

BUSTICATI PRODUCTIONS PRESENTS
DISSECTING THE ANDROID BOUNCER



STARRING

DR. OBERHEIDE and DR. MILLER



THE FOLLOWING **PREVIEW** HAS BEEN APPROVED FOR
ALL AUDIENCES
BY THE MOTION PICTURE ASSOCIATION OF AMERICA

THE FILM ADVERTISED HAS BEEN RATED



For information on film ratings,
go to www.filmratings.com

PINATA TIME!



- CANDY!
- SNACKS!
- BEER!



THE FOLLOWING **PREVIEW** HAS BEEN APPROVED FOR
RESTRICTED AUDIENCES ONLY
BY THE MOTION PICTURE ASSOCIATION OF AMERICA, INC.



www.filmratings.com

www.mpa.org

BACK IN THE GOOD OL' DAYS



No.	Time	Source	Destination	Protocol	Length	Info
4	0.154282	74.125.95.102	192.168.1.100	TCP	66	http > 48186 [ACK] Seq=1 Ack=715 Win=1002 Len=
5	0.452701	74.125.95.102	192.168.1.100	TCP	1484	[TCP segment of a reassembled PDU]
6	0.452713	192.168.1.100	74.125.95.102	TCP	66	48186 > http [ACK] Seq=715 Ack=1419 Win=685 Le
7	0.453030	74.125.95.102	192.168.1.100	TCP	1419	[TCP segment of a reassembled PDU]
8	0.453039	192.168.1.100	74.125.95.102	TCP	66	48186 > http [ACK] Seq=715 Ack=2772 Win=685 Le
9	0.454084	74.125.95.102	192.168.1.100	HTTP	990	HTTP/1.1 200 OK (application/binary)
10	0.454095	192.168.1.100	74.125.95.102	TCP	66	48186 > http [ACK] Seq=715 Ack=3696 Win=685 Le
11	1.681493	192.168.1.100	74.125.95.102	TCP	284	[TCP segment of a reassembled PDU]
12	1.727197	74.125.95.102	192.168.1.100	TCP	66	http > 48186 [ACK] Seq=3696 Ack=933 Win=1002 L

Frame 9: 990 bytes on wire (7920 bits), 990 bytes captured (7920 bits)

Ethernet II, Src: 2wire_61:10:b9 (00:25:3c:61:10:b9), Dst: Intel_90:e5:4a (00:19:d1:90:e5:4a)

Internet Protocol Version 4, Src: 74.125.95.102 (74.125.95.102), Dst: 192.168.1.100 (192.168.1.100)

Transmission Control Protocol, Src Port: http (80), Dst Port: 48186 (48186), Seq: 2772, Ack: 715, Len: 924

[3 Reassembled TCP Segments (3695 bytes): #5(1418), #7(1353), #9(924)]

0120	47 53 45 0d 0a 0d 0a 1f 8b 08 00 00 00 00 00	GSE....
0130	00 ed 5a 5d 6c 1c 57 15 ce d8 e4 a7 93 a4 8d 1d	..Z]l.w.
0140	42 9b 94 9f 1b 13 b5 d4 d8 e3 99 fd 5f 0b 21 6f	B....._!o
0150	1c 27 71 13 c7 ee ee 26 69 9f a2 bb 33 d7 bb 63	.'q...& i...3..c
0160	cf ce dd cc 8f 37 eb 27 ab 02 09 45 e2 05 09 097.'...E....
0170	04 0f a8 15 3c 20 44 51 25 84 08 12 2a 96 10 16< DQ %...*...
0180	bc 23 04 0f a0 3e f1 00 52 79 a2 42 20 f1 9d 3b	.#...>.. Ry.B ..;
0190	b3 5e db 75 7e da 54 42 42 89 25 ef 7a e6 ce fd	.^..u~.TB B.%.z...

The Android Market app used to primarily use HTTP as a transport!

BACK IN THE GOOD OL' DAYS



```
1 {
  1: "-1376480270119955654"
  2: "Liquor Run Mobile"
  3: 1
  4: "Tallgrass Labs, LLC"
  5: "2.0.5"
  7: "4.1950920245398775"
  8: 815
  12 {
    13: "Liquor Run 2.0 is here! We find the closest beer, liquor, & wine stores and remind you of area closing times. You make the Liquor Run! \n\nGet turn-by-turn directions to over 31,000 US liquor stores, browse 1000+ drink recipes, discover how many calories are in your favorite beer. \n\nBest of all, never miss your last call again!"
    14: 0
    15: "android.permission.INTERNET"
    15: "android.permission.ACCESS_COARSE_LOCATION"
    15: "android.permission.ACCESS_FINE_LOCATION"
    16: 322424
    17: "com.kansassoftware.liquorrun"
    18: "Shopping"
    19: 0
    20: "support@liquorrunmobile.com"
    23: "50,000-250,000"
    27: "http://www.liquorrunmobile.com"
```

So you could MITM the protobuf, inject your app into search results, trick people into installing malicious apps, etc

A NEW APP STORE?



Can't do as much nowadays, but still can play some tricks...

Fire up your Google Play app if you're on the wifi!



