Mobile Exploit Intelligence Project

Dan Guido, Trail of Bits

Mike Arpaia, iSEC Partners

GFIRST, 08/23/2012

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, TEMPEST, …
- 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?

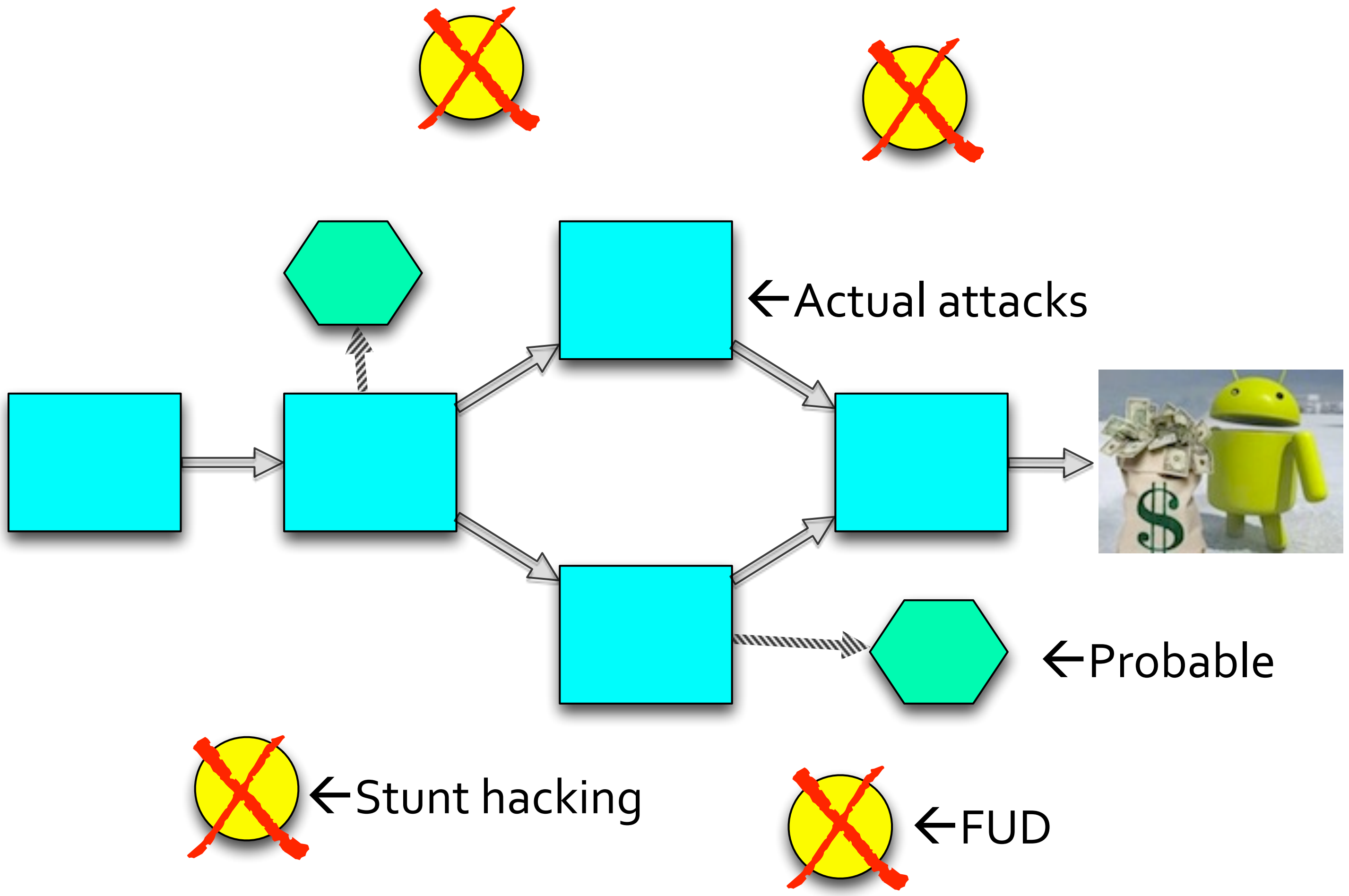
Millions of Mobile Attacks

1	3	1
Attack Vector	Exploits	Platform

* Android and iOS, 2011-2012

What are we doing wrong?

Your Defense Lacks Intelligence



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

How does it work?

