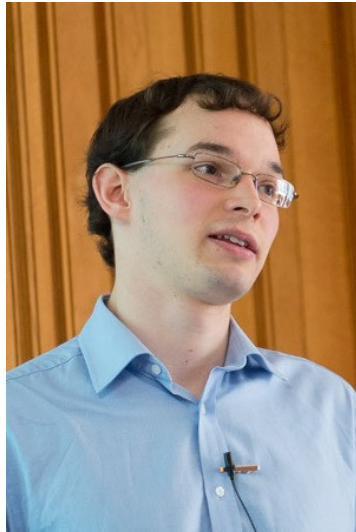


Security metrics for the Android ecosystem



**Daniel
Thomas**



**Alastair
Beresford**



**Andrew
Rice**

Firstname.Surname@cl.cam.ac.uk
<http://androidvulnerabilities.org>

| | | | | | | | | | | |
|---------------|------|------|------|------|------|------|------|------|------|------|
| Daniel gpg: | 5017 | A1EC | 0B29 | 08E3 | CF64 | 7CCD | 5514 | 35D5 | D749 | 33D9 |
| Alastair gpg: | 9217 | 482D | D647 | 8641 | 44BA | 10D8 | 83F4 | 9FBF | 1144 | D9B3 |
| Andrew gpg: | 43BF | 45D1 | 1B36 | F45C | 3F07 | DA49 | BDB8 | 8932 | 5CAC | F039 |

Smartphones contain many apps written by a spectrum of developers



How “secure” is a smartphone?

Root/kernel exploits are harmful

- Root exploits break permission model
- Cannot recover to a safe state
- 37% Android malware uses root exploits (2012)
- We're interested in critical vulnerabilities, exploitable by code running on the device

Hypothesis: devices vulnerable because they are not updated

- Anecdotal evidence is that updates rarely happen
- Android phones, sold on 1-2 year contracts

No central database of Android vulnerabilities: so we built one

[AVO](#) [HOME](#) [SUBMIT VULNERABILITY](#)

AndroidVulnerabilities.org

Stagefright

(json)

CVE numbers: CVE-2015-1538 [[nakedsecurity-stagefright](#)], CVE-2015-1539 [[nakedsecurity-stagefright](#)], CVE-2015-3824 [[nakedsecurity-stagefright](#)], CVE-2015-3826 [[nakedsecurity-stagefright](#)], CVE-2015-3827 [[nakedsecurity-stagefright](#)], CVE-2015-3828 [[nakedsecurity-stagefright](#)], CVE-2015-3829 [[nakedsecurity-stagefright](#)]

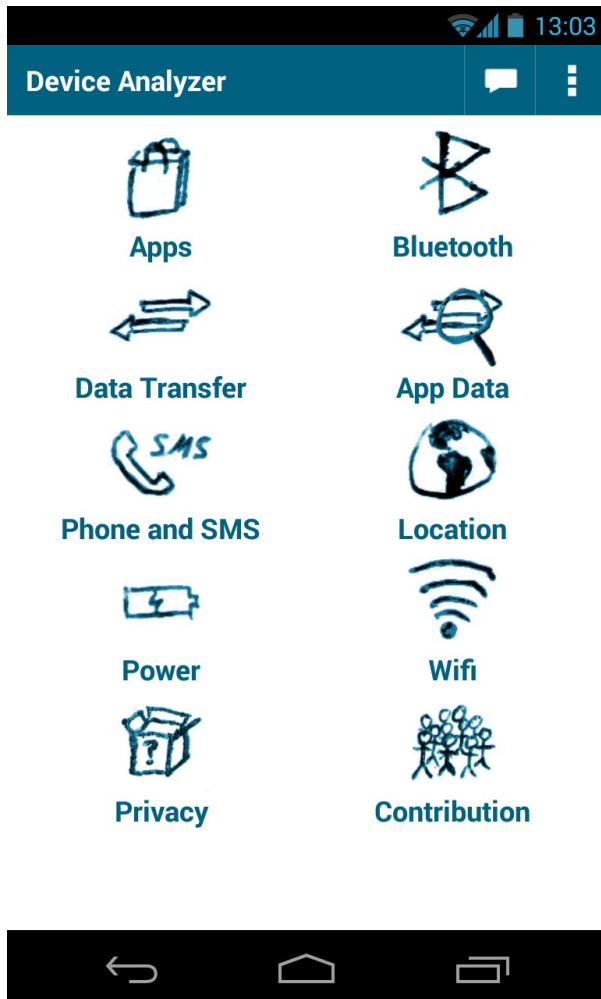
Responsibly disclosed?: True

Categories: system, network

Details: Drake said that the vulnerabilities can be exploited by sending a single multimedia text message to an unpatched Android smartphone. While the exploit is deadly, in some cases, where phones parse the attack code prior to the message being opened, the exploits are silent and the user would have little chance of defending their data. [[techworm-stagefright](#)] Stagefright is the media playback service for Android, introduced in Android 2.2 (Froyo). Stagefright in versions of Android prior to 5.1.1_r9 may contain multiple vulnerabilities, including several integer overflows, which may allow a remote attacker to execute code on the device. [[cert-kb-stagefright](#)]