Google play



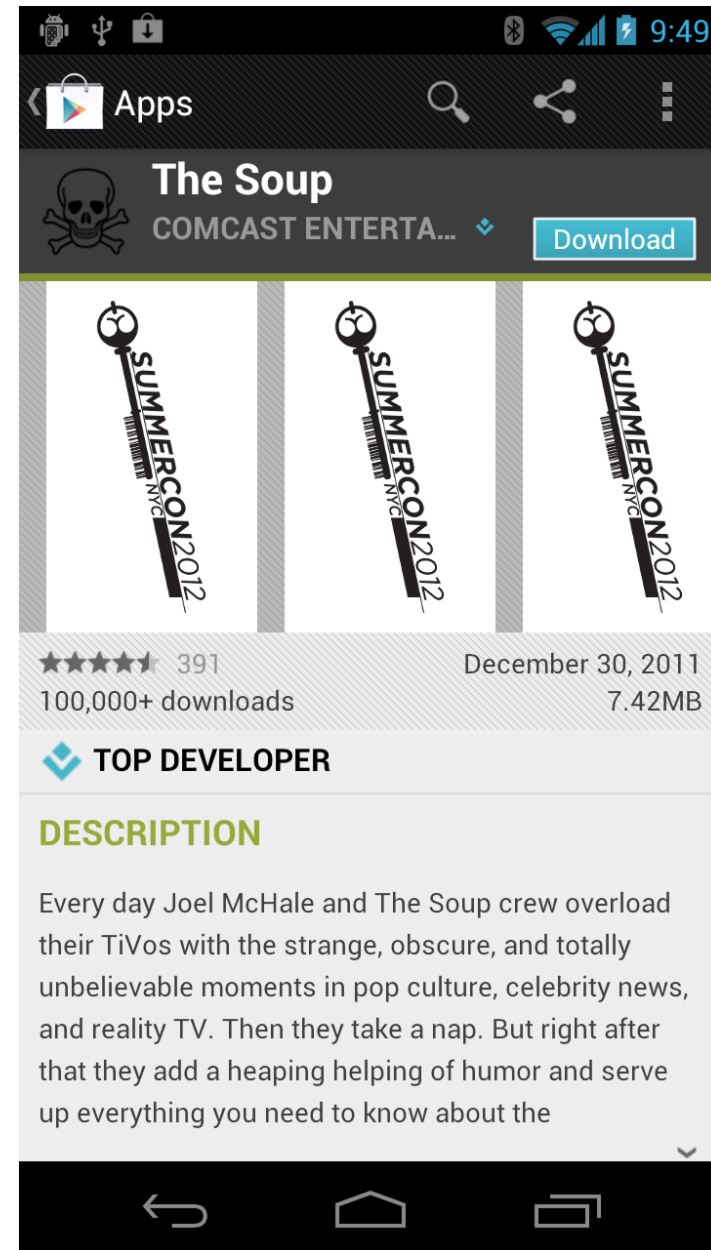
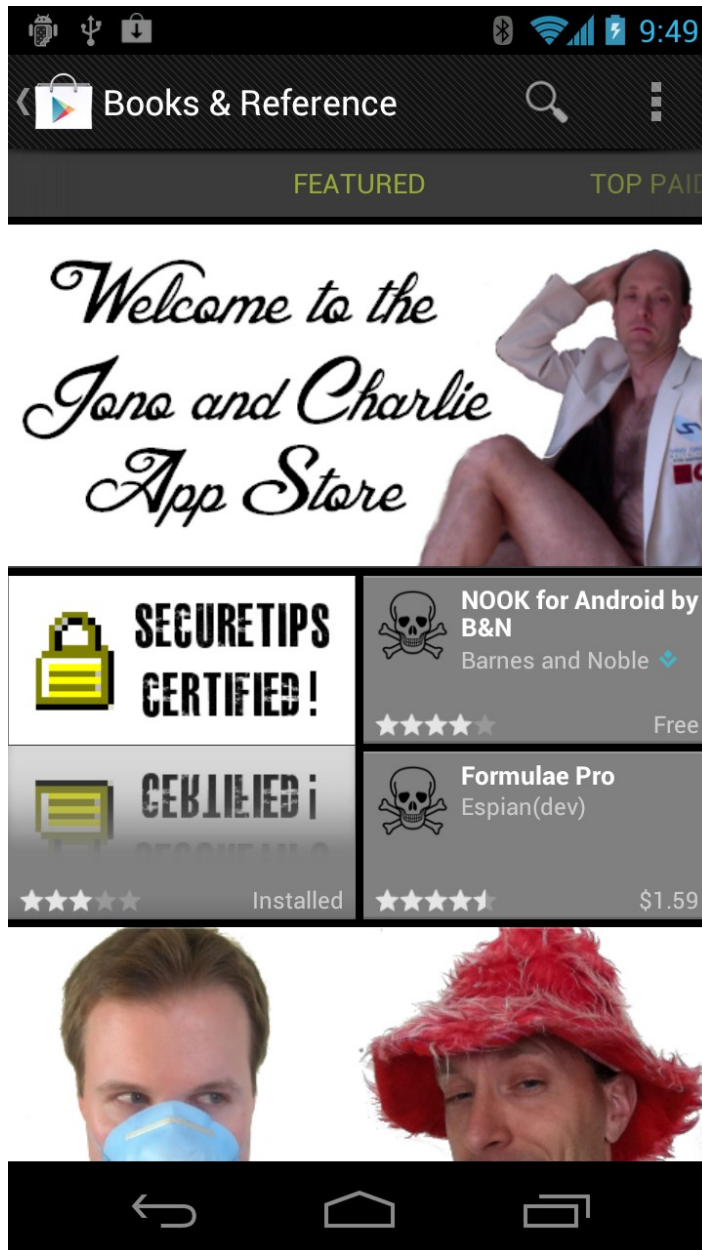
A NEW APP STORE!



