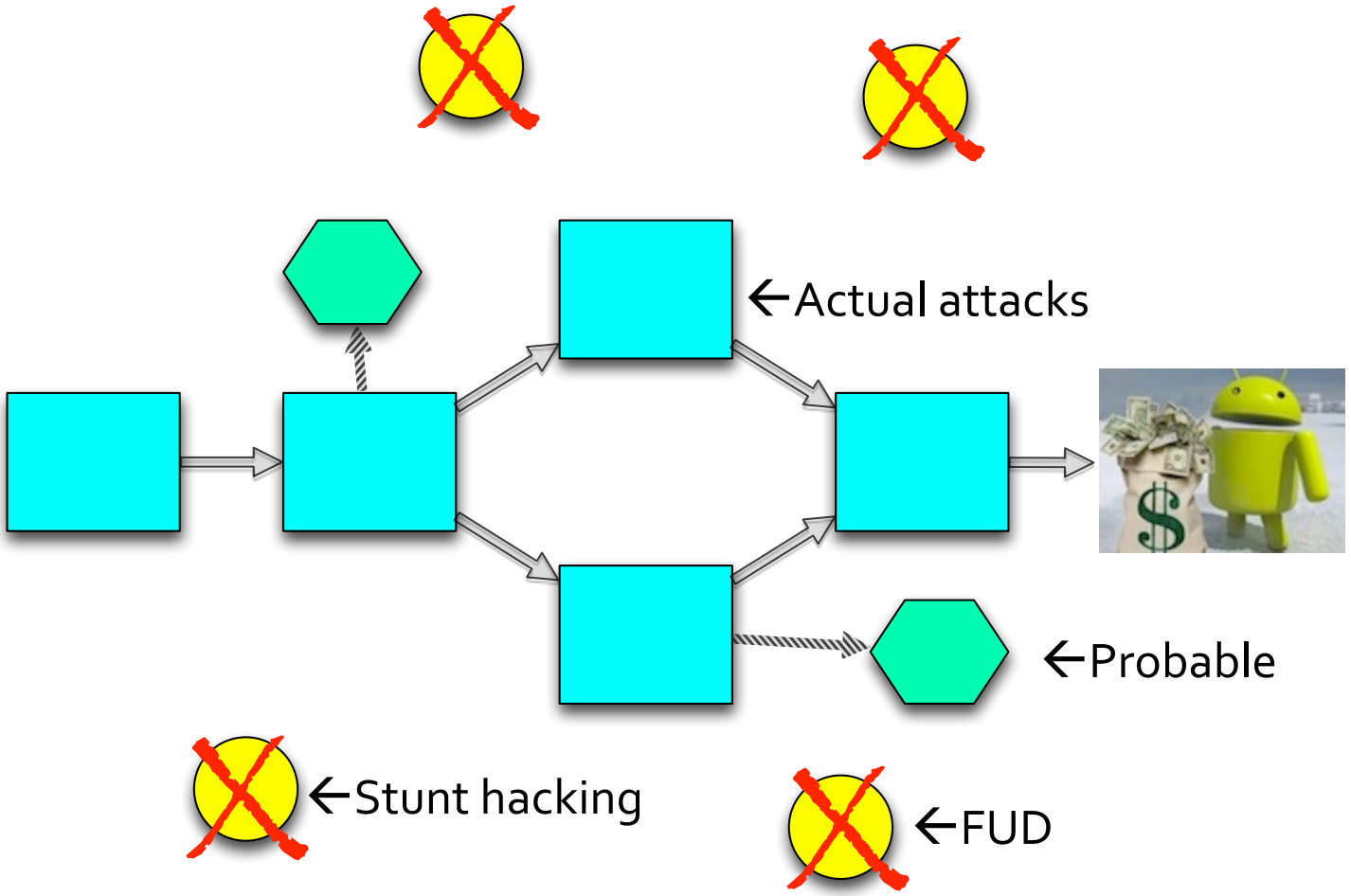
Exploits

Platform

* Android and iOS, 2011-2012

What are we doing wrong?

Look at Attacks, Not Vulnerabilities



Attackers choose the least cost path to their *objective*

Attacker Math 101

- Cost(Attack) < Potential Revenue
  - Attacks must be financially profitable
  - Attacks must scale according to resources

- Cost(Attack) = Cost(Vector) + Cost(Escalation)
  - What we know from Mobile OS architectures

Cost of Attack

- Ease
- Enforcement
- Established Process

Potential Revenue

- # of Targets
- Value of Data
- Ability to Monetize

Mobile Malware Today

Setting it up

- Develop malware
- Add malware to many applications

Scaling it out

- Put malware online (app store)
- Drive installations (SMS, e-mail, ads)

Accessing data

- Exploit a flaw to access user data
- Send stolen data back to attacker

TRAIL OF BITS

Intrusion Kill Chains

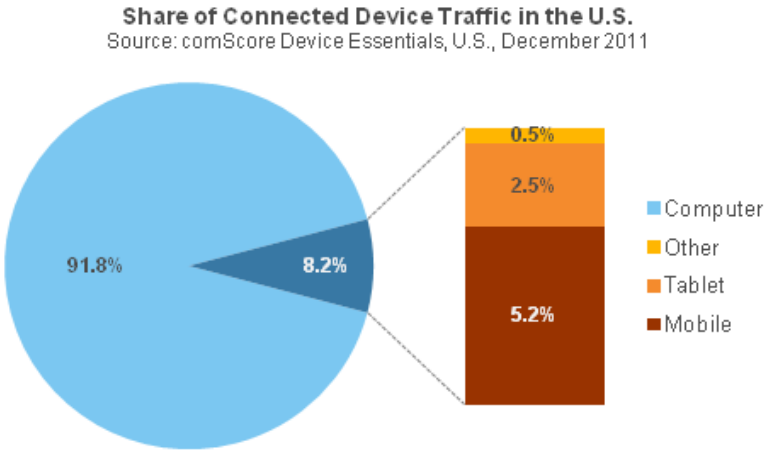
- Systematic process that an intrusion must follow
  - Deficiency in one step will disrupt the process
- Evolves response beyond point of compromise
  - Prevents myopic focus on vulnerabilities or malware
  - Identifies attacker reuse of tools and infrastructure
- Guides our analysis and implementation of defenses
  - Align defenses to specific processes an attacker takes
  - Force attackers to make difficult strategic adjustments

Where are Mobile Drive-Bys?