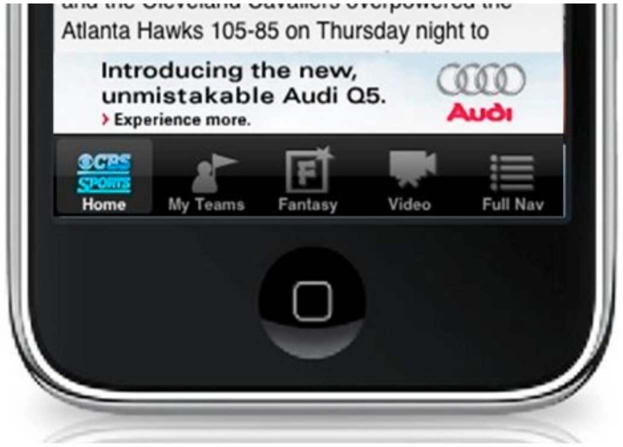
Mobile Malware – The Setup

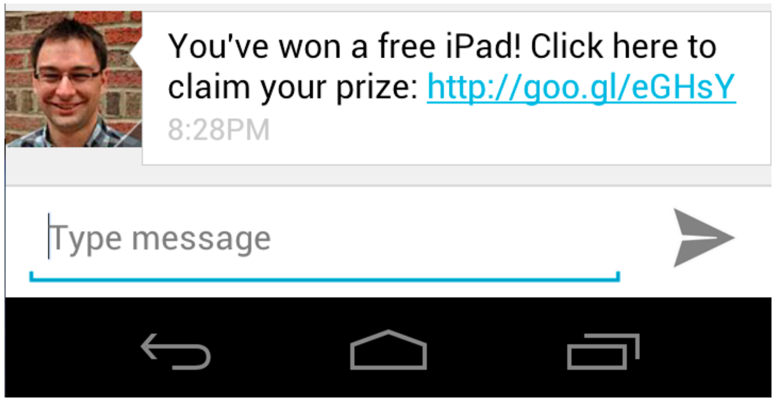
1. Develop malware

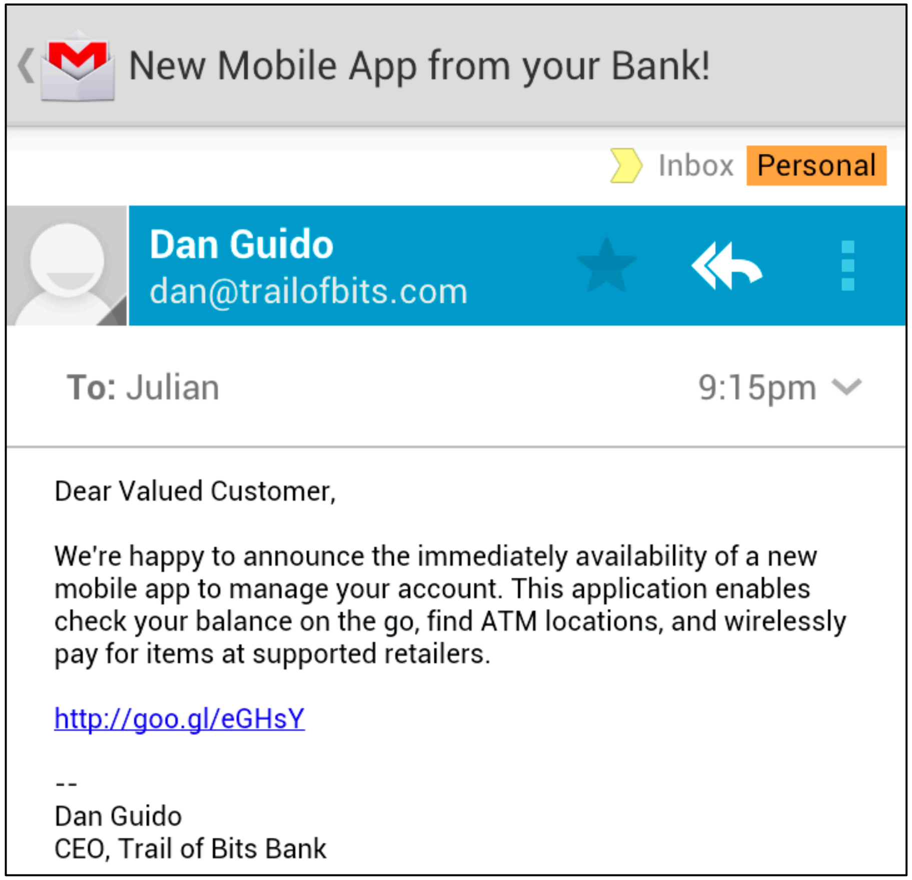
2. Add malware to many applications

3. Put malware online

Drive Installs


Atlanta Hawks 105-85 on Thursday night to


You've won a free iPad! Click here to claim your prize: http://goo.gl/eGHsY
8:28PM


New Mobile App from your Bank!

Dan Guido
dan@trailofbits.com

To: Julian 9:15pm

Dear Valued Customer,

We're happy to announce the immediately availability of a new mobile app to manage your account. This application enables check your balance on the go, find ATM locations, and wirelessly pay for items at supported retailers.

http://goo.gl/eGHsY

--
Dan Guido
CEO, Trail of Bits Bank

Mobile Malware – 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

Intrusion Kill Chains

- Systematic process that an intrusion must follow
  - Deficiency in one step will disrupt the process

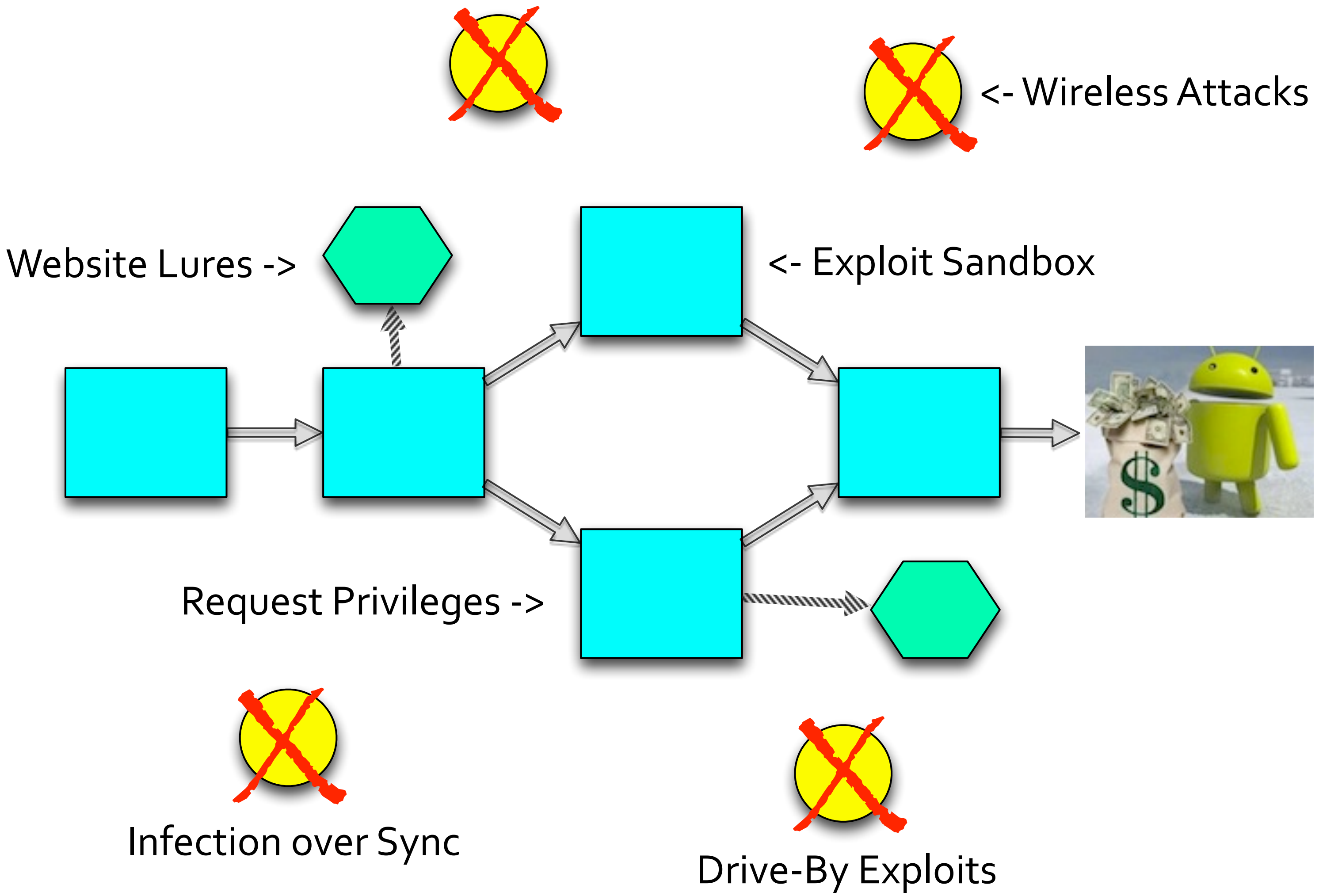
- Evolves response beyond point of compromise
  - Prevents myopic focus on vulnerabilities or malware
  - Identifies attacker reuse of tools and infrastructure

- Guides our analysis and implementation of defenses
  - Align defenses to specific processes an attacker takes
  - Force attackers to make difficult strategic adjustments

There’s Not Much a Spy Can Do

Hundreds of malwares	MalwareDev	Platform Security – Vendor Controlled
Thousands of modified apps	AppDev	App Store Policies – Vendor Controlled
Thousands of advertisements	Expose	Blacklists
Millions of installs	Installs	Platform Security – Vendor Controlled
Tens of escalation exploits	Escalate	Platform Security – Vendor Controlled
Hundreds of domains	C&C	Blacklists
Cyber Pompeii	Actions	Abuse Monitoring

Why Did This Chain Form?