*Welcome to the
Jon and Charlie
App Store*



JONO AND CHARLIE APP STORE

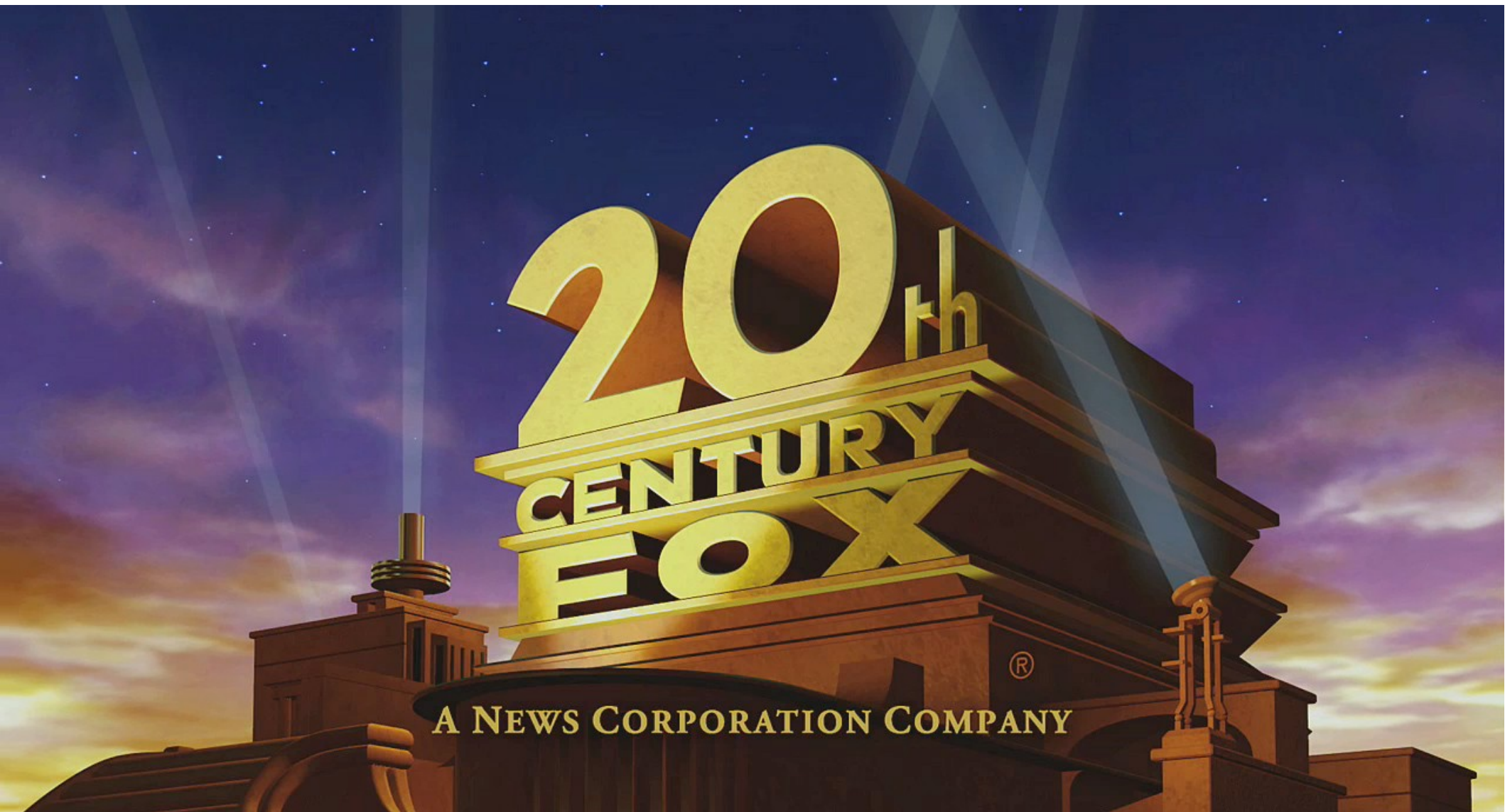


COMING SOON...



Coming soon to a
GitHub repository near you?

FEATURE PRESENTATION



STARRING...



Google Android's Bouncer



- **Diagnosis**
 - Intro to Bouncer and Google Play
- Exploratory surgery
 - Fingerprinting Bouncer and its environment
- Open surgery
 - Abusing Bouncer in all sorts of fun ways
- Suture and close
 - How Google can fix up Bouncer

ANDROID BOUNCER



Android and Security, Feb 2, 2012

*Today we're revealing a service we've developed, codenamed Bouncer, which provides **automated scanning of Android Market** for potentially malicious software without disrupting the user experience of Android Market or requiring developers to go through an application approval process.*

*The service **performs a set of analyses** on new applications, applications already in Android Market, and developer accounts. Here's how it works: once an application is uploaded, the service immediately starts **analyzing it for known malware, spyware and trojans**. It also looks for **behaviors** that indicate an application might be misbehaving, and compares it against previously analyzed apps to detect possible red flags. We actually **run every application** on Google's cloud infrastructure and **simulate how it will run on an Android device** to look for hidden, malicious behavior.*



- Bouncer is easily bypassed
 - No surprise there
 - Google is trying to solve a very difficult problem
- We'll show a bunch of ways
 - System, network, framework, timing, etc
- Story of how we analyzed Bouncer
 - Full of mystery and intrigue!
 - Also, pinatas and beer!



- How do we go about dissecting Bouncer?
- How would we create such a system?
- We had lots of unanswered questions:
 - Does Bouncer use static/dynamic analysis?
 - When does Bouncer analyze the app? Are all apps analyzed?
 - How do we get Market accounts to start figuring this out?
 - Network access: open, filtered, emulated, unrestricted?
 - Environment: what's the system execution environment look like?
 - Timing: how long does our app run? Accelerated clock?
 - Input: Artificial input to the app? Program state exploration?
 - Any triggers, vulnerable services, etc?

FIRST THINGS FIRST



We need some Play accounts...

