RootedCON 2014: “iOS: Back to the Future”
CCN-CERT 2014: “iOS: Back to the Future II”

IN PREVIOUS ‘EPISODES’...
iOS: Back to the Future
iOS: Back to the Future II
iOS: Back to the Future II

- 2012 (Feb): Vulnerability discovery
  - iOS 5.x (*iOS OTA software updates* – 5.0.1) - Private during 2+ years
- 2014 (Feb 6): Vulnerability notified to Apple
- 2014 (March 1): Changes in Apple update servers
  - For iOS 7.x: Future dates verifications in server (If-Modified-Since)
- 2014 (March 8): Vulnerability presented at RootedCON 2014 (ES)
  - 2014 (June 3): Vulnerability presented at Area41 (EN)
    - (June 4) iCamasu: [http://blog.dinosec.com/2014/06/icamasu.html](http://blog.dinosec.com/2014/06/icamasu.html)
  - 2014 (June 24): Vulnerability published in DinoSec blog
- 2014 (Sep 17): iOS 8 (*CVE-2014-4383*)
  - iOS 8: Future dates verifications in mobile client (Last-Modified)
  - iOS vulnerable versions: 5, 6 y 7
- 2014 (Nov 20): Vulnerability re-notified to Apple
- **2015 (TODAY): iOS 8- still vulnerable (similar to March 1)**

TODAY...
How does the OTA firmware (or system) update process work in Android?

What about Android? (Lack of public details or research)
Android Fragmentation
Android Fragmentation

Platform Versions

This section provides data about the relative number of devices running a given version of the Android platform.

For information about how to target your application to devices based on platform version, read Supporting Different Platform Versions.

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>0.4%</td>
</tr>
<tr>
<td>2.3.3 - 2.3.7</td>
<td>Gingerbread</td>
<td>10</td>
<td>7.4%</td>
</tr>
<tr>
<td>4.0.3 - 4.0.4</td>
<td>Ice Cream Sandwich</td>
<td>15</td>
<td>6.4%</td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>18.4%</td>
</tr>
<tr>
<td>4.2.x</td>
<td></td>
<td>17</td>
<td>19.8%</td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td>18</td>
<td>6.3%</td>
</tr>
<tr>
<td>4.4</td>
<td>KitKat</td>
<td>19</td>
<td>39.7%</td>
</tr>
<tr>
<td>5.0</td>
<td>Lollipop</td>
<td>21</td>
<td>1.6%</td>
</tr>
</tbody>
</table>

Data collected during a 7-day period ending on February 2, 2015. Any versions with less than 0.1% distribution are not shown.

Note: Because this data is gathered from the new Google Play Store app, which supports Android 2.2 and above, devices running older versions are not included. However, in August, 2013, versions older than Android 2.2 accounted for about 1% of devices that checked in to Google servers (not those that actually visited Google Play Store).

https://developer.android.com/about/dashboards/index.html

Focus: 4.x - 4.4.4
Google Play Services (GPS)

• New (security) features updated...
  – ... without getting new system updates
    • Android version remains (Android 2.3+)
    • Automatic GPS updates
      – Cannot uninstall GPS app updates
      – Advantages (updates) & disadvantages (ecosystem)

• Security examples
  – Fix for CSS (ChangeCipherSpec) injection vuln.
    • OpenSSL (CVE-2014-0224)
  – Device administrator (app) – e.g. remote wipe
  – Verify Apps (Settings)
Google Play Services (GPS)
(Recommended) Security Research

• State-of-the-Art (Re-search)
  – Search, search, search, search, search, search...
  – Common use cases (e.g. regular users)
• Architecture analysis
  – Android internals
• Network traffic analysis
• Code analysis
  – Source code and/or reverse engineer (RE)

*Multiple ways & tools to do the same thing*

Cyclical & Repetitive process
ANDROID (OTA) Firmware Updates
State-of-the-Art
How To Check for OTA Updates?

- Manually or automatically
- Settings – About phone: System updates

Sometimes you get it, sometimes you don’t... 😊
How (Not) To Force an OTA Update?