Discrepancies

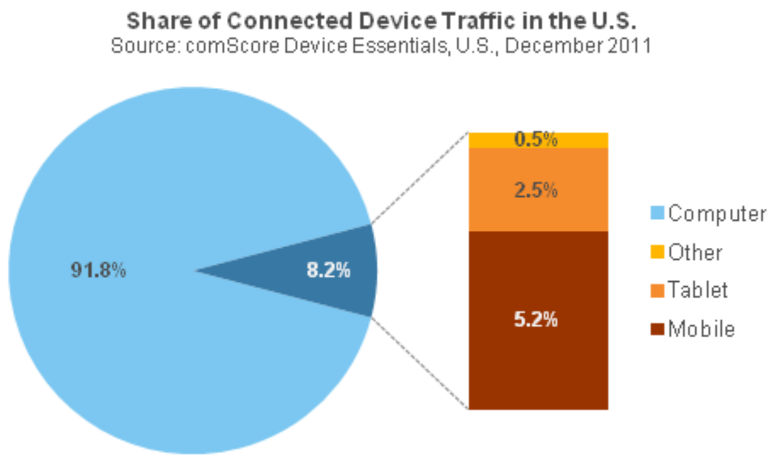
- Is the security industry lying to us?
  - Assumptions that mobile threat == desktop threat
  - Fascination with new attack vectors
  - Myopic focus on ease of attack and malware
- We have no idea how attackers actually work
  - Always more possibilities than probable attacks
  - Attacker economics are different on mobile
- Use economics and adversarial characterization!
  - Why don’t we / why won’t we see certain attacks?

Where are Mobile Drive-Bys?

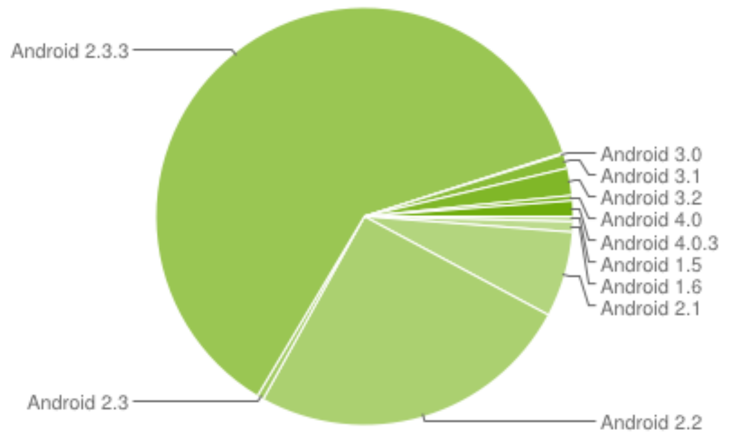
Mobile Town

Desktop City

Not Enough Mobile Targets

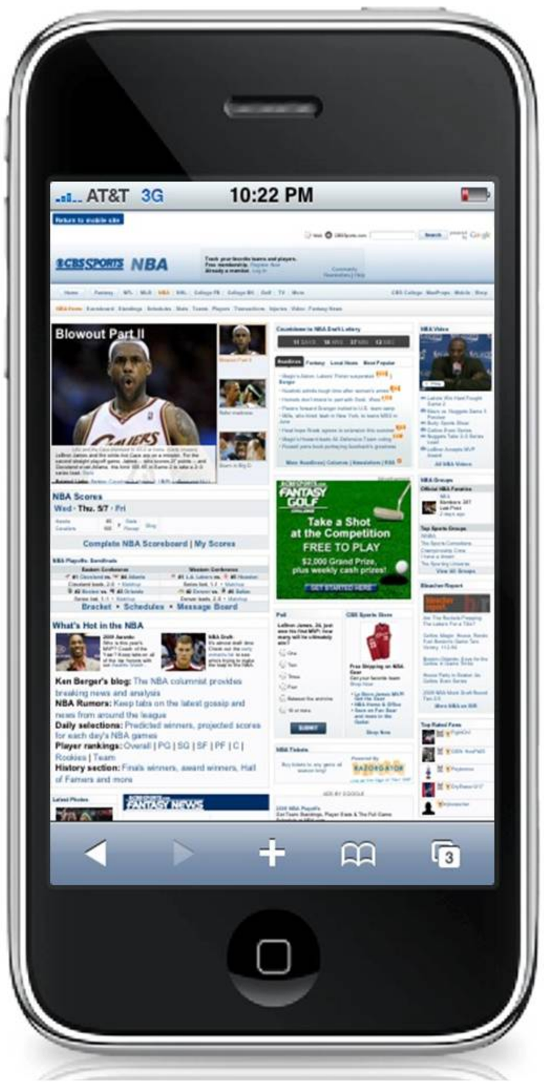


~8% of total web traffic
comes from mobile devices

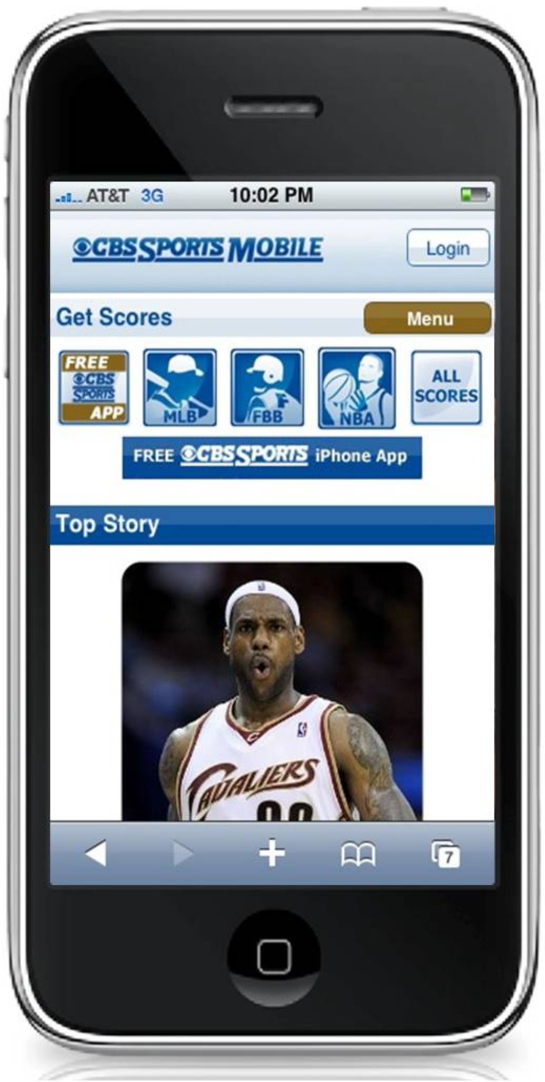


Breakdown by version / features
(+ varying rates of feature support)