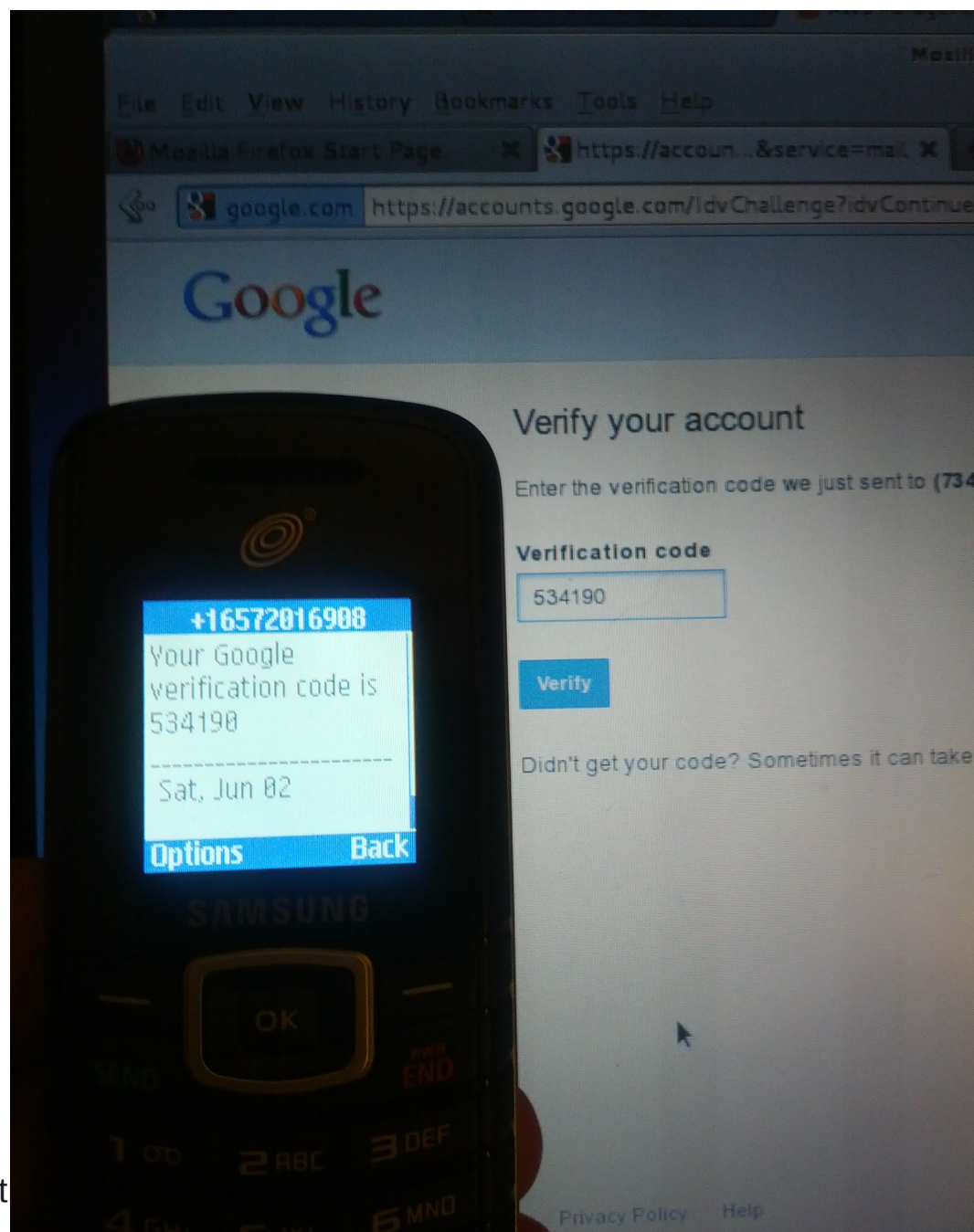
WHAT YOU NEED



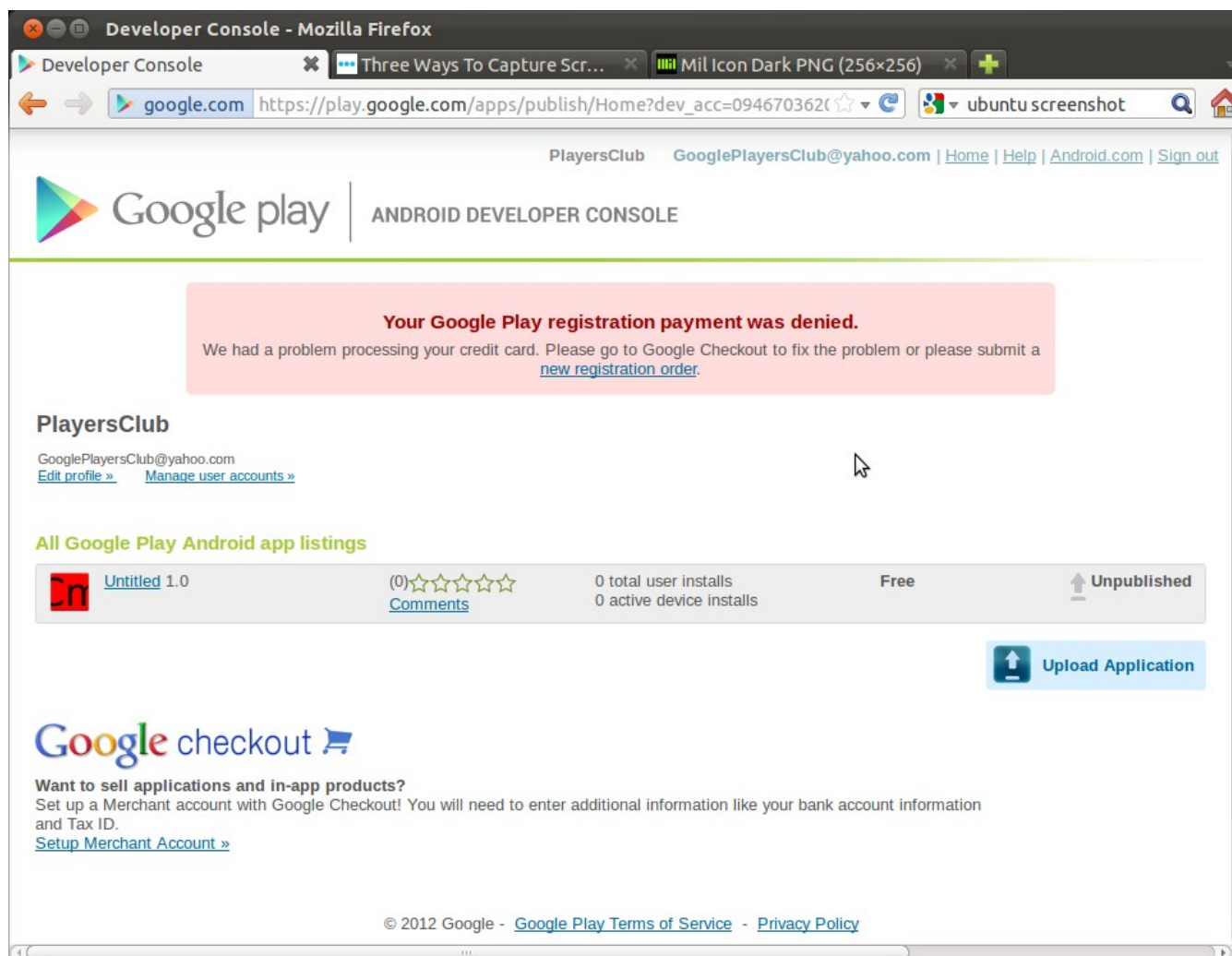
- Money
- Prepaid phones
- Prepaid CCs
- EC2 micros



BURNERS FOR GMAIL



PAYMENT LOOPHOLE



We can submit apps without paying!

HOW DO WE START?



- How do we start?
- Submit a simple app that phones home to our C&C server
- See what happens?



- Hippocratic Oath forbids us from pushing malware onto innocent bystanders
 - Put warnings in the description
 - Only make available to impossible hardware
 - Make the app not interesting
 - ...ugh...
- Any other way???