- Settings – Apps – All:
  - Google Services Framework (GSF)
    - Clear data & Clear cache
  - Google Play services (GPS)
    - Manage space (...) & Clear cache
Android OTA Updates Rollouts

- Dan Morrill (Android Engineer)
  - Rollouts in phases
    - 1% for 24-48h (errors?)
    - 25%-50%-100% in 1-2 weeks
  - Randomly (it is not 1\textsuperscript{st}-come/1\textsuperscript{st}-served)
    - If you are out of luck... wait for next batch
    - “Check for update” button – one roll of the dice/batch
    - “… devices check for updates every 24h (I think?)…”

Kit-Kat4.4 Kitkat has started rolling out on the N4!
http://www.reddit.com/r/Android/comments/1r1dz4/44_kitkat_has_started_rolling_out_on_the_n4/cdiwt6f

Official details: Reddit? Google+?

Be patient! 😊
Don’t Clear Google Services Framework Data

• Dan Morrill (Android Engineer)
  – Changes the primary device ID (known by Google)
    • Like a factory reset ([GCM] ID is random)
  – Invalidates the tokens used by apps using GCM
    • Apps stop getting GCM push-messages (get new ID)
    • Nearly all Google apps, plus third-party apps
      – Play Store: log out and log back in
      – Gmail: transparently (won’t get e-mail notifications for a while?)
      – Other apps: clear data

PSA: Do not clear data for Google Service Framework

http://www.reddit.com/r/Android/comments/1r1n5z/psa_do_not_clear_data_for_google_service_framework/cdiymrp

Official details: Reddit? Google+?

A new roll of the dice
ANDROID INTERNALS
COMPONENTS INVOLVED IN THE FIRMWARE UPDATE PROCESS
Who Is The Hacker in Frozen?
Android and GPS?

While Android remains free for anyone to use as they would like, only Android compatible devices benefit from the full Android ecosystem. By joining the Open Handset Alliance, each member contributes to and builds one Android platform—not a bunch of incompatible versions.

Google Play Services (GPS)

• Part of GMS (Google Mobile Services) ≈ GPS
  – Open Android Open Source Project (AOSP)
  – Closed GMS add-on (Proprietary - licenses)
    • Google apps, services and API extensions
    • Certify mobile devices (OEMs)
  – Google Apps (GApps)
    • Chrome, Maps, Google+, Hangouts, Google Play, Google Drive, Fit, Ads, YouTube, Wallet, Gmail, Google app, Analytics, Wearables, Games...

• Google Apps & Google Play apps updates
  – ≠ Play Store app

• Google services authentication and features
  – Google APIs + GPS SDK

Google Play Services (GPS)

• Main versions
  – Google Play services, Version 6.5 (December 2014)
  – Google Play services, Version 6.1 (October 2014)
  – Google Play services, Version 5.0 (July 2014)
    • Security API: Dynamic security provider
  – Google Play services, Version 4.4 (May 2014)
    • ???
  – Google Play services, Version 4.3 (March 2014)

• Runs as a background service
  – System integration?
Android OTA Update Components

• Who is the OTA update component?
  – Settings?

• Google Services Framework (GSF)
  – SystemUpdateService

• Google Play services (v4.4+)
  – SystemUpdateService (not in v4.3)
Google Services Framework (GSF)

• SystemUpdateService (& Activity)
  – GPS < v4.4
  – com.google.android.gsf.update.SystemUpdateService

• APK: GoogleServicesFramework.apk

$ adb shell am start -n com.google.android.gsf/.update.SystemUpdateActivity
Google Play Services (GPS v4.4)

- **SystemUpdateService (& Activity)**
  - GPS >= v4.4 (May 2014)
    - android.permission.RECOVERY
  - com.google.android.gms.update.SystemUpdateService
  - Different app/process from ‘Settings’
- **APK: GoogleServicesFramework.apk**
  - PrebuiltGmsCore.apk (< v4.4)

```
$ adb shell am start -n com.google.android.gms/.update.SystemUpdateActivity
```

Google Play Services 4.4... With... Firmware Installer...

http://www.androidpolice.com/2014/05/08/apk-teardown-google-play-services-4-4-explodes-with-android-wear-support-firmware-installer-and-much-more/
Inspecting Android App Components (1/3)