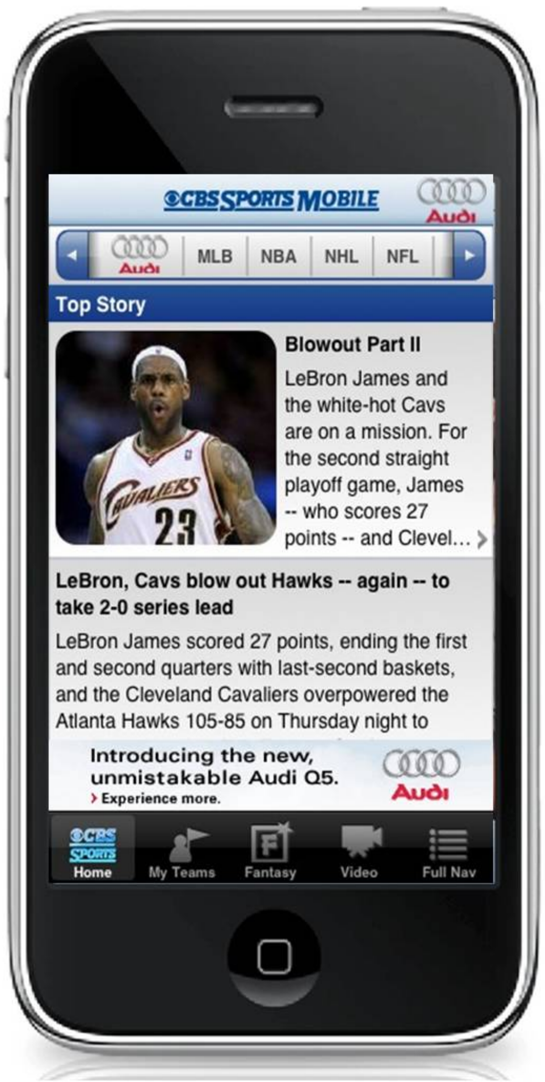
Lack of Ads Limits Targeting Potential



Normal Website



Mobile Website



Blowout Part II

LeBron James and the white-hot Cavs are on a mission. For the second straight playoff game, James -- who scores 27 points -- and Clevel…

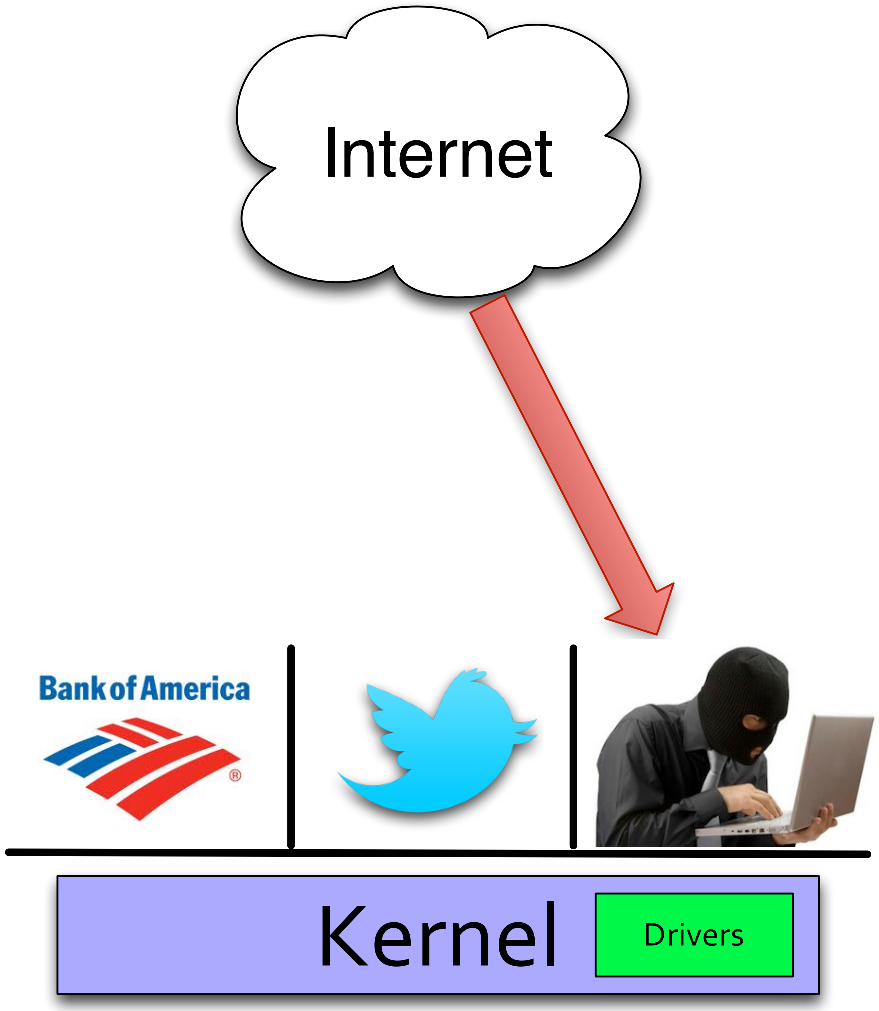
LeBron, Cavs blow out Hawks -- again -- to take 2-0 series lead

LeBron James scored 27 points, ending the first and second quarters with last-second baskets, and the Cleveland Cavaliers overpowered the Atlanta Hawks 105-85 on Thursday night to

Introducing the new, unmistakable Audi Q5.
› Experience more.
Audi

Mobile App

Mobile 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

Vendor App Stores

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!

Mobile Drive-by Takeaway

- 10-20x less potential targets than desktops
  - Not many mobile browsers, split between platforms
  - Mobile websites commonly won't have ads

- Increased costs to exploit relative to desktops
  - Harder to target due to feature disparities, lack of flash
  - Multiple exploits required for browser + jailbreak
  - However, anonymity comes easier

- Possible, but incentives are stacked against it
  - *Zero* identified cases in the data
  - Cost not likely to change but Potential Revenue could…

Sidenote: Why remotely attack individual Apps when even the browser isn't viable yet?