SUBMISSION STEP 1



Upload new APK

Required: Select your application's APK

Optional: Add an expansion file
If your app exceeds the 50MB APK limit, you can add expansion files. [Learn more](#)

Upload your APK...

SUBMISSION STEP 2



Edit Application

Product details | APK files | [Publish](#) | [Save](#)

Upload assets

Screenshots at least 2	Add a screenshot: <input type="text"/> Browse... Upload	Screenshots: 320 x 480, 480 x 800, 480 x 854, 1280 x 720, 1280 x 800 24 bit PNG or JPEG (no alpha) Full bleed, no border in art You may upload screenshots in landscape orientation. The thumbnails will appear to be rotated, but the actual images and their orientations will be preserved.
High Resolution Application Icon [Learn More]	Add a hi-res application icon: <input type="text"/> Browse... Upload	High Resolution Application Icon: 512 x 512 32 bit PNG or JPEG Maximum: 1024 KB
Promotional Graphic optional	Add a promotional graphic: <input type="text"/> Browse... Upload	Promo Graphic: 180w x 120h 24 bit PNG or JPEG (no alpha) No border in art
Feature Graphic optional [Learn More]	Add a feature graphic: <input type="text"/> Browse... Upload	Feature Graphic: 1024 x 500 24 bit PNG or JPEG (no alpha) Will be downsized to mini or micro
Promotional Video optional	Add a promotional video link: <input type="text" value="http://"/>	Promotional Video: or YouTube URL
Privacy Policy [Learn more]	Add a privacy policy link: <input type="text" value="http://"/> <input type="checkbox"/> Not submitting a privacy policy URL at this time	
Marketing Opt-Out	<input checked="" type="checkbox"/> Do not promote my application except in Google Play and in any Google-owned online or mobile properties. I understand that any changes to this preference may take sixty days to take effect.	

Listing details

Fill in application metadata...



- Press “Save” button...

```
74.125.19.84 - - [08/Apr/2012:23:33:05 -0400]  
"GET /?id=9774d56d682e549c HTTP/1.1" 200 5 "-"  
"Apache-HttpClient/UNAVAILABLE (java 1.4)" "-"
```

- Wait, what was that?!?
- Looks like Bouncer ran our app!
 - Before it was actually published to the market!



- Diagnosis
 - Intro to Bouncer and Google Play
- **Exploratory surgery**
 - **Fingerprinting Bouncer and its environment**
- Open surgery
 - Abusing Bouncer in all sorts of fun ways
- Suture and close
 - How Google can fix up Bouncer



- Bouncer in a nutshell
 - Dynamic runtime analysis of app
 - Emulated Android environment
 - Runs for 5 minutes
 - On Google's infrastructure
 - Allows external network access
- If we can fingerprint the environment
 - Pretend to be benign when run on Bouncer
 - Execute malicious activity when run on real devices



- Underlying system
 - Linux, QEMU emulator, system properties, etc
- Android Framework
 - Sensors: camera, accelerometer, gps, etc
 - Data sources: address book, sms, photos, files, etc
- Environment and behaviors
 - IP address, timing attacks, input automation, etc



- Lots of low-hanging fruit
 - /proc/cpuinfo: goldfish
 - getprop attributes: ro.kernel.qemu
 - Obvious QEMU stuff: /sys/qemu_trace, etc
 - Many many more...
- Once the easy stuff is fixed
 - Fingerprinting QEMU based on emulation discrepancies
 - http://static.usenix.org/event/woot09/tech/full_papers/paleari.pdf
 - Could fingerprint the exact QEMU version (and exploit ;-)

SYSTEM VITAL SIGNS



	Galaxy Nexus	Bouncer*
Brand	Google	Tmobile
CPUABI	armeabi-v7a	armeabi
CPUABI2	armeabi	unknown
Host	vpbs3.mtv.corp.google.com	android-test-2.mtv.corp.google.com
Manufacturer	samsung	HTC
Model	Galaxy Nexus	T-Mobile myTouch 3G
Product	yakju	opal
Serial	01469107030XXXXX	unknown

*May be version dependent on requested SDK version of submitted application

INVASIVE VITAL SIGNS



	Galaxy Nexus	Emulator	Bouncer*
Phone number	1248760XXXX	15555215554	15555215504
Phone device	358350040XXX XXX	000000000000 00	112358132134559
Phone serial	8901260362485 XXXXXX	8901410321111 8510720	89014103211118510720
Sim name	T-Mobile	Android	T-Mobile
Network name	T-Mobile	Android	T-Mobile

*May be version dependent on requested SDK version of submitted application



- Android ID: 9774d56d682e549c
 - All emulators return this ID
 - Some older phones return this as well
 - Flashed OS mods tend to return this too
 - <http://stackoverflow.com/questions/6106681/android-ho>
- More recent tests indicate this ID may be changing and/or dynamic



- Google account associated with the Bouncer device:

```
base64.b64decode( '0yBtaWxlcy5rYXJsc29  
uQGdtYWlsLmNvbSwgY29tLmdvb2dsZQ== ' )  
' ; miles.karlson@gmail.com,  
com.google '
```

- miles.karlson@gmail.com



- Who does Miles hang out with?

- Check the Android contact lists

```
74.125.184.94 - - [10/May/2012:09:34:19  
-0500] "GET /index.html?  
q=TWljaGVsbGUgTG92aW4gbWljaGVsbGUuay5sZXZp  
bkBnbWFpbC5jb20= HTTP/1.1" 200 44
```

- michelle.k.levin@gmail.com

WHO IS MICHELLE?