Discovered by: Joshua J. Drake [[zimmerium-stagefright](#)] on: 2015-04-09 [[techworm-stagefright](#)]

Device Analyzer gathers statistics on mobile phone usage



- Deployed May '11
- 30,000 contributors
- 4,000 phone years
- 180 billion records
- 10TB of data
- 1089 7-day active contributors

The screenshot shows the 'Phone and SMS' screen. At the top, it says 'Saving screenshot...'. Below that, a header bar says 'Phone and SMS'. The main area displays 'Phone calls:' and 'Text messages:' sections. The 'Phone calls' section has a table with columns: Incoming, Outgoing, and Total. The 'Text messages' section has a table with columns: Received, Sent, and Total. Below the tables, it shows 'Active Operator giffgaff', 'Roaming no', 'Signal strength 19', 'Ringer mode normal', and 'Data Collected 12 Nov 2013 13:12:25'. At the bottom, there's a navigation bar with back, home, and recent apps buttons.

| | Incoming | Outgoing | Total |
|------------|----------|----------|---------|
| Today | 0:00 | 0:00 | 0:00 |
| This Month | 11:40 | 36:23 | 48:03 |
| Last Month | 28:53 | 1:05:07 | 1:34:00 |

Text messages:

| | Received | Sent | Total |
|------------|----------|------|-------|
| Today | 1 | 1 | 2 |
| This Month | 61 | 56 | 117 |
| Last Month | 176 | 150 | 326 |

Active Operator **giffgaff**
Roaming **no**
Signal strength **19**
Ringer mode **normal**
Data Collected **12 Nov 2013 13:12:25**

<https://deviceanalyzer.cl.cam.ac.uk>

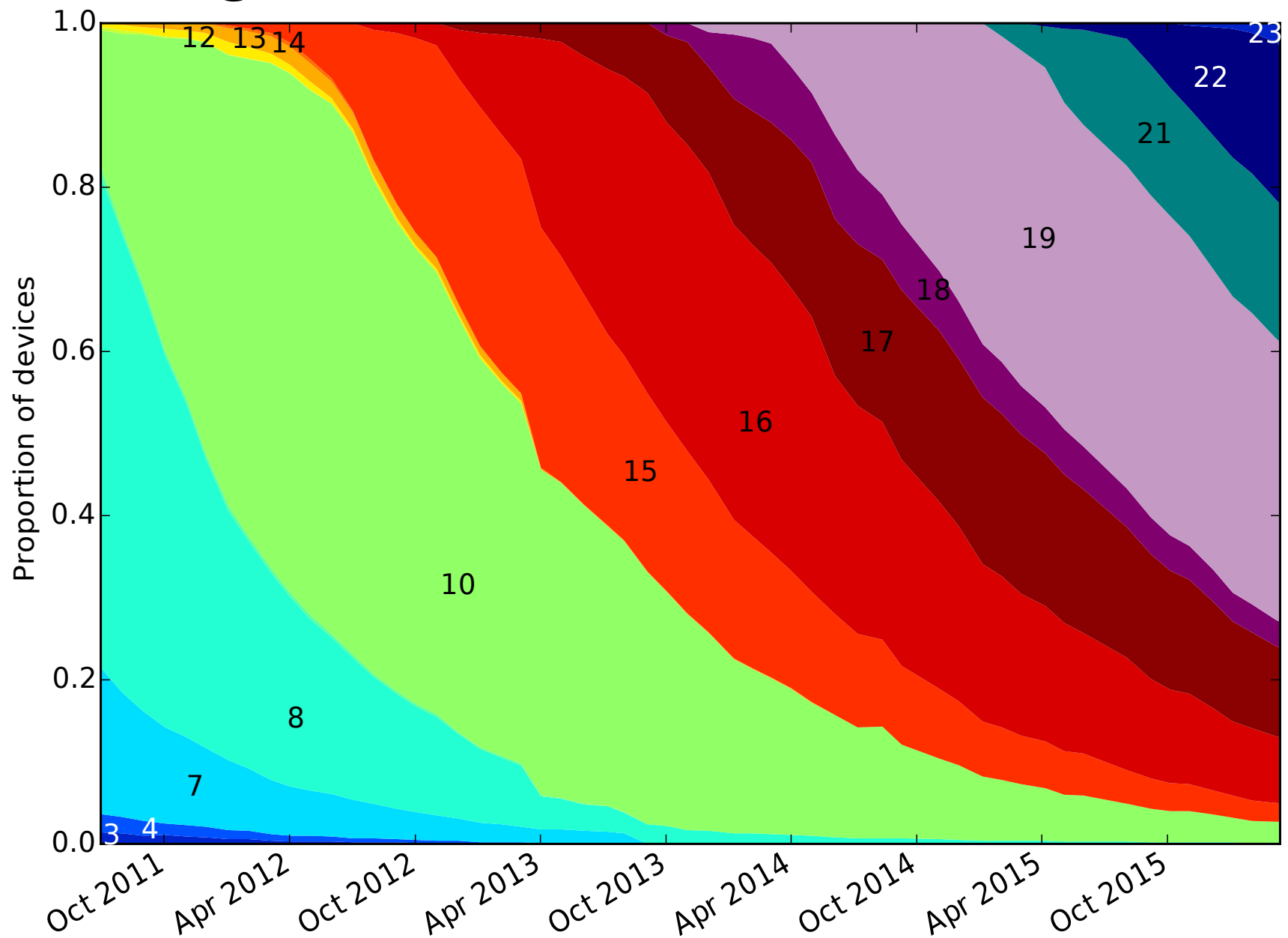
Device Analyzer gathers wide variety of data