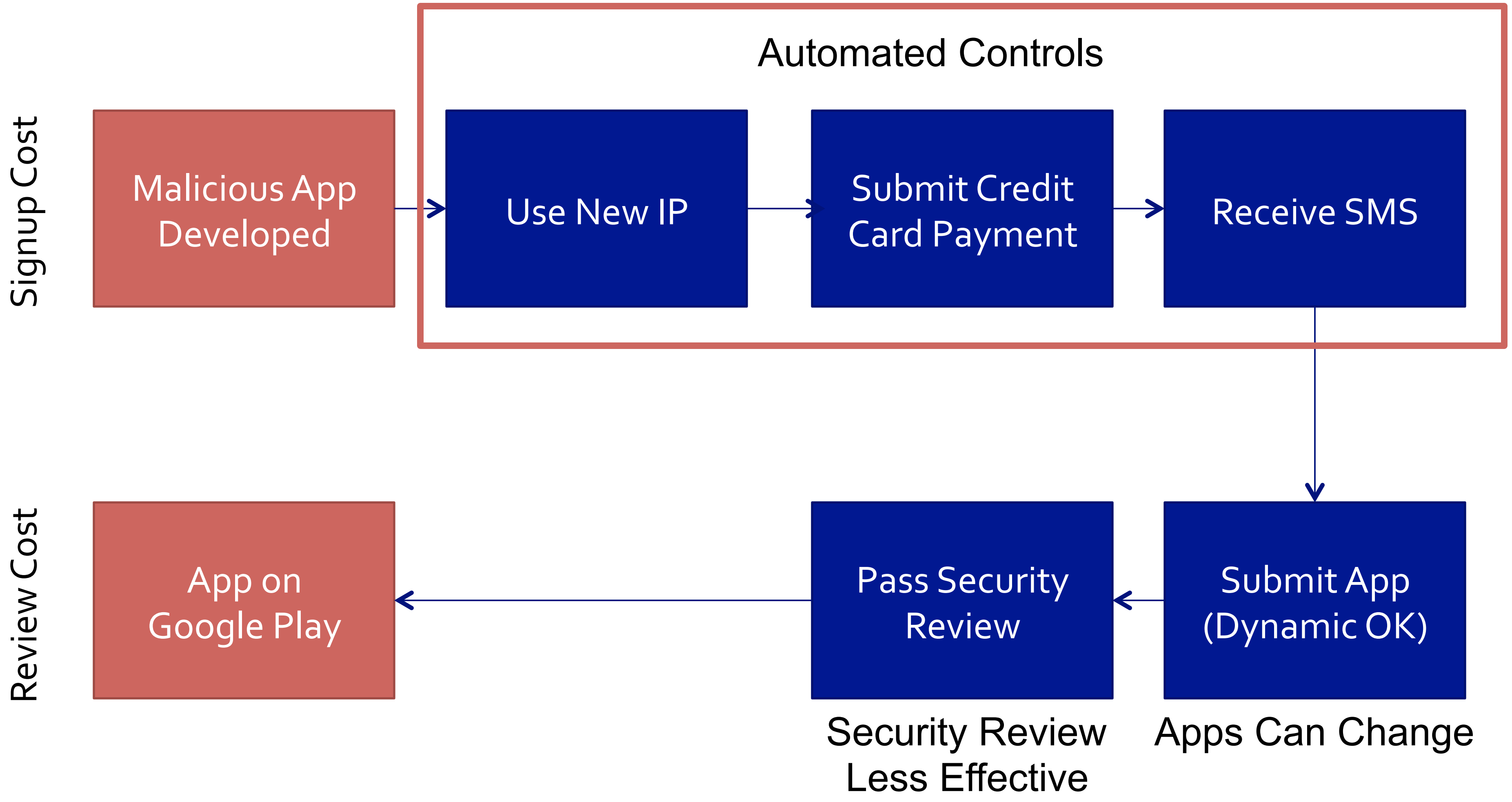
Scaling the Setup

1. Develop malware

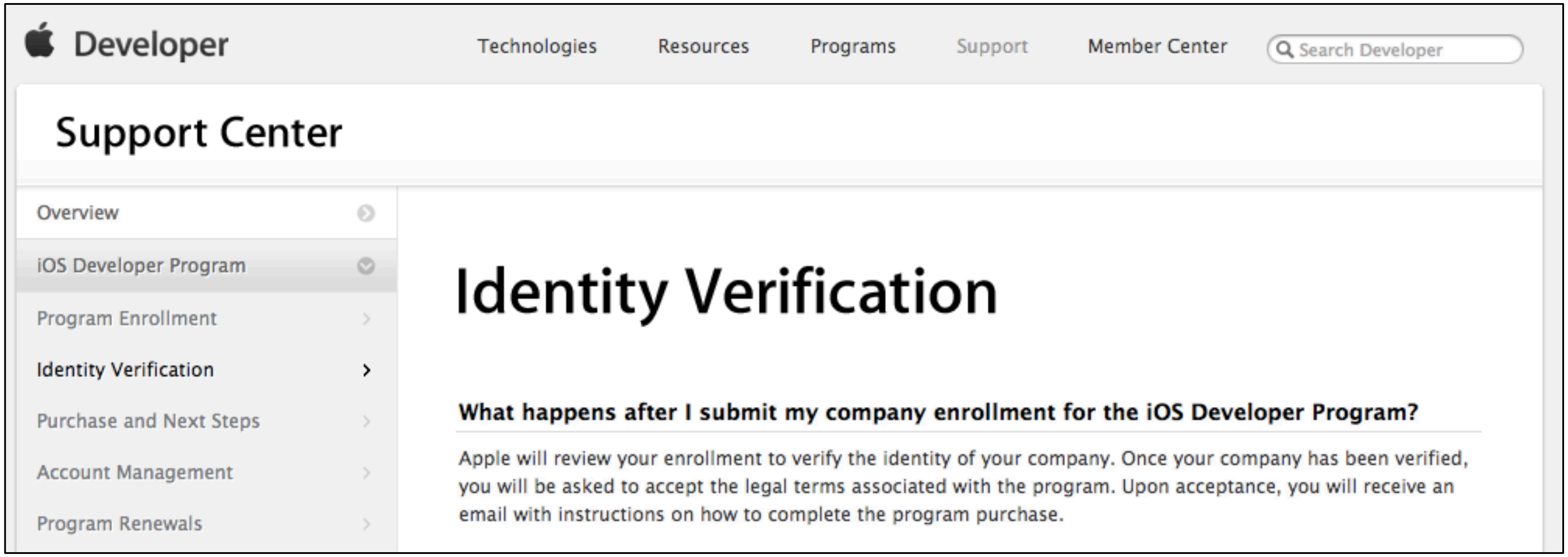
2. Add malware to many applications

3. Put malware online

Scaling Malicious App Submission



Think Different



Support Center

Identity Verification

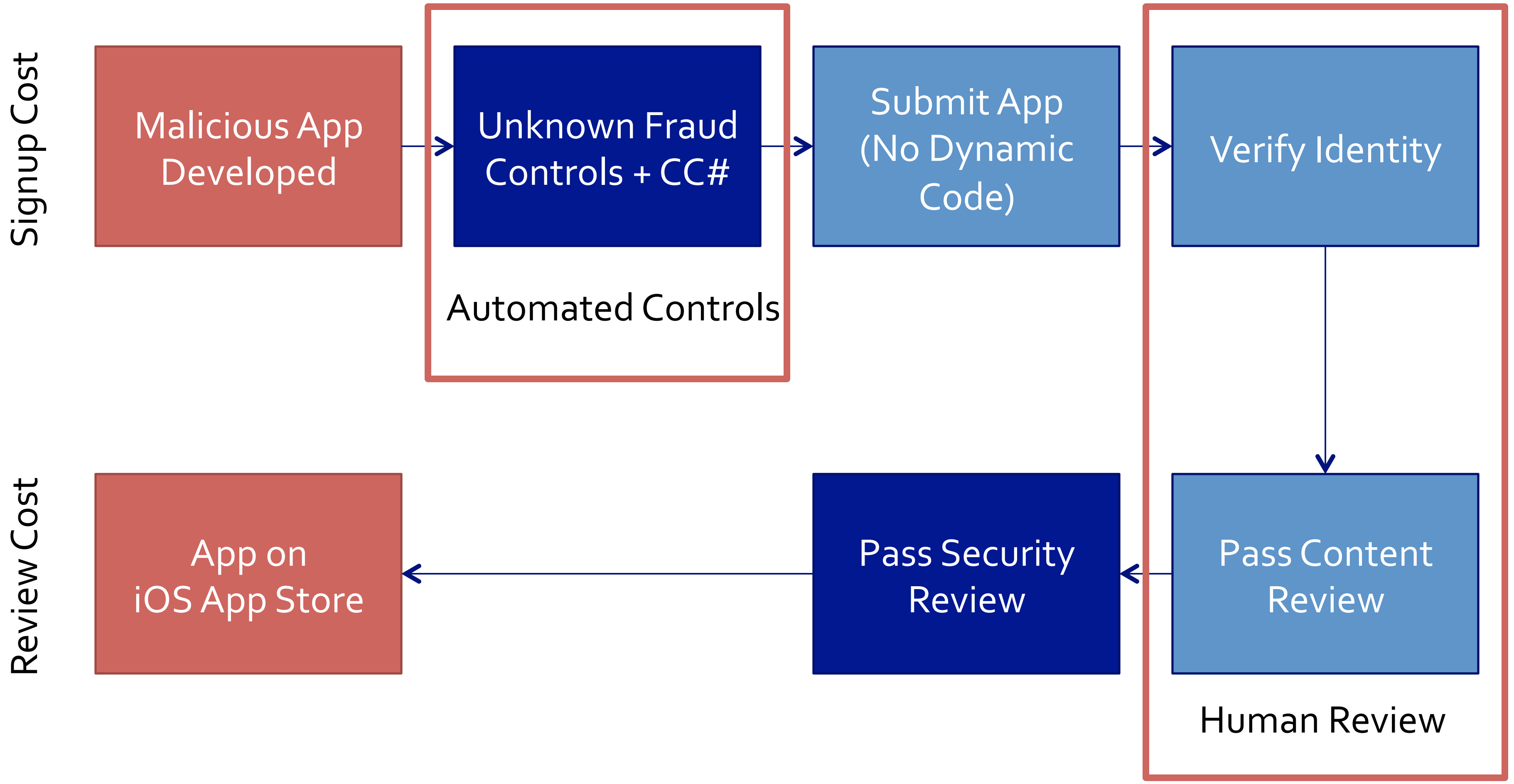
What happens after I submit my company enrollment for the iOS Developer Program?

Apple will review your enrollment to verify the identity of your company. Once your company has been verified, you will be asked to accept the legal terms associated with the program. Upon acceptance, you will receive an email with instructions on how to complete the program purchase.

- Automate new CC/SMS/IPs < Automate new LLCs
  - Forces malware authors to scale with humans