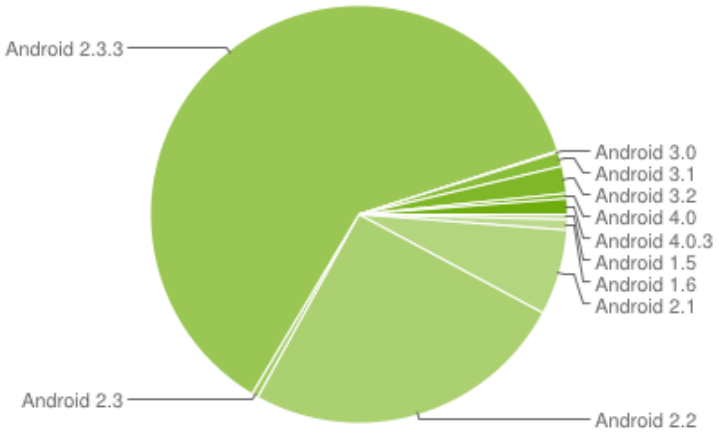
Mobile Town

Desktop City

Less Mobile Web Targets



~8% of total web traffic comes from mobile devices

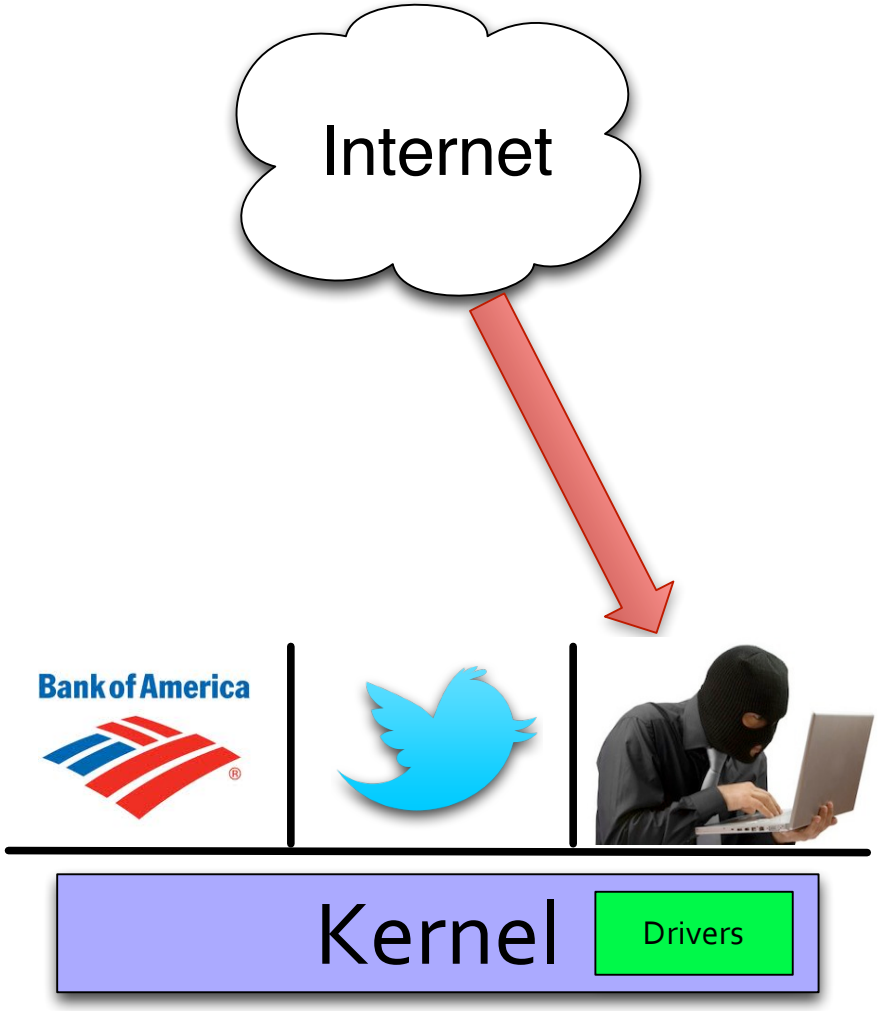


Breakdown by version / features (+ varying rates of feature support)

Lack of Ads Limits Targeting Potential

Normal Website | Mobile Website | Mobile App

Browser Exploits are Harder



Browser Permissions

- .INTERNET
- .ACCESS_FINE_LOCATION
- .ACCESS_COARSE_LOCATION
- .ACCESS_FINE_LOCATION
- .ACCESS_DOWNLOAD_MANAGER
- .ACCESS_NETWORK_STATE
- .ACCESS_WIFI_STATE
- .SET_WALLPAPER
- .WAKE_LOCK
- .WRITE_EXTERNAL_STORAGE
- .SEND_DOWNLOAD_COMPLETED_INTENTS

TRAIL OF BITS

Mobile Browser Tradeoffs

Cost of Attack

- How to Drive Installs?
  - No Flash
  - Limited Ads
- 2x Exploits Required
  - Where to get them?
  - Version support?
- New Payload Required
  - What code to run?
  - How to persist access?