- Including: system stats
 - OS version and build number
 - Manufacturer and device model
 - Network operators



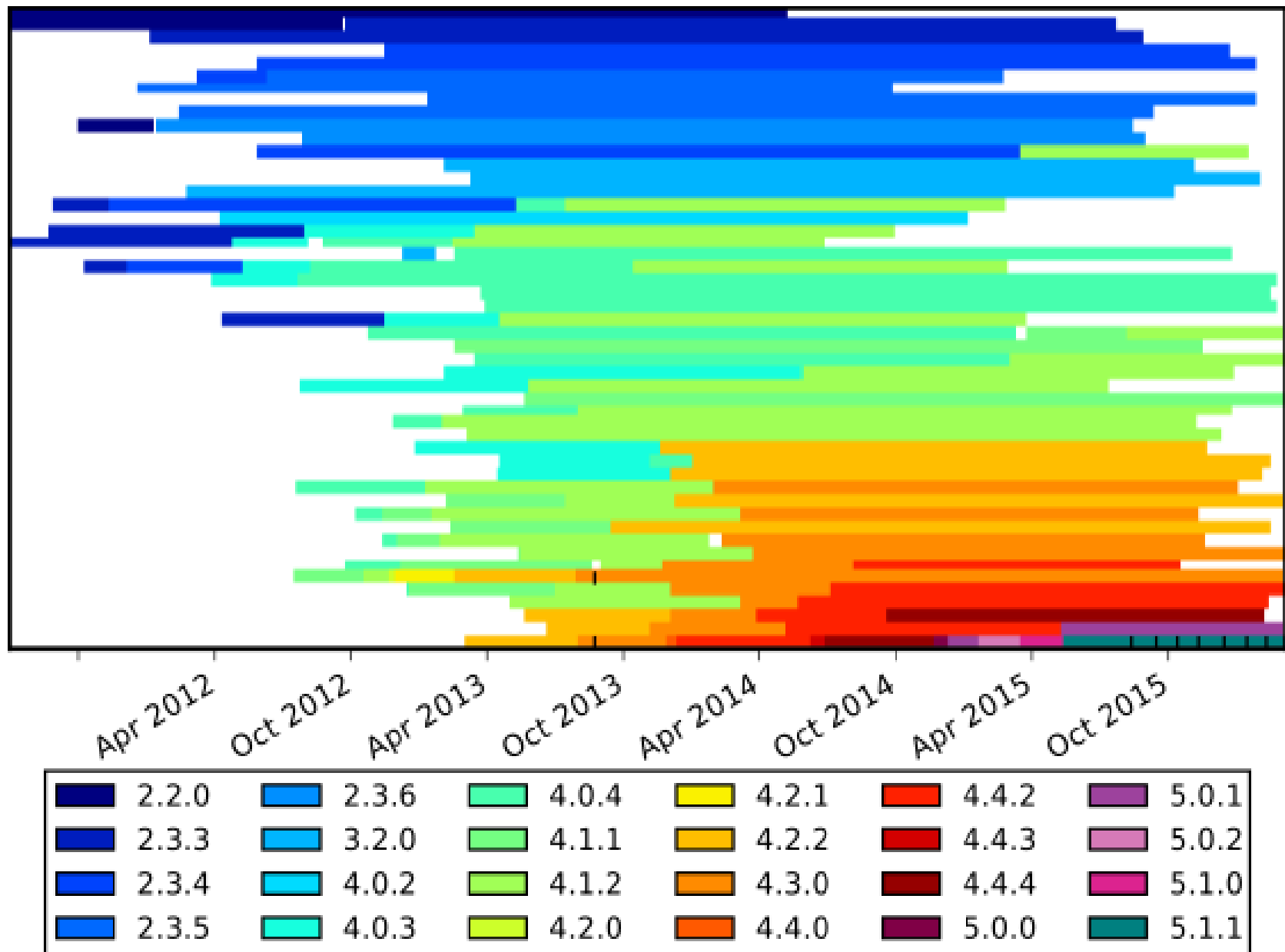
Is the *ecosystem* getting updated?