• Inspect device and app components and behavior

• Android system log information
  — Live log: "adb logcat"
  — Clear log: "adb logcat -c"
  — Launch app or open window (GUI)
  — Show log (& exit): "adb logcat -d"

Settings: com.android.settings[.DeviceInfoSettings]
Inspecting Android App Components (2/3)

- Clear log, launch app, and show log
  - Settings - About phone - System updates

```bash
$ adb logcat -c
<Launch app>
$ adb logcat -d
--------- beginning of /dev/log/system
I/ActivityManager(  773): START u0
{cmp=com.google.android.gsf/.update.SystemUpdateActivity} from
pid 3169
--------- beginning of /dev/log/main
...
I/ProcessStatsService(  773): Prepared write state in 4ms
I/ActivityManager(  773): Displayed
com.google.android.gsf/.update.SystemUpdateActivity: +223ms
```
Inspecting Android App Components (3/3)

• Clear log, launch app, and show log
  – Settings - About phone - System updates

$ adb logcat -c
<Launch app>
$ adb logcat -d
--------- beginning of /dev/log/system
I/ActivityManager( 512): START u0
{cmp=com.google.android.gms/.update.SystemUpdateActivity} from
pid 2497
--------- beginning of /dev/log/main
...
I/ProcessStatsService( 512): Prepared write state in 4ms
I/ActivityManager( 512): Displayed
com.google.android.gms/.update.SystemUpdateActivity: +289ms

Android 4.4.3
GPS: v6.5.x
GSF: v4.4.3-...
HTTP & HTTPS & ...

NETWORK TRAFFIC ANALYSIS

GOOGLE PROTOCOL BUFFERS & CERTIFICATE PINNING
Google Protocol Buffers

• Google's data interchange binary format
  – Efficient yet extensible format, and upgradeable (schema changes)
    • “... think XML, but smaller, faster, and simpler"
      – Optimize human readable formats: JSON, XML...
    – Language and platform neutral (C++, Java or Python...)
    – Serialize structured data (objects/types) in protobuf messages (communication protocols, storage, etc)

• Google's lingua franca for data
  – Sep 3, 2014: "...there are 48,162 different message types defined in the Google code tree across 12,183 .proto files."

https://code.google.com/p/protobuf/
https://github.com/google/protobuf/
https://developers.google.com/protocol-buffers/
https://developers.google.com/protocol-buffers/docs/overview
Protocol Buffers Definition

• Create a .proto file
  – Protobuf message type's definition
    • Names and declared type for each field
• Compile with protoc (protocol buffer compiler)
  – C++, Java or Python
  – Generates code to work with the message types
    • Get and set field values
    • Serialize messages to an output stream
    • Parse messages from an input stream

https://developers.google.com/protocol-buffers/docs/tutorials
https://developers.google.com/protocol-buffers/docs/reference/overview
(API reference)
### .proto File

- **Messages & fields** (tag #, name and data type)
  - [https://developers.google.com/protocol-buffers/docs/proto](https://developers.google.com/protocol-buffers/docs/proto)

```proto
message Eval {
    required string name = 1;  // UTF-8 or …
    optional int32 score = 2;  // 1-10
    optional string comments = 3;  // …7-bit ASCII
    optional Goal goal = 4;
}

message Goal {
    required int32 top = 1 [default = 10];
}
```

Default values: string (**""**), bools (false), numeric (0), enums (first value in definition)
Protocol Buffers Encoding

- Encoded binary wire format
  - Protobuf message serialized to output stream
  - Small and numeric (varints: base128)
  - Protobuf message: key-value pairs
    - Key in wire (two values):
      - Field number from .proto file (tag #)
      - Wire type (length of the following value)
    - Value in wire (but not the name)
      - Guess the semantics of each field (or .proto file)

https://developers.google.com/protocol-buffers/docs/encoding
Protobuf Decoder Burp Extension