Potential Revenue

- Lower # of targets
  - Mobile web, meh
- Fragmented market
  - Different vendors
  - Different versions

Scaling the Setup

1. Develop malware

2. Add malware to many applications

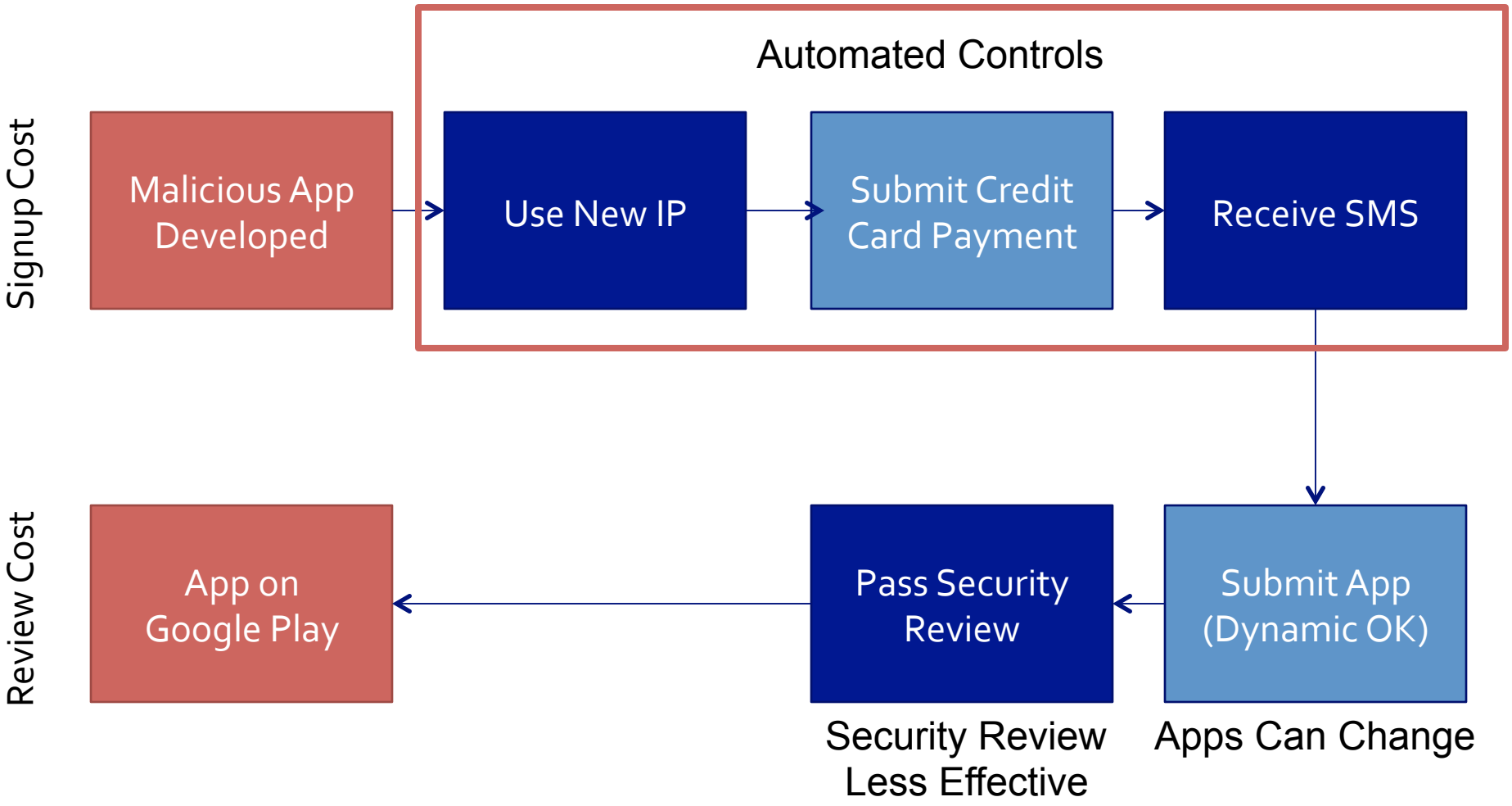
3. Put malware online

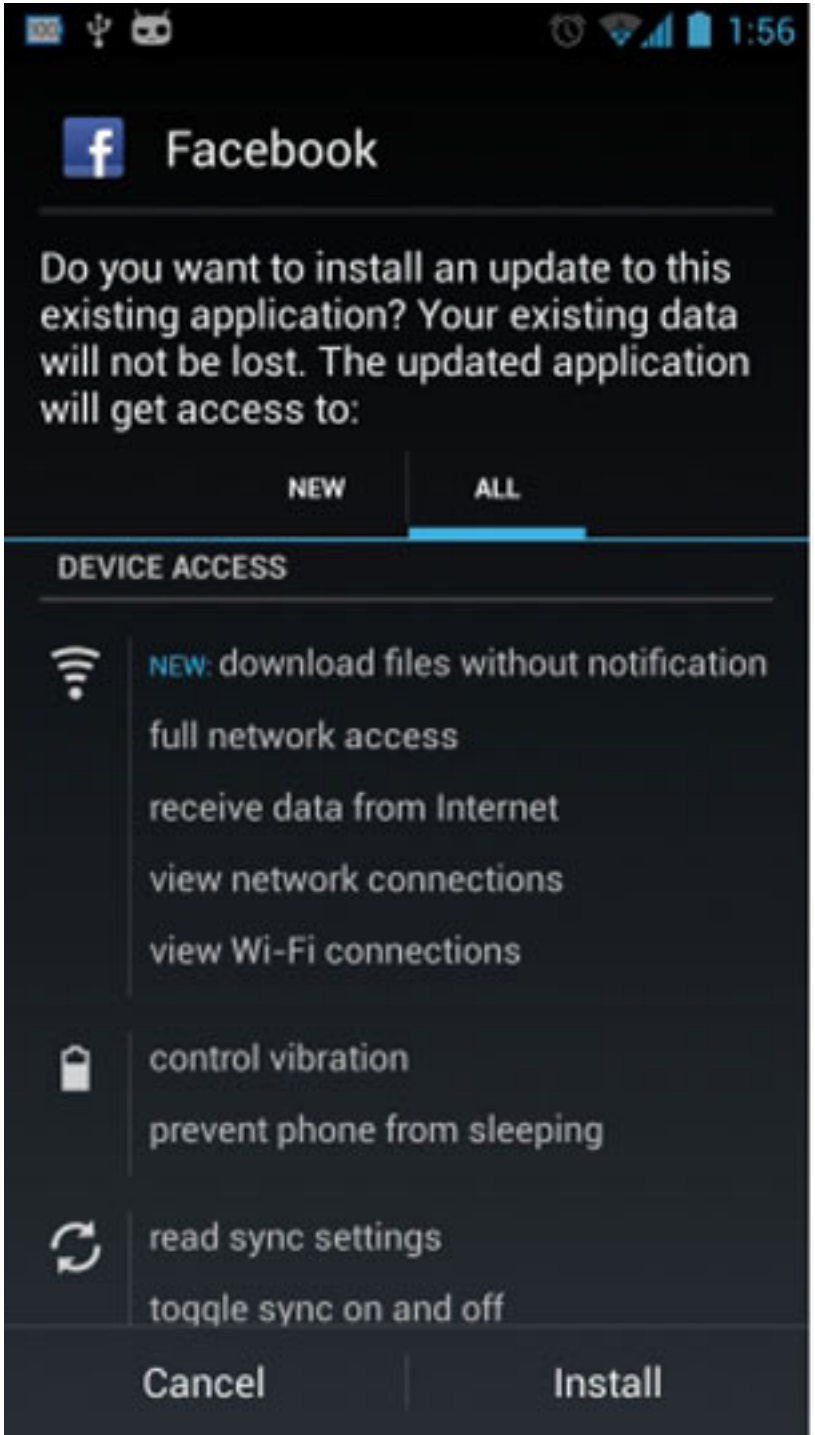