Google data: device API levels

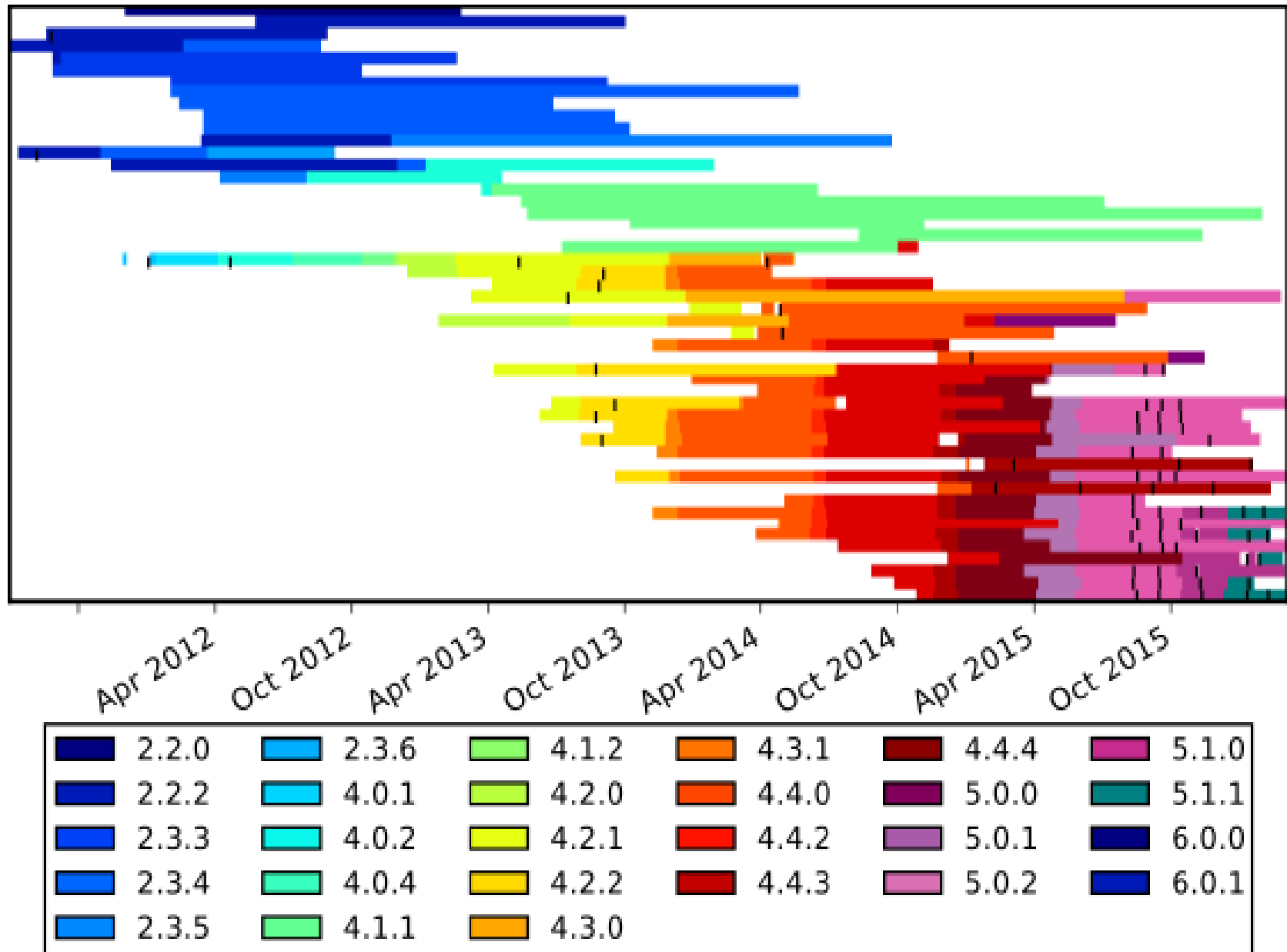


Are *devices* getting updated?