- Enforces accountability along with ban on dynamic code
  - More difficult to recover from bans

Scaling Malicious App Submission



Apple Enforces Accountability

	iOS App Store	Google Play
Sign Up	Fraud Controls	Fraud Controls
Identification	Drivers License Articles of Incorporation	IP/SMS/CC#
App Review	Unknown Analysis	Bouncer
Architecture	No runtime modification	Runtime modification

Malicious App Campaigns

0	30
Apple App Store	Google Marketplace

"Say what you will about police states, but they have very little crime."

Scaling the Heist

5. Access data outside the app sandbox

6. Send stolen data to a remote location

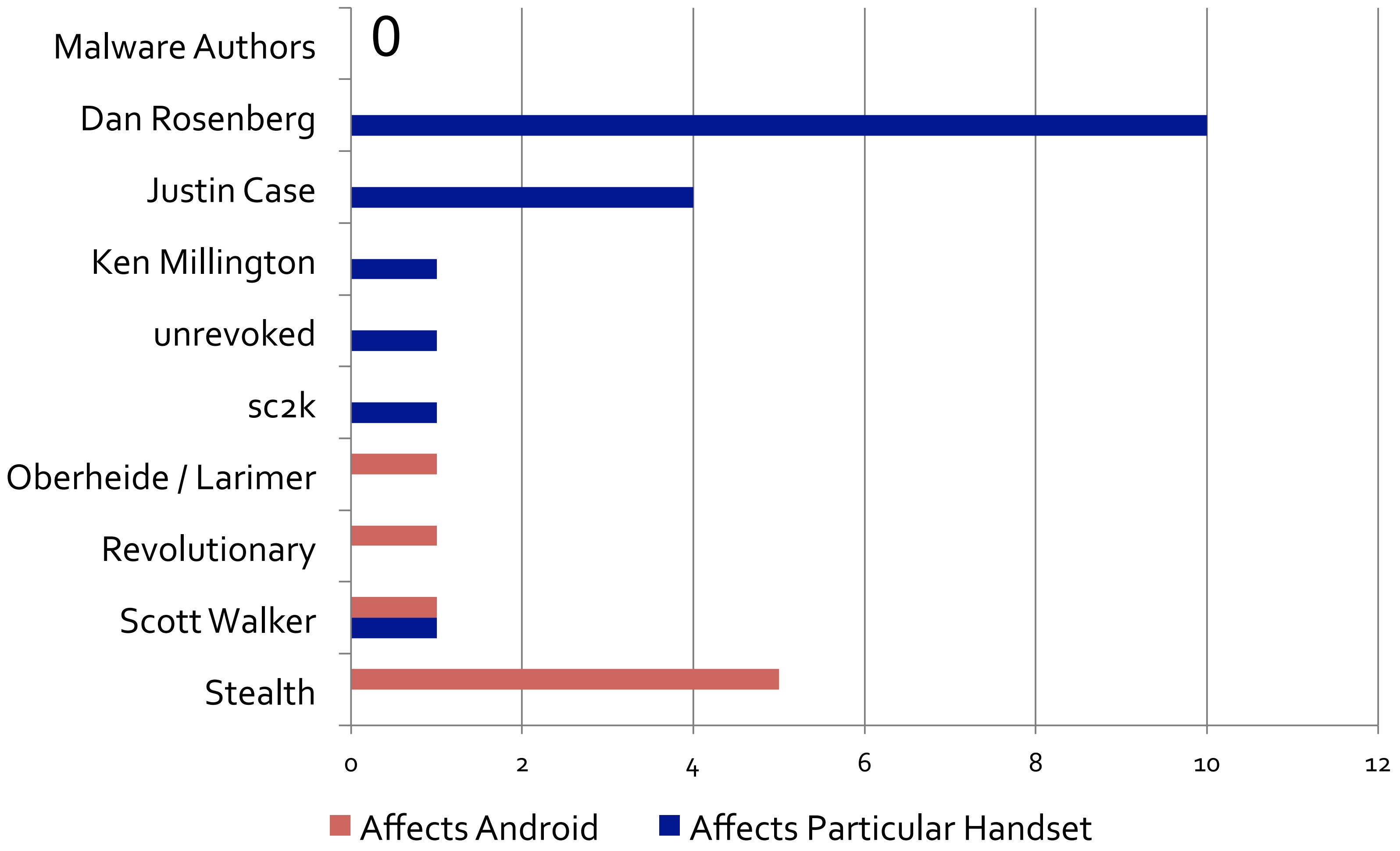
7. Abuse the data somehow to make money

Which Exploits Get Used?