Better Targets, Unique to Mobile

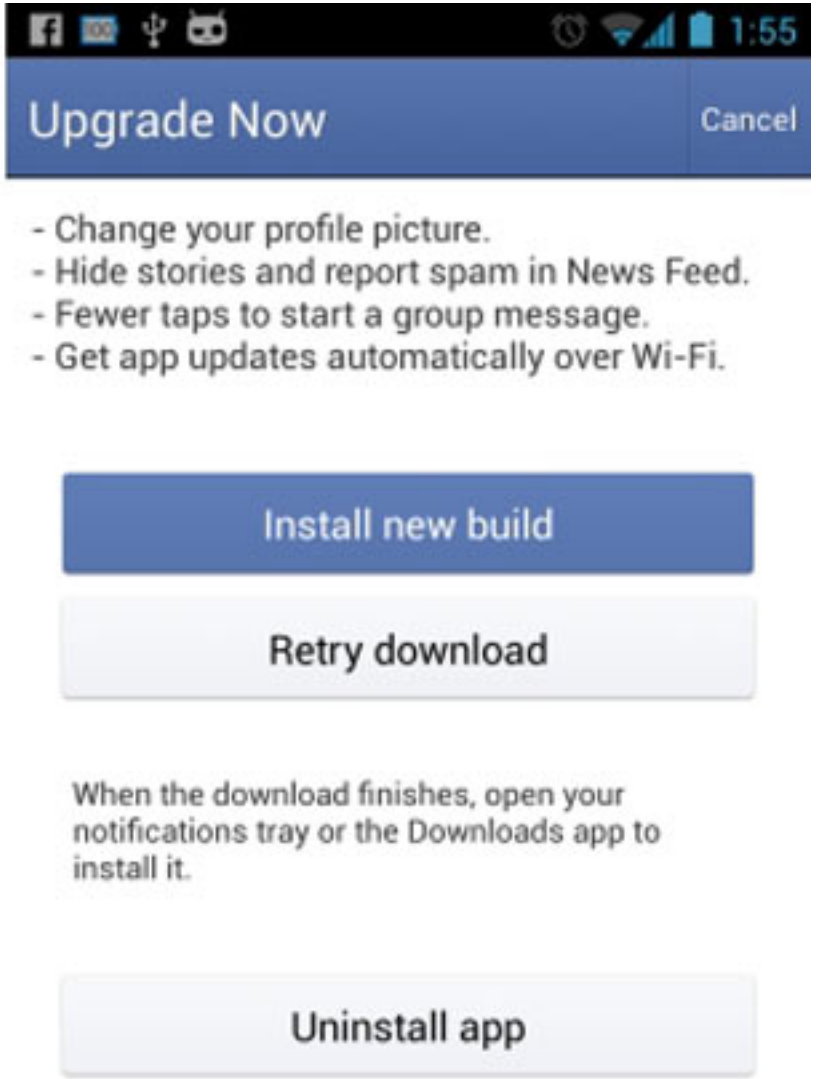
Incentives	Browser Exploits	Malicious Apps
# of Targets	Minimal	All Devices (300 mil+)
Ability to Target	Ads	App Store SEO, Lures
Ease of Exploit	Multiple Exploits	Single Exploit
Enforcement	Anonymous	Anonymous?

App stores look like a great value proposition!