Samsung devices by OS version



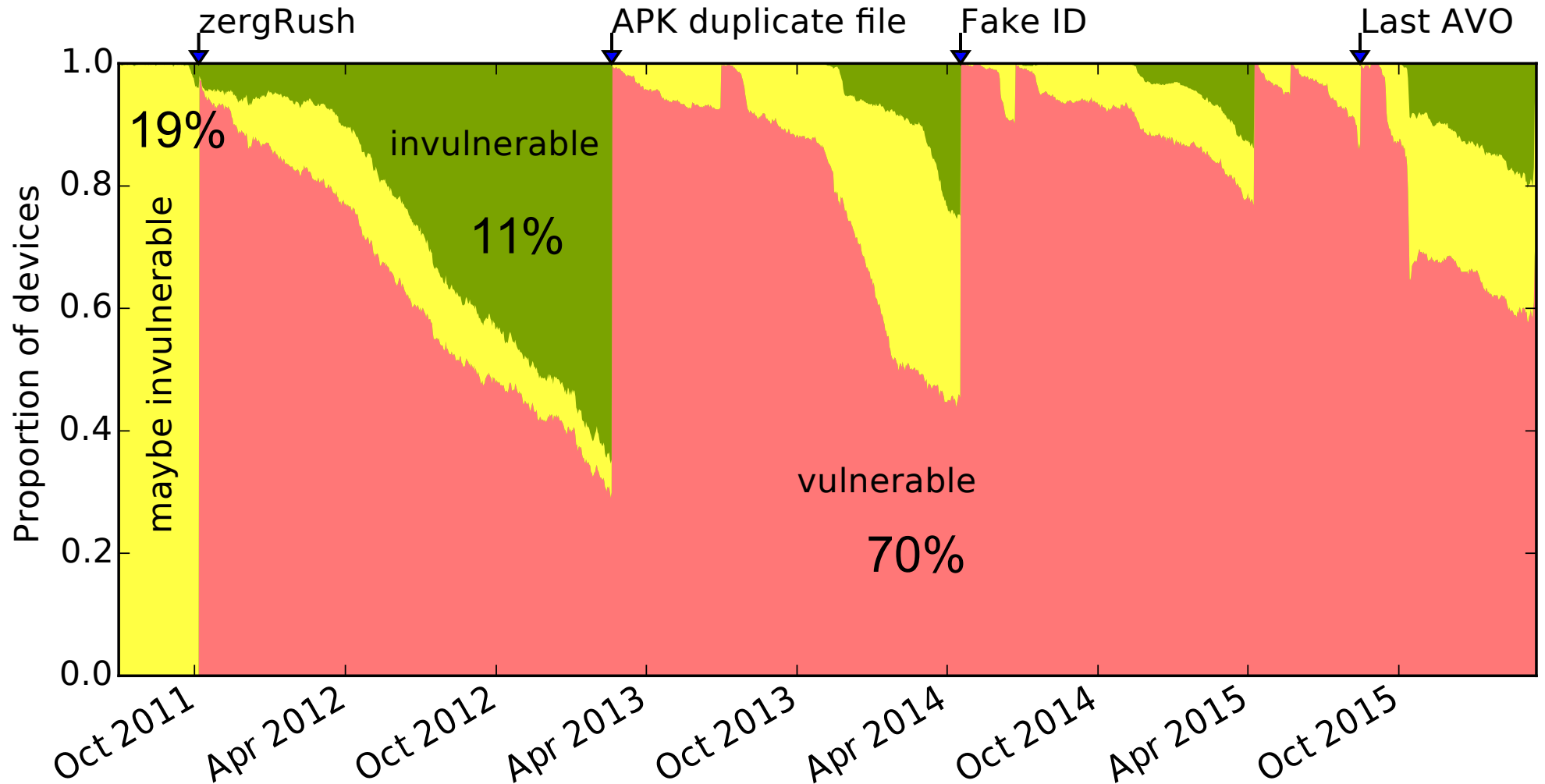
LG devices by OS version



Connecting the two data sets: assume OS version → vulnerability

- We have an OS version from Device Analyzer
- We have vulnerability data with OS versions
- Match on OS and Build Number and assign:
 - Vulnerable
 - Maybe invulnerable
 - Invulnerable (not known vulnerable)