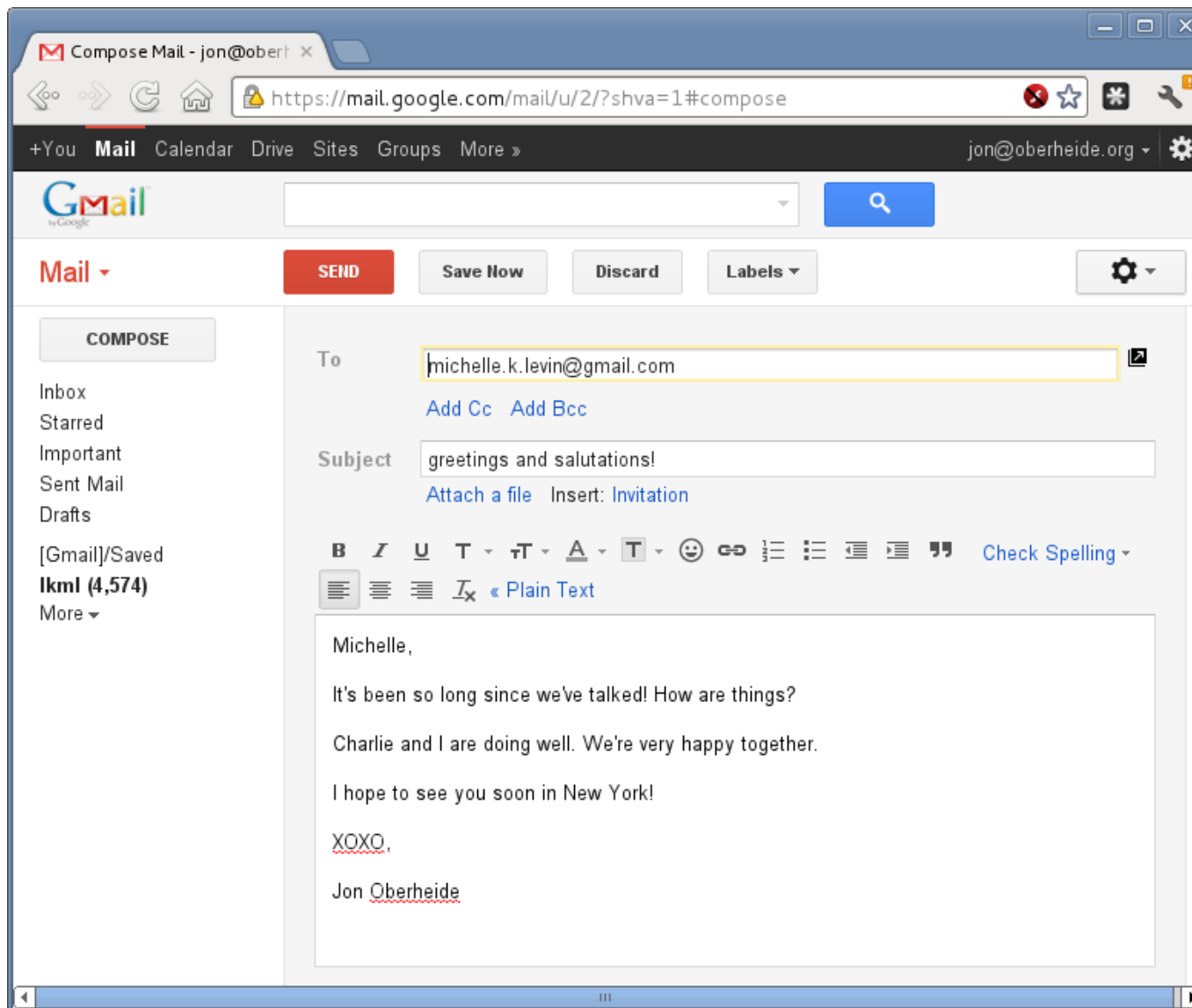
Search bar: michelle levin [camera icon] [magnifying glass icon]

Account: Charlie Miller [Share icon] [user icon]

Results: About 2,370,000 results (0.50 seconds) [SafeSearch dropdown] [Settings icon]



LET'S GET IN TOUCH!



MICHELLE LOVES SECURITY



The screenshot shows a web browser window with the title "Full-Disclosure Info Page". The address bar displays the URL <https://lists.grok.org.uk/mailman/listinfo/full-disclosure>. The page content includes:

- A heading: "Using Full-Disclosure"
- Text: "To post a message to all the list members, send email to full-disclosure@lists.grok.org.uk."
- Text: "You can subscribe to the list, or change your existing subscription, in the sections below."
- A heading: "Subscribing to Full-Disclosure"
- Text: "Subscribe to Full-Disclosure by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. This is a hidden list, which means that the list of members is available only to the list administrator."
- A form with the following fields:
 - "Your email address:" with the value "michelle.k.levin@gmail.com"
 - "Your name (optional):" with the value "Michelle Levin"
 - "Which language do you prefer to display your messages?" with the value "English (USA)"
 - "Would you like to receive list mail batched in a daily digest?" with radio buttons for "No" (selected) and "Yes"
- A "Subscribe" button.
- A heading: "Full-Disclosure Subscribers"
- Text: "To unsubscribe from Full-Disclosure, get a password reminder, or change your subscription options enter your subscription email address:"
- An input field and an "Unsubscribe or edit options" button.
- Text: "If you leave the field blank, you will be prompted for your email address"
- Footnote: "[Full-Disclosure](#) list run by [johnc at grok.org.uk](#)
[Full-Disclosure administrative interface](#) (requires authorization)"
- Logos at the bottom: "DELIVERED BY MAILMAN", "PYTHON Powered", and a cartoon ram head.

SDCARD CONTENTS



- download/cat.jpg
- download/
lady-gaga-300.jpg
- DCIM/Camera/
IMG_20120302
_142816.jpg
- android/data/
passwords.txt



lady-gaga-300.jpg



- Bouncer allows Internet access
- So what IPs does it come from?
 - 74.125.0.0/16
 - Also in recent tests: 209.85.128.0/17
 - Manual review: 173.194.99.0/16

```
$ whois 74.125.19.84 | grep OrgName
OrgName:          Google Inc.
$ whois 173.194.99.18 | grep OrgName
OrgName:          Google Inc.
```



- Bouncer runs your app for 5 minutes
 - Don't do anything bad for 5 minutes! Duh.
 - Not long term. Could be run later, longer...
- Timing attacks
 - Bouncer is not a physical device, QEMU is SLOW!
 - Performance/benchmark fingerprinting
 - NEON, Thumb, etc make it even more obvious



- Bouncer explores the app by emulating UI input, clicking, etc:

```
74.125.184.81 - - [10/May/2012:10:41:10 -0500]
"GET /foo?q=opened HTTP/1.1" 200 413
74.125.184.89 - - [10/May/2012:10:41:11 -0500]
"GET /foo?q=after_alert HTTP/1.1" 200 413
74.125.184.32 - - [10/May/2012:10:41:41 -0500]
"GET /foo?q=clicked_ok HTTP/1.1" 200 413
74.125.184.89 - - [10/May/2012:10:41:48 -0500]
"GET /foo?q=clicked HTTP/1.1" 200 413
```