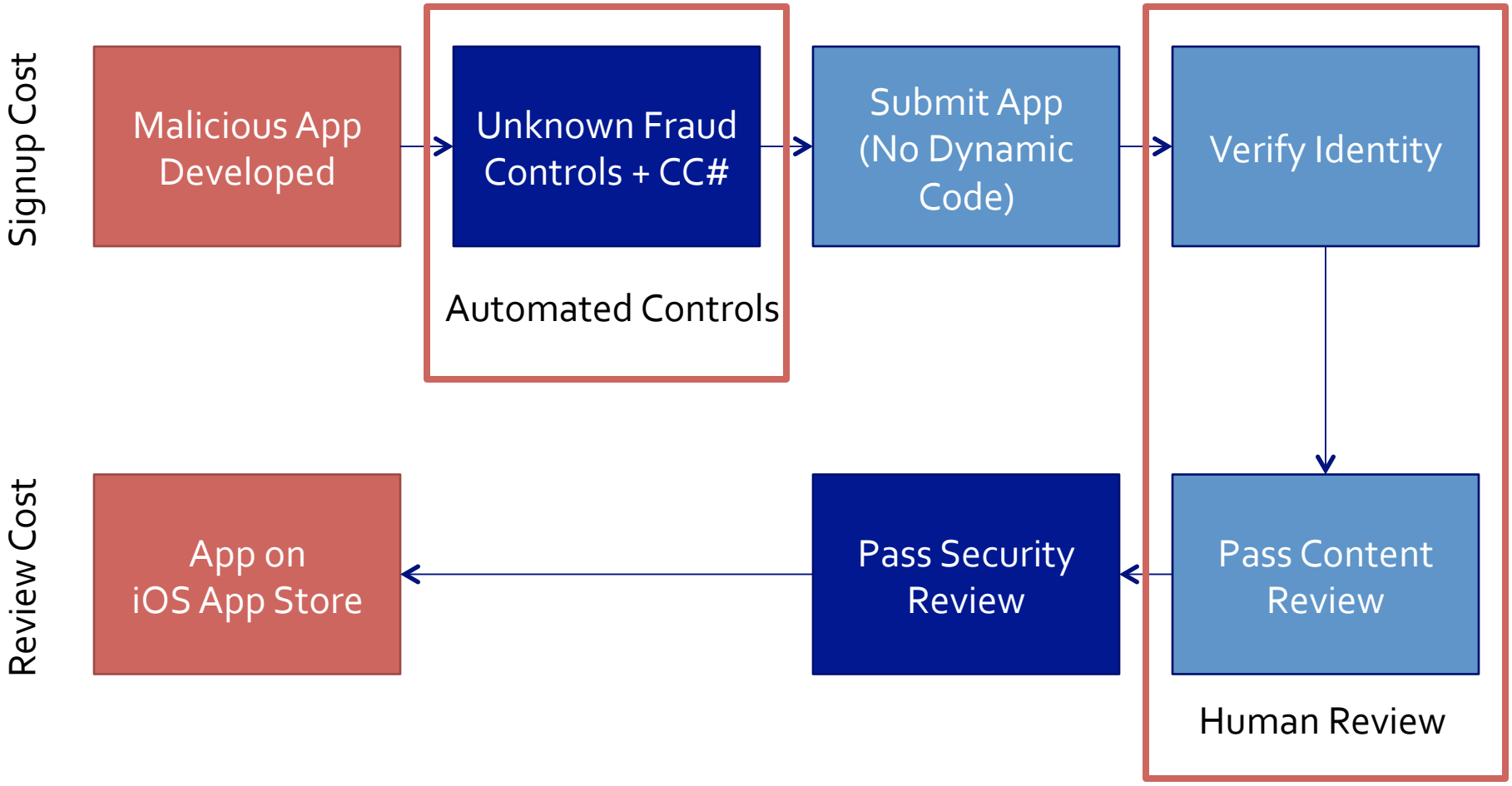
Scaling Malicious App (Android)



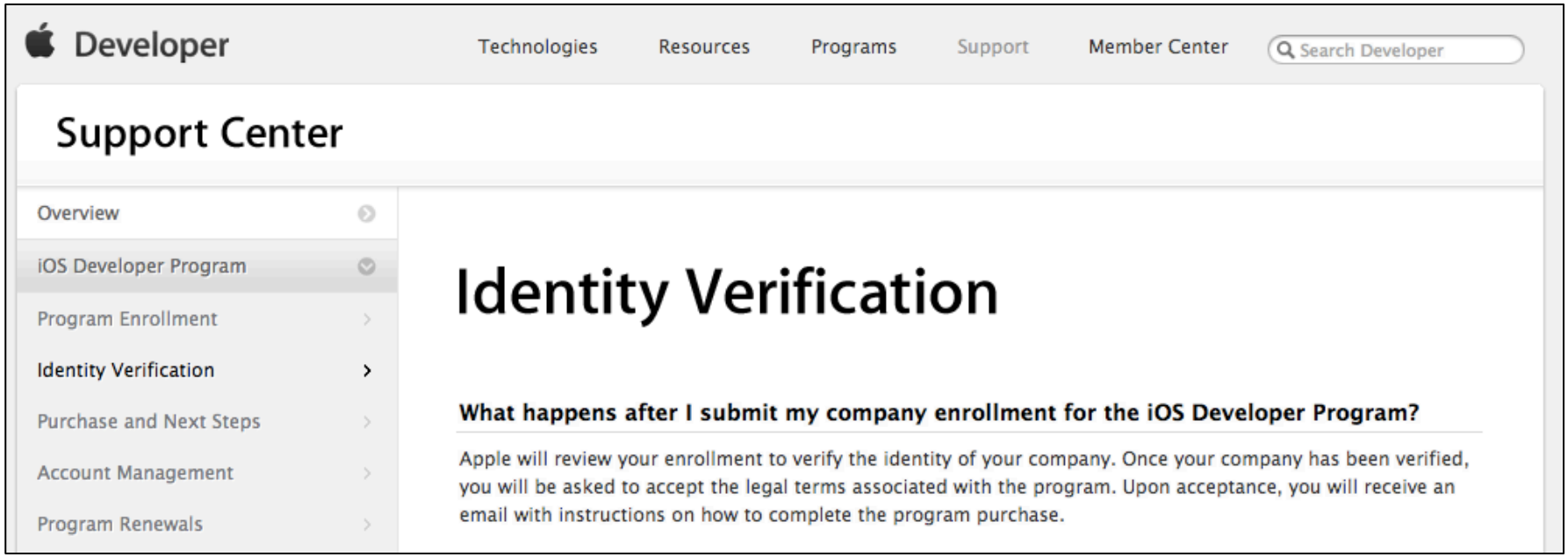

1:56
Facebook
Do you want to install an update to this existing application? Your existing data will not be lost. The updated application will get access to:
NEW
ALL
DEVICE ACCESS
NEW: download files without notification
full network access
receive data from Internet
view network connections
view Wi-Fi connections
control vibration
prevent phone from sleeping
read sync settings
toggle sync on and off
Cancel
Install


1:55
Upgrade Now
Cancel
- Change your profile picture.
- Hide stories and report spam in News Feed.
- Fewer taps to start a group message.
- Get app updates automatically over Wi-Fi.
Install new build
Retry download
When the download finishes, open your notifications tray or the Downloads app to install it.
Uninstall app

Scaling Malicious App (Apple)