Exploit Scenario	Cost of Attack	Value of Data	# of Targets
Universal Jailbreak	High?	High (all data)	High (all)
Request SMS	Free	High (2FA)	Medium (2FA users)
Handset Jailbreak	Limited Availability	High (all data)	Low
App-to-App	Limited Availability	Low (limited data)	Low

- Both platforms have active jailbreaker communities
  - Android: 26 jailbreaks from 10 different authors
  - iOS: 25 jailbreaks from ~4 main groups

Android Jailbreaks by Target



Universal Android Exploits

Exploit Name	Last Affected Version	Abused?
Exploid	2.1	Malware
RageAgainstTheCage	2.2.1	Malware
Zimperlich	2.2.1	No
KillingInTheNameOf	2.2.2	No
Psneuter	2.2.2	No
GingerBreak	2.3.4	Malware
zergRush	2.3.5	No (config per device)
Levitator	2.3.5	No (low # of devices?)
mempodroid	4.0.3	No (config per device)

What to do?

- Jailbreaks are a certainty after enough popularity

  "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)

- How we do prevent malicious use of jailbreaks?
  1. Slow jailbreak development by increasing costs
  2. Discourage app-accessible jailbreaks
  3. Decrease potential revenue by patching quickly

- Make less to react to, then react quickly
  - Probably some kind of Tao proverb that says this better

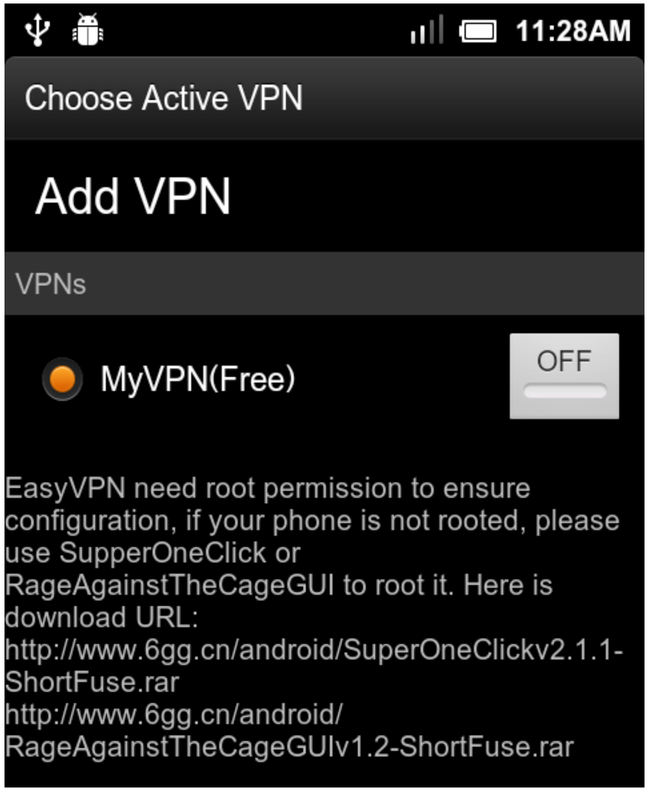
Factors Influencing JB Availability

Mitigation	iOS	Android
Code Injection	Code Signing	No-Execute
Randomization	Strong ASLR	ASLR*
Containment	Seatbelt	UNIX Permissions
Shell Available?	No	Yes

1. Code Signing is significantly stronger than NX (Partial vs Full ROP)
2. Does ASLR in Android 4.0.4+ matter if less than 7% are running it?
3. Android app permissions don't make privilege escalation harder
4. Shell access makes jailbreak development easier on Android

Android Jailbreak Equivalents

- Android Private Signing Keys
  - jSMSHider: http://goo.gl/vPzjg
  - Affects custom ROMs only
- Have the user do it (no joke) ---------->
  - Lena: http://goo.gl/eiTBA
- Request Device Admin API Privs
  - DroidLive: http://goo.gl/c3EET
- Request SMS privileges
  - Almost 100% of non-privesc malware
- They're less effective (user interaction), less used, but still work



EasyVPN need root permission to ensure configuration, if your phone is not rooted, please use SupperOneClick or RageAgainstTheCageGUI to root it. Here is download URL:
http://www.6gg.cn/android/SuperOneClickv2.1.1-ShortFuse.rar
http://www.6gg.cn/android/RageAgainstTheCageGUIv1.2-ShortFuse.rar