- Predictable input actions can be used to fingerprint vs real user



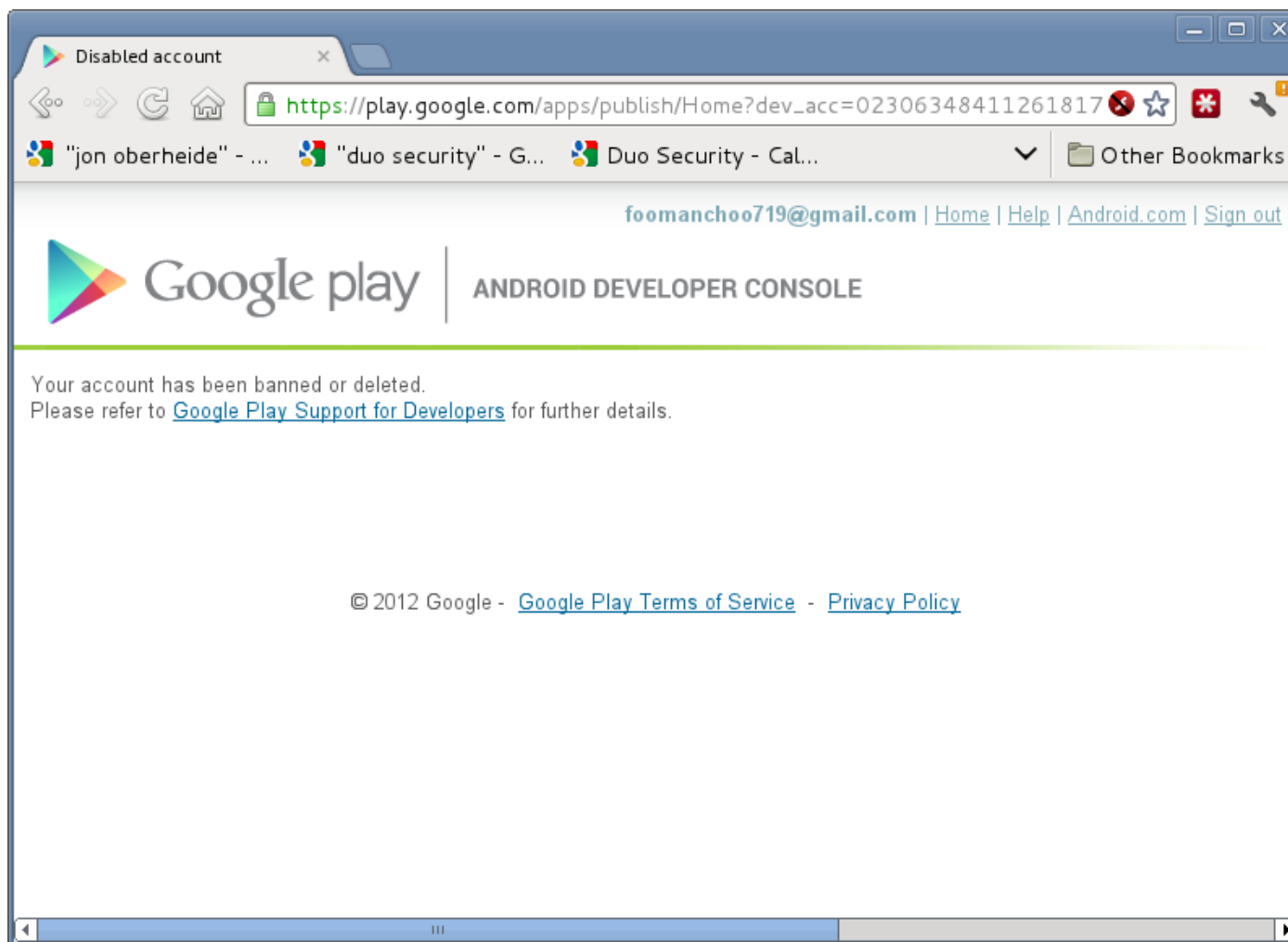
- Diagnosis
 - Intro to Bouncer and Google Play
- Exploratory surgery
 - Fingerprinting Bouncer and its environment
- **Open surgery**
 - **Abusing Bouncer in all sorts of fun ways**
- Suture and close
 - How Google can fix up Bouncer



Remote connect-back shell demo!

<http://www.youtube.com/watch?v=ZEIED2ZLEbQ>

MEDICAL LICENSE ISSUES



We got caught a couple times in our early experiments doing blatantly stupid stuff




- What happens when you get flagged?
- Inferred Bouncer process
 - Dynamic analysis of submitted app
 - If flagged, manual analysis by human operator
 - If deemed malicious, goodbye account!
- Manual analysis comes from different IP range (173.194.99.0/16)
 - Accidentally sent commands to the human operator once thinking it was my connect-back shell :-P

SUSPENDED



Notification of Google Play Developer Account Suspension

 **Google Play Support** googleplay-developer-support@google.com
to me ▾

This is a notification that your Google Play Publisher account has been terminated.


REASON FOR TERMINATION: Prior violations of the [Content Policy](#) and [Developer Content Policy](#) this or associated accounts as outlined in previous emails sent to the registered account(s)

Please note that Google Play Publisher suspensions are associated with developer registrations and related Google services. If you feel we have made an error, please contact us with additional information regarding this termination.

Do not attempt to register a new developer account. We will not be restoring suspended accounts.

The Google Play Team

Charlie
couldn't appeal :-(
Now banned from
iOS AND Android!


 **Charlie Miller**
to Google ▾


Hi. I received this email and am not sure why my account was suspended. It says I was suspended for policy violations. I just joined the program a few days ago and have not even published an app yet. Let me address the requirements required by the appeal process.

1. I do not have any other email addresses of any existing or prior Google Market or Play developer accounts.
2. I do not have additional removal or suspension violations on other Google Market or Play developer accounts.
3. I believe my account should be reinstated because I have done nothing wrong.

Thanks for your consideration in this matter.

...

 **Charlie Miller**
Can someone check the status of this? The account was invalidated and no one ...

 **Google Play Developer Support** googleplay-developer-support@google.com
to me, security, aludwig, cmiller ▾

May 24 (5 days ago)

Hi Charlie,

Thank you for your note.