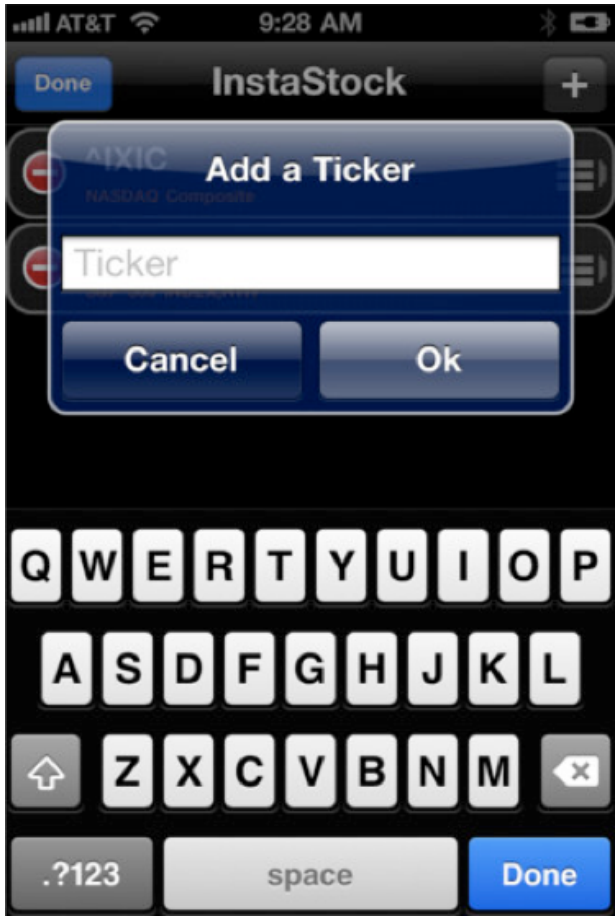
Think Different



- Automate new LLCs > Automate new CC#/SMS/IPs
  - Forces malware authors to scale with humans
- Restrict dynamic code by policy
  - Easy to detect attempts to circumvent
  - Enables review of code that actually runs on device

Charlie Miller, a Success Story

- Identified vulnerability in iOS code-signing
  - Send down new executable code at runtime
  - Bypass Apple review (default on Android!)
- InstaStock got Charlie InstaBanned
  - Removed from dev program
- Charlie can't submit any more apps
  - Get fake gov-backed ID, LLC and DUNS #
  - Or, another human with those documents



Malicious App Campaigns

Apple App Store

Google Marketplace

* data from 2012, still the same in spirit

Scaling the Heist

5. Access data outside the app sandbox

6. Send stolen data to a remote location

7. Abuse the data somehow to make money

Jailbreak Development

Mitigation	iOS	Android	BB10
Code Injection (1)	Code Signing	No-eXecute	No-eXecute
Randomization (2)	Strong ASLR	ASLR	ASLR
Containment (3)	Seatbelt	UNIX Perms	Abilities PathTrust
Shell Available (4)	No	Yes	Yes

1. Payload development w/ code-signing is a PITA (Partial vs Full ROP)
2. Does ASLR matter for local exploits?
3. UNIX permissions expose a large kernel attack surface
4. Shell access makes jailbreak development easier (ex. ASLR less effective)

Jailbreaks are a Reality!

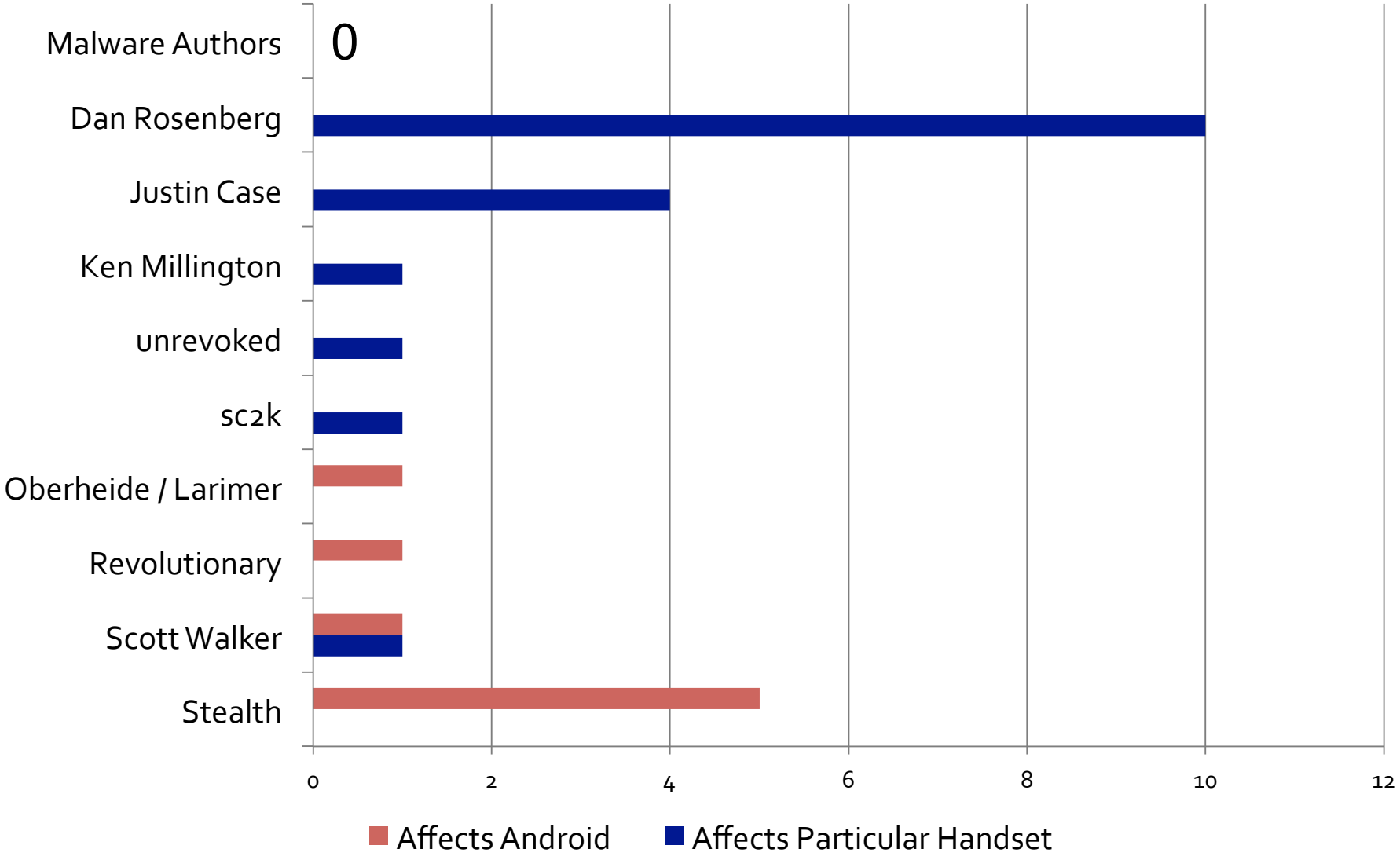
"My Gingerbreak works, but I wont release it before a couple of devices are in the wild so the issue is not fixed before it can become useful."