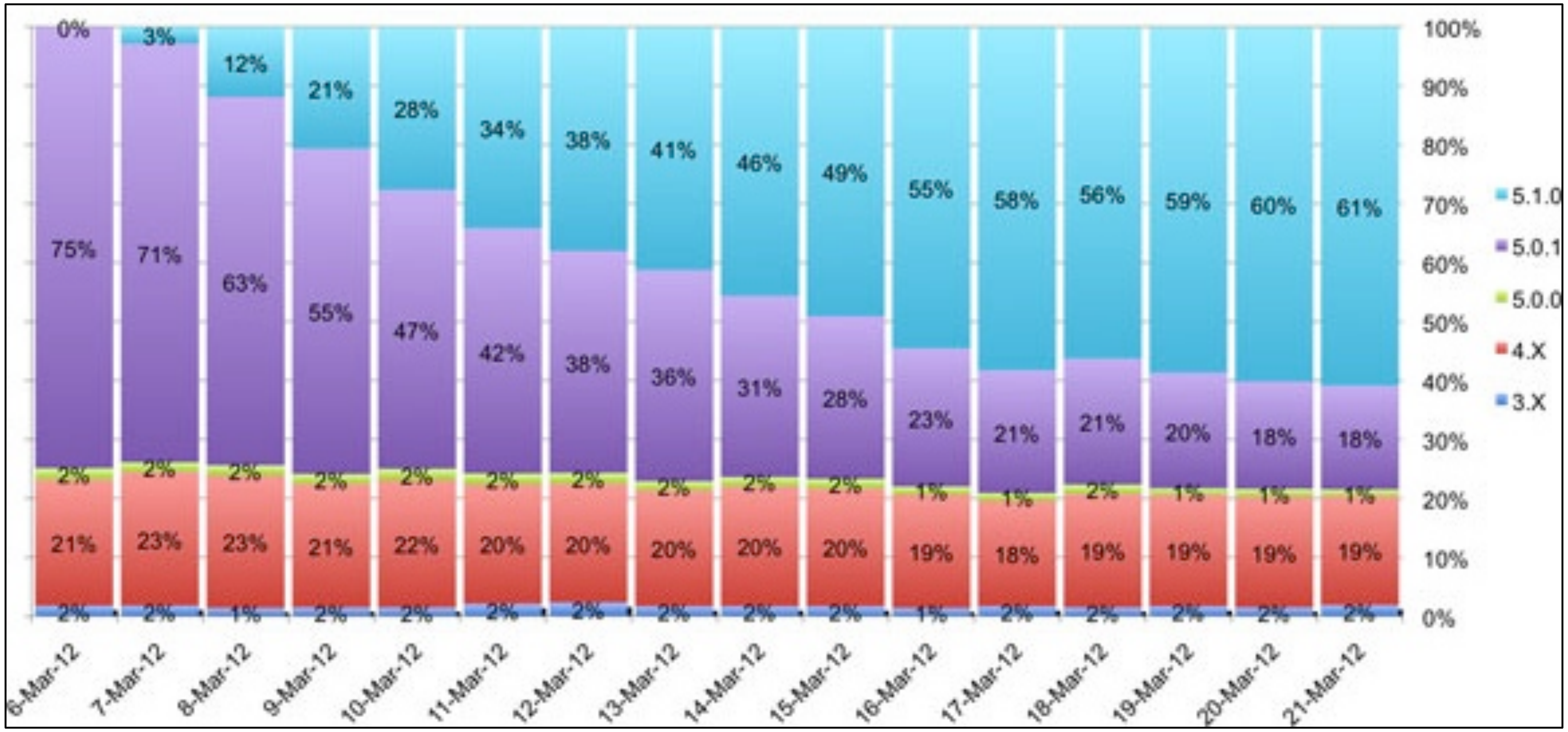
Android Maximizes Potential Revenue

Platform	Codename	03/12/2012	4/18/2012	06/04/2012
1.x	Cupcake / Donut	1.2%	1.0%	0.9%
2.1	Eclair	6.6%	6.0%	5.2%
2.2	Froyo	25.3%	23.1%	19.1%
2.3	Gingerbread	62.0%	63.7%	65.0%
3.x	Honeycomb	3.3%	3.3%	2.7%
4.x	Ice Cream Sandwich	1.6%	2.9%	7.1%

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

http://blog.mylookout.com/blog/2011/08/04/inside-the-android-security-patch-lifecycle/

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

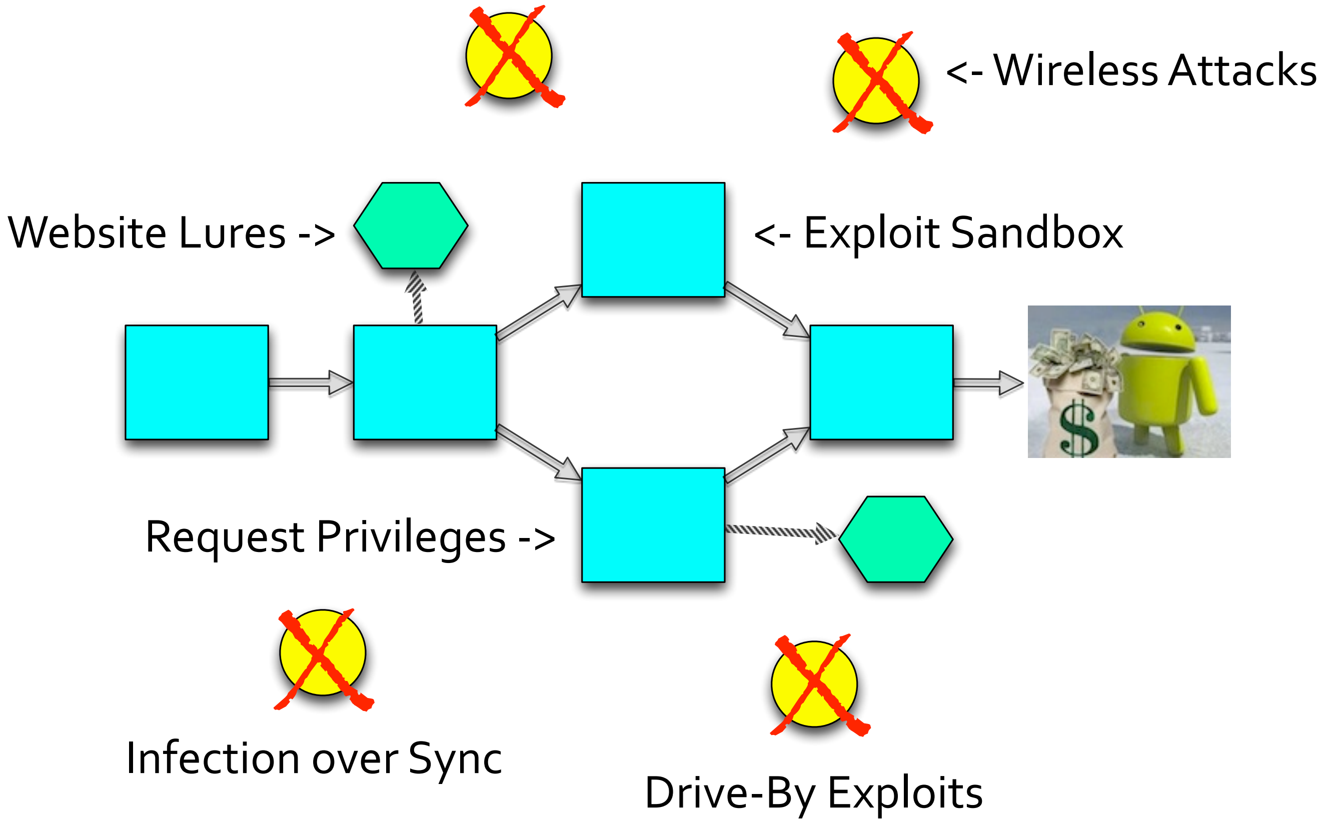
http://david-smith.org/blog/2012/03/10/ios-5-dot-1-upgrade-stats/index.html

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 sandbox is weak against privilege escalation
  - Platform has many alternate escalation scenarios
  - Implemented mitigations are weaker than on iOS
- Android patches have little effect on problem
  - Google has no ability to force carriers / OEMs to react
  - Even if they could, it's too easy to write new exploits

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

Where this leads us



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

Mobile Malware Predictions

- Malware continues to be App and Android-centric
  - "The Setup" is getting harder, but not by enough
    - It's still worth it to get malware into Google Play
  - "The Heist" scales extremely well on Android
    - Not likely to change any time soon
  - Innovation will revolve around Driving Installs
    - Ex. NotCompatible only differs in how it drives installs

- Upcoming Android mitigations incorrectly focused
  - Bouncer, Chrome, ASLR have limited impact
  - Changes in 4.0 / 4.1 don't significantly affect problem

Mobile Malware Predictions

- Browser, NFC, Ads (incl. mobile) are not as attractive
  - Higher costs than app-centric strategy
  - # of targets still too low
  - Lack of established process impedes growth

- iOS will steer clear of similar attacks for now
  - Real-world verification trumps all the technical attacks
  - Mitigations slow jailbreaks, quick patches reduce value

- Attackers are resource-constrained and rational

App Development Strategies

- Not all keychains are created equal
  - Android only stores keys. No keygen, no data storage.
    - Try not to shoot yourself in the foot!
  - Jailbreak means exposure of Android keystore
    - iOS DP API is HW-backed, significantly limits exposure

- Limit accessible data and implement a circuit breaker
  - Apps shouldn't request an entire DB of content
    - Alert / modify access after a threshold – circuit breaker
  - Determine accessible data by context
    - Why is your mobile device downloading AutoCAD files?

* http://www.trailofbits.com/resources/#ios-eval

Enterprise BYOD Strategies

- Mobile groupware must follow app security strategy
  - Limit accessible data, implement a circuit breaker
  - Ask your vendor these questions!

- Assume that BYOD devices are compromised
  - Less likely on iOS, a certainty on Android
  - Existing jailbreak detection is fallible
    - Malicious attackers aren't connecting to Cydia

- If Android users can install their own apps, they will be compromised by accident
  - Restrict access to internal App Catalogue if possible

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