Your Google Play Publisher account has been suspended due to prior violations of our terms of service by this or associated accounts. You may view these terms here:

<http://www.android.com/us/developer-distribution-agreement.html#agreement>
<http://www.android.com/market/terms/developer-content-policy.html>

Please note that Google Play Publisher suspensions are associated with developer registrations, and may span multiple account registrations and related Google services.

Please do not attempt to register a new developer account. Any subsequent registrations without an explicit reinstatement will be closed and your developer account fee not refunded.

We are unable to provide further details regarding this issue. We recommend your utilizing an alternative application distribution system and payment method for future orders.

Regards,
The Google Play Team



- Hmm, Bouncer runs app for 5 minutes
 - 5 free minutes of Google's computation resources!
- What to do with this “free” compute power provided by Google?
 - Find aliens? Cure cancer? Nah...
 - Let's fuzz Android on Android using Android!



- Android self-fizzer
- Queries server for which file to test
- Grabs the file with the browser
- Checks logs for crashes
- Reports crashlog to server if crash

FUZZING LOGS



```
74.125.184.23 - - [11/May/2012:09:47:35 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
74.125.184.19 - - [11/May/2012:09:47:37 -0500] "GET /pngs/178.png HTTP/1.1" 200 371
74.125.184.95 - - [11/May/2012:09:47:39 -0500] "GET /favicon.ico HTTP/1.1" 200 3638
74.125.184.83 - - [11/May/2012:09:47:41 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
74.125.184.92 - - [11/May/2012:09:47:42 -0500] "GET /pngs/179.png HTTP/1.1" 200 371
74.125.184.42 - - [11/May/2012:09:47:43 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
74.125.184.83 - - [11/May/2012:09:47:44 -0500] "GET /pngs/180.png HTTP/1.1" 200 371
74.125.184.21 - - [11/May/2012:09:47:46 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
74.125.184.46 - - [11/May/2012:09:47:47 -0500] "GET /pngs/181.png HTTP/1.1" 200 371
74.125.184.89 - - [11/May/2012:09:47:48 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
74.125.184.80 - - [11/May/2012:09:47:49 -0500] "GET /pngs/182.png HTTP/1.1" 200 371
74.125.184.41 - - [11/May/2012:09:47:51 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
74.125.184.31 - - [11/May/2012:09:47:52 -0500] "GET /pngs/183.png HTTP/1.1" 200 371
74.125.184.82 - - [11/May/2012:09:47:55 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
74.125.184.24 - - [11/May/2012:09:47:57 -0500] "GET /pngs/184.png HTTP/1.1" 200 371
74.125.184.86 - - [11/May/2012:09:47:58 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
74.125.184.37 - - [11/May/2012:09:47:59 -0500] "GET /pngs/185.png HTTP/1.1" 200 371
74.125.184.38 - - [11/May/2012:09:51:17 -0500] "GET /pngs/223.png HTTP/1.1" 200 380
74.125.184.94 - - [11/May/2012:09:51:24 -0500] "GET /cgi-bin/getfile.pl HTTP/1.1" 200 78
```

- EULA FUN
 - Bouncer clicks dialogs
 - Our submitted app pops up a EULA dialog
 - Bouncer agrees to our EULA?!?!
- “You agree you are not Bouncer”, Bouncer will click yes! Liar!





- Areas to explore further
 - Static analysis by Bouncer
 - Taint propagation disruption
- Challenges
 - Time, effort
 - Clean feedback loop



- Did submit rageinthecage once
 - Still ran in Bouncer?!?
 - But probably flagged.
- One would expect a static analysis stage to short-circuit dynamic run
 - But dynamic info may still be useful to Google



- Sometimes the APK never calls back
 - Presumably this means it wasn't dynamically tested
 - The guess is it fails some static detection
- One inferred signature: “/system/bin”
 - App with “/system/bin/lis” in it never called back
 - But did call back when string was constructed dynamically!



- Taint tracking!
- Example use case:
 - Snarf contact data and send over the network
 - Write “signature” to flag such suspicious
- Depends on propagating taint
- How to disrupt taint propagation?
 - Reflect/filter data off/through interfaces that do not track taint metadata



- Tricky interfaces to propagate through
 - Android's SharedPreferences
 - Android's Binder IPC
 - Android's LogCat interface
 - Java's DirectBuffer interface
- Implemented these “taint breakers”
- Not enough testing to conclude which were effective though



- Diagnosis
 - Intro to Bouncer and Google Play
- Exploratory surgery
 - Fingerprinting Bouncer and its environment
- Open surgery
 - Abusing Bouncer in all sorts of fun ways
- **Suture and close**
 - **How Google can fix up Bouncer**



- Some easy stuff
 - eg. hide strings, emulator identifiers, etc
- Some medium stuff
 - eg. diversify IP ranges (re-use Safe Browsing crawling infrastructure)
- Some hard stuff
 - eg. prevent a sufficiently convincing model of a real user's Android device
- Generally, avoid being an oracle



- Dynamic analysis is HARD.
 - That part of Bouncer will never be perfect
 - So, attack the problem from a different angle
- Dynamic analysis portion of Bouncer only looks at the submitted app
- There's a lot of metadata related the app submission that Google judges
 - eg. Charlie got his wife's CC rejected since he used the same IP to sign up for a subsequent account



CODE SIGNING!!!

- Over two years later, still no code signing
- Static and dynamic analysis suddenly becomes less horrible
- Good for exploit mitigation too

Wrap-Up

- Native code support sucks.
 - Not so easy to take away
 - Build-time signing / loader verification
- Android homework
 - Poke at the GTalkService code path
 - Write some RootStrap payloads



- Bouncer doesn't have to be perfect to be useful
 - It will catch crappy malware
 - It won't catch sophisticated malware
 - Same as AV, IDS, <your favorite security tech>
- How much does Bouncer raise the bar?
 - Currently: not much
 - Future: hopefully more?



- Special thanks
 - Dr. Valasek, Dr. Trumbour, and Dr. Jimbo
- Greetz
 - #busticati
 - redpantz, jlamer, deft, redpig, krnlpool, bliss, nelhage, tavisio, twiz, rocky, larry, deft, thing2, drb
 - Space Pope



QUESTIONS?

Jon Oberheide
@jonoberheide
jon@oberheide.org

Charlie Miller
@0xcharlie
cmiller@openrce.org