On average, 70% are vulnerable



The FUM metric measures the security of Android devices

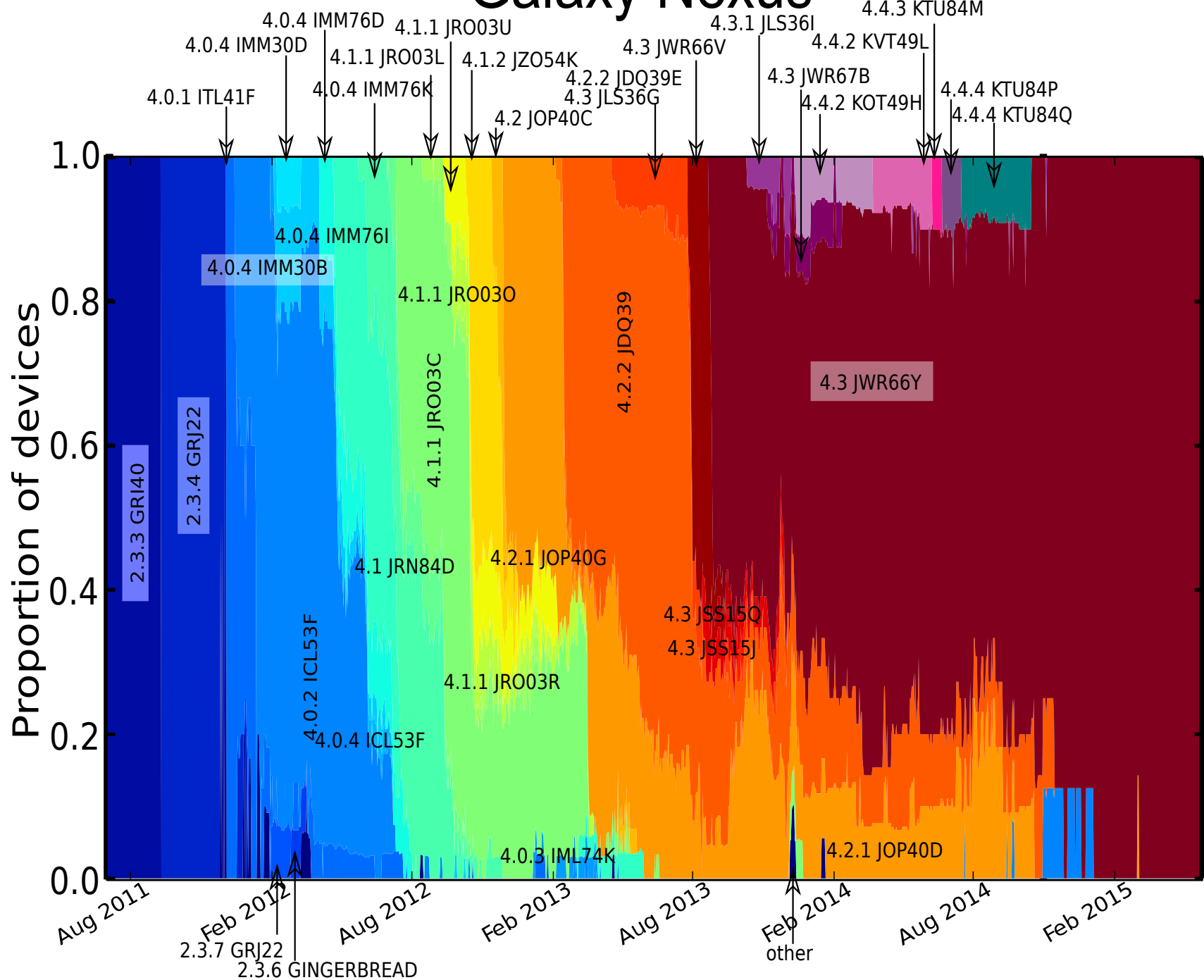
$$FUM\ score = 4 \cdot f + 3 \cdot u + 3 \cdot \frac{2}{1 + e^m}$$