-- stealth (prior to releasing Gingerbreak)

- Must prevent malicious use of jailbreaks
  1. Mitigations slow down development
  2. Patch quickly to decrease potential revenue
  3. Discourage remote jailbreaks, enable local ones
  4. Disincentivize jailbreak community (Chromebook)

- Effective patching lets users self-segregate
  - Choose to be patched or choose run a jailbreak

Waiting for Handouts



Malware Devs are Choosy

Exploit Name	Last Affected Version	Abused?
Exploid	2.1	YES
RageAgainstTheCage	2.2.1	YES
Zimperlich	2.2.1	No
KillingInTheNameOf	2.2.2	No
Psneuter	2.2.2	No
GingerBreak	2.3.4	YES
zergRush**	2.3.5	No
Levitator	2.3.5	No
Mempodroid**	4.0.3	No (diff offset per device)
PERF_EVENTS	CURRENT!	Not yet!

Market Size

How does patching affect potential revenue?

Android Maximizes Potential Revenue

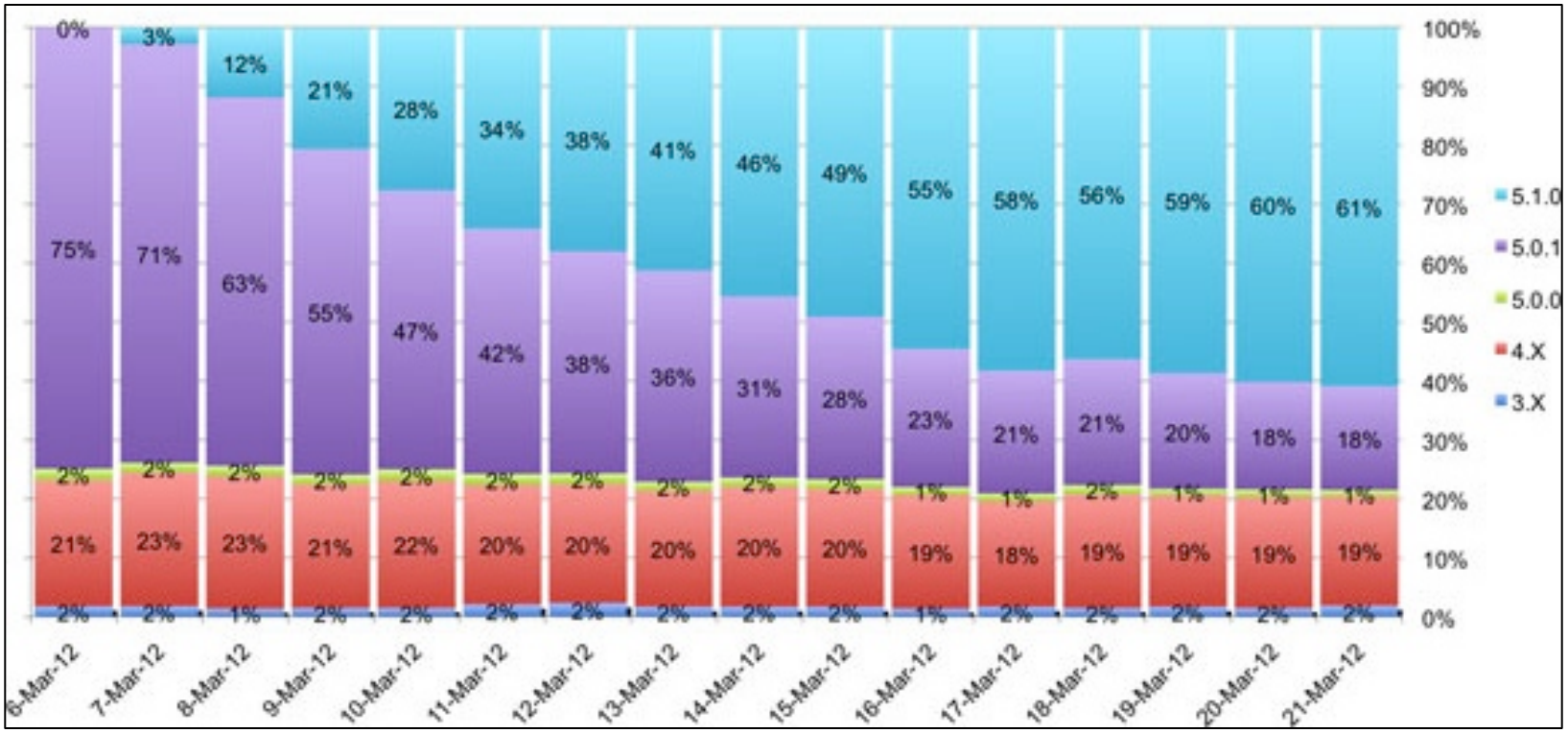
Platform	03/12/2012	4/18/2012	06/04/2012	06/03/2013
1.X Cupcake / Donut	1.2	1.0	0.9	0.1
2.1 Eclair	6.6	6.0	5.2	1.5
2.2 Froyo	25.3	23.1	19.1	3.2
2.3 Gingerbread	62.0	63.7	65.0	36.5
3.X Honeycomb	3.3	3.3	2.7	0.1
4.X ICS	1.6	2.9	7.1	25.6
4.1.x Jelly Bean				29.0
4.2.x Jelly Bean				4.0

Android Exploit	Time to Patch 50%
Exploid (2.1)	294 days
RageAgainstTheCage (2.2.1)	> 240 days