f free from (known) vulnerabilities

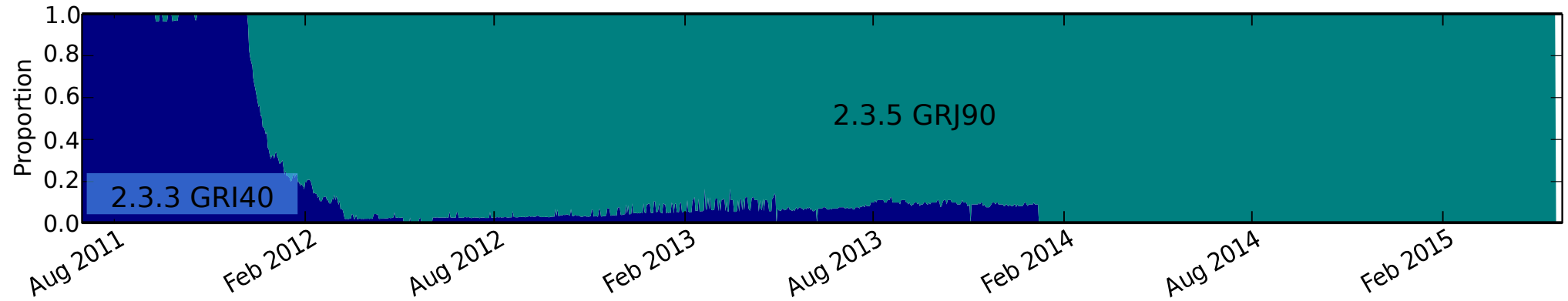
u updated to the latest version

m mean unfixed vulnerabilities

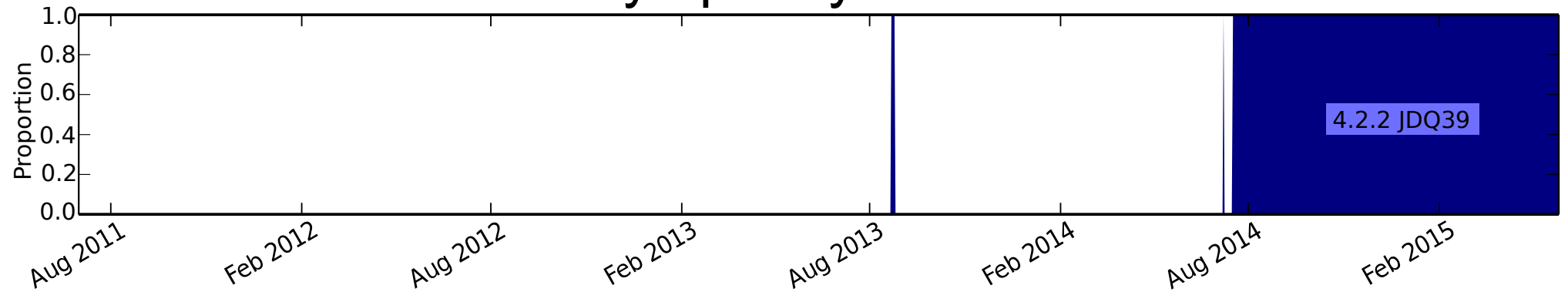
Galaxy Nexus



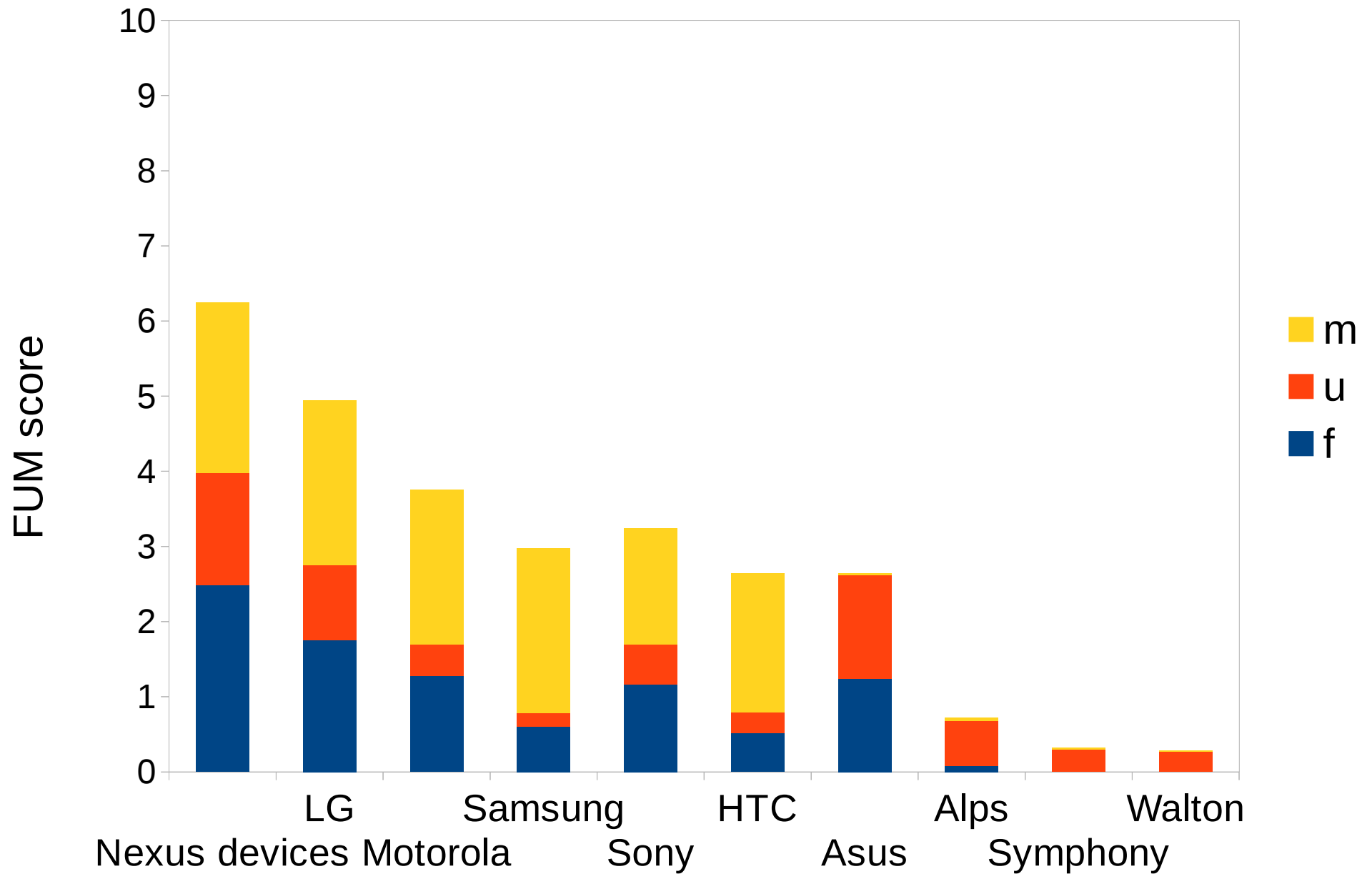
HTC Desire HD A9191



Symphony W68



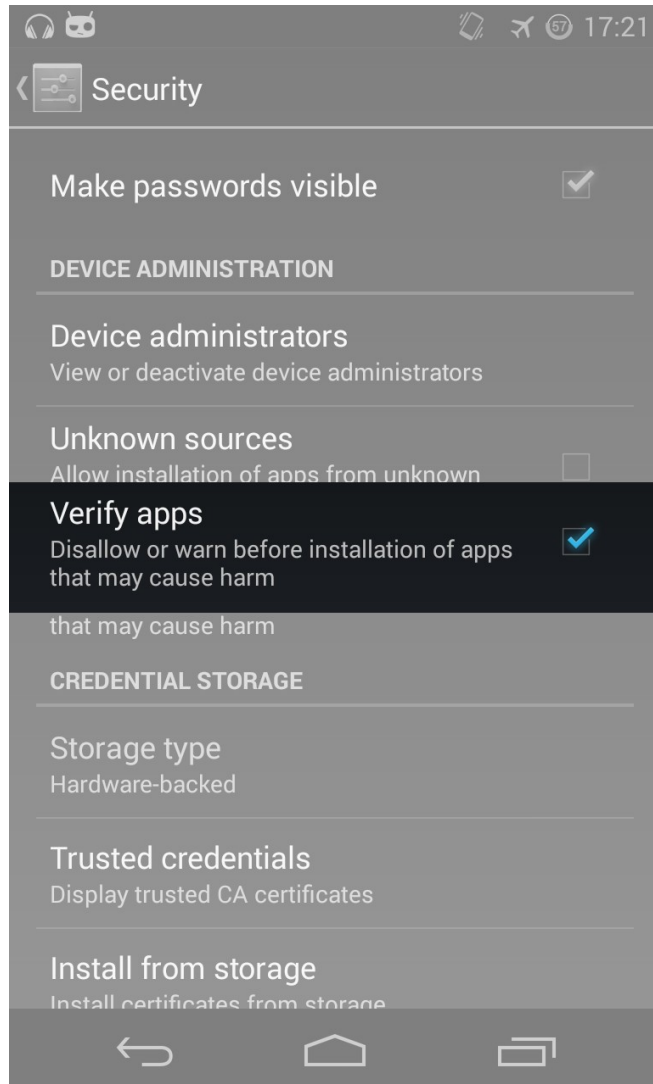
FUM scores



Why is fixing vulnerabilities hard: software ecosystem is complex

- Division of labour
 - Open source software
 - Core OS production
 - Driver writer
 - Device manufacturer
 - Retailer
 - Customer
- Apple and Google have different models
 - Hypothesis: Apple's model is more secure

Google to the rescue

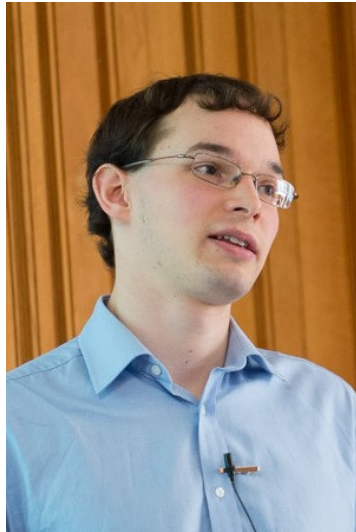


- Play Store
- Verify apps
- Android Security Patch Level

Conclusions

- 70% of Android devices are vulnerable
- Ecosystem complex; lack of transparency
- FUM metric is a robust measure of security
 - A step towards an economic incentive

Security metrics for the Android ecosystem



**Daniel
Thomas**



**Alastair
Beresford**



**Andrew
Rice**



UNIVERSITY OF
CAMBRIDGE

| | | | | | | | | | | |
|---------------|------|------|------|------|------|------|------|------|------|------|
| Daniel gpg: | 5017 | A1EC | 0B29 | 08E3 | CF64 | 7CCD | 5514 | 35D5 | D749 | 33D9 |
| Alastair gpg: | 9217 | 482D | D647 | 8641 | 44BA | 10D8 | 83F4 | 9FBF | 1144 | D9B3 |
| Andrew gpg: | 43BF | 45D1 | 1B36 | F45C | 3F07 | DA49 | BDB8 | 8932 | 5CAC | F039 |

Example: Android APK duplicate file

- OS does not check for duplicate files in APK
- Not a traditional kernel vulnerability
- Affected all manufacturers and versions > 1.5
- Timeline:
 - February 2013: discovered
 - February 2013: fixed
 - July 2013: Public announcement
- Is the responsible disclosure period sufficient to protect users?

Device Analyzer is a good example of Privacy by Design principles

- Transparency, consent, notice and disclosure
- Purpose
- Security
- Access to data and withdrawal
- Proactive privacy design
- Privacy by default

Device Analyzer is representative

- Compared with Google Play API data: Device Analyzer is slightly better
- Compared with User-Agent headers from Rwanda: Device Analyzer is better
- Compared with MDM data from a FTSE 100 company: Device Analyzer is slightly worse

Nexus and non-Nexus devices