Android Jailbreak Equivalents

- Android Private Signing Keys
  - jSMSHider: http://goo.gl/vPzjg
  - Affects custom ROMs only
- Request Device Admin API Privs
  - DroidLive: http://goo.gl/c3EET
- Request SMS privileges
  - Almost 100% of non-jailbreak malware
  - SMS short codes, bank two-factor auth, etc

iOS Limits Potential Revenue



Vulnerability	Exploit	Patch Availability
Malformed CFF	Star (JailbreakMe 2.0)	10 days
T1 Font Int Overflow	Saffron (JailbreakMe 3.0)	9 days

http://david-smith.org/blog/2012/03/10/ios-5-dot-1-upgrade-stats/index.html

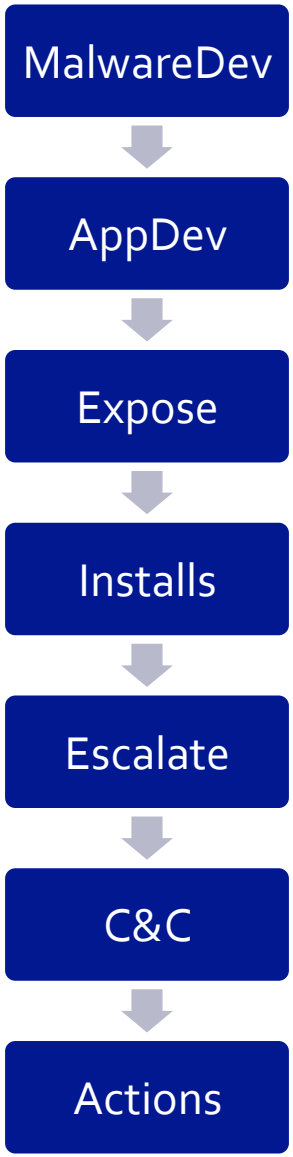
Privilege Escalation Takeaways

- Malware authors have no ability to write exploits
  - The only exploits abused are public jailbreak exploits
- Cost to exploit Android is significantly lower than iOS
  - App permissions required to jailbreak? NONE
  - Many available jailbreaks, easier to deliver to device
- Lack of Android patches create an opportunity
  - 60% of Android devices affected by a public jailbreak

If you can install an app, you can take over Android

Recap

Attacker Workflow Recap (iOS)



Learn Objective-C programming

Repackage app with an already public jailbreak (no runtime mods)

Register an LLC / steal a real ID, make app store account
Hope that a public jailbreak in your app evades detection

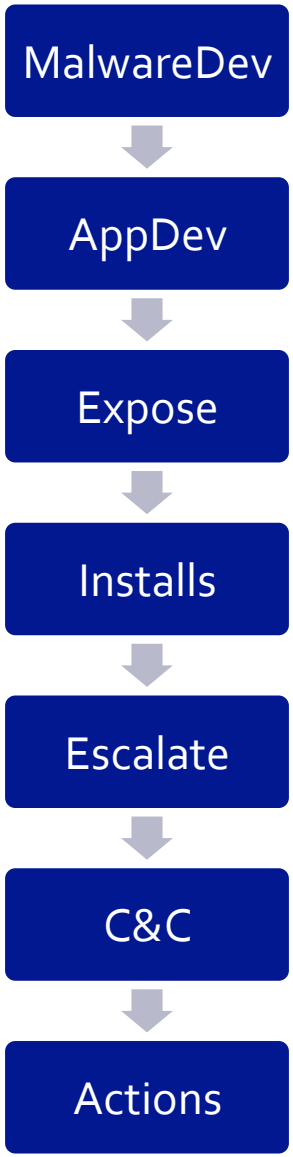
Hire a spammer to drive install of jailbreak app (can't be ad-hoc)

Run the jailbreak and payload embedded in your app

Call out to a server you control, dump data there

Sell data to broker and cash out

Attacker Workflow Recap (Android)



Learn C programming

Repackage app with malware you just wrote

Purchase stolen credit card, submit app to store (repeat as needed)

Hire a spammer to drive installs of innocuous app (can be adhoc)

Download and run jailbreak when available

Call out to a server you control, dump data there

Sell data to broker and cash out

Android Mitigation Outlook

- Chrome for Android
  - Makes browser exploits hard
  - Not an exploited vector now
  - No effect on current Android malware
- SEAndroid
  - Kills userspace jailbreaks, but not kernel!
  - Kernel exploits demonstrated on iOS
  - What handsets will use it?
- ASLR in Ice Cream Sandwich 4.x
  - Little to no effect on privilege escalations
  - Useful to make browser exploits difficult
  - Can't help 300+ million existing devices

Google is ahead of threats that don't exist yet, but far behind on ones that do

What Works?

- Require identification during App Store signup
  - Discrepancy in value between malware and legit devs
  - Ban real-world identities, not e-mail addresses
- Use policy to make app review more effective
  - Deny apps obfuscation, runtimes, self-updating
- Detect unique patterns of malicious app distribution
  - AppStore SEO, spam e-mails, SMS, advertisements…
- Patch quickly to deny profitability of jailbreaks
  - Disincentivize creation of them at all!
  - Further tighten Abilities, restrict ability to run new code

Predictions

- Malware continues to be App and Android-centric
  - Bouncer gained only a temporary upper hand
  - Inability to patch creates market opportunity
- Innovation will revolve around "Driving Installs"
  - Eg. NotCompatible via web, Zeus via sideloading, etc
- Lots of possible attacks ignored by real attackers
  - iOS will steer clear of similar attacks for now
  - Web is probable vector only when value high enough
- Attackers are resource-constrained and rational

Questions / Comments

dan@trailofbits.com

www.trailofbits.com

@TrailofBits

References

- Attacker Math 101, Dino Dai Zovi
  - www.trailofbits.com/research/#attackermath
- iOS Security Evaluation, Dino Dai Zovi
  - www.trailofbits.com/research/#ios-eval
- Exploit Intelligence Project, Dan Guido
  - www.trailofbits.com/research/#eip
- Lookout Security Mobile Threat Report
  - https://www.mylookout.com/mobile-threat-report
- Contagio Mini Dump, Mila
  - http://contagiominidump.blogspot.com/
- Dissecting Android Malware, Yajin Zhou, Xuxian Jiang
  - http://goo.gl/AOrCJ
- Androguard, Anthony Desnos
  - https://code.google.com/p/androguard/
- Dissecting the Android Bouncer, Jon Oberheide and Charlie Miller
  - http://goo.gl/BK82s