- Burp BApp Store (Burp extensions)
  - Protobuf Decoder (v1.0)
    - Jython 2.7+
    - Marcin Wielgoszewski
    - Decode raw protobuf messages
      - Content-type: application/x-protobuf
      - Uses “protoc --decode_raw”
      - Even if its .proto file is not available: view
      - If .proto file is loaded (right click): modify and tamper
        » Automatically deserialize message as... (or select manually)
        » Reserialize (alert if there are required fields missing)

https://pro.portswigger.net/bappstore/
https://github.com/mwielgoszewski/burp-protobuf-decoder
http://www.tssci-security.com/archives/2013/05/30/decoding-and-tampering-protobufserialized-messages-in-burp

State-of-the-art of open-source security tools
Using Protobuf Decoder Burp Ext. (1/3)

http://android.clients.google.com

• GSF (4.4.4 default - no gzip)

```http
POST /checkin
Content-type: application/x-protobuf
User-Agent: Android-Checkin/2.0 (hammerhead KTU84P); gzip
...  
Content-Length: 7
```

• GPS (4.4.4 updated - gzip)

```http
POST /checkin
Content-type: application/x-protobuf
Content-encoding: gzip
User-Agent: Dalvik/2.0.0 (Linux; U; Android 4.4.4; Nexus 5 Build/KTU84P)
...
Content-Length: 27
```
Using Protobuf Decoder Burp Ext. (2/3)

• Cannot decode HTTP /checkin requests
  – Coming from Android 4.4.x
    • E.g. Google Play services v6.5.99
  – Content-type
  – Burp Proxy – Options – Miscellaneous:
    • Unpack gzip in requests (off) / responses (on)

• Unfortunately, it is not enough...
  – New version
    • New Content-type’s and built-in gunzip capabilities and improved auto .proto selection
Using Protobuf Decoder Burp Ext. (2/3)

POST /checkin HTTP/1.1
Content-type: application/x-protobuf
Accept-encoding: gzip
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.3; Nexus 7 Build/KTU84L)
Host: android.clients.google.com
Connection: Keep-Alive
Content-Length: 27

Failed to parse input.

POST /checkin HTTP/1.1
Content-type: application/x-protobuf
Accept-encoding: gzip
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.3; Nexus 7 Build/KTU84L)
Host: android.clients.google.com
Connection: Keep-Alive
Content-Length: 7

Failed to parse input.

```protobuf
def
1: ""
20: 0
```
Recovering and Using the .proto File(s)

- The easy way... searching through Google 😊
  - [android google] checkin protobuf ext:proto
  - GooglePlayDownloader (Tuxicicoman)
  - Android Checkin (nviennnot)

- Same three .proto files

<table>
<thead>
<tr>
<th>File name</th>
<th>Size</th>
<th>Revision</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkin.proto</td>
<td>1 kB</td>
<td>35</td>
<td>2014-06-11 22:09:21</td>
</tr>
<tr>
<td>logs.proto</td>
<td>1 kB</td>
<td>35</td>
<td>2014-06-11 22:09:21</td>
</tr>
</tbody>
</table>
Request: /checkin

- checkin.proto: message AndroidCheckinRequest {}

  ```protobuf
  message AndroidCheckinRequest {
    optional AndroidCheckinProto checkin = 4;
    optional int32 fragment = 20;
    ...
  }
  ```

- logs.proto: message AndroidCheckinProto {}

  ```protobuf
  message AndroidCheckinProto {
    optional AndroidBuildProto build = 1;
    ...
  }
  message AndroidBuildProto {
    optional string id = 1;
    ...
  }
  ```
Response: /checkin

- checkin.proto: message AndroidCheckinResponse {}

```protobuf
message AndroidCheckinResponse {
  optional bool statsOk = 1;
  optional int64 timeMsec = 3;
  ...
}
```

1: 1
3: 1424459701272

Stats recorded properly

Time from server (Java epoch)
Different Protobuf Versions

- Used to generate the library contents
  - Protobuf Decoder extension & Local system

### Burp Extensions

Extensions let you customize Burp's behavior using your own or third-party code.

<table>
<thead>
<tr>
<th>Add</th>
<th>Remove</th>
<th>Up</th>
<th>Down</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Loaded</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Python</td>
<td>Protobuf Decoder</td>
</tr>
</tbody>
</table>

```
Error importing proto! /root/Burp/proto_files/checkin.proto!
Traceback (most recent call last):
File "/root/Burp/bapps/bdb8c70d3f1b74679b2996e0d3d36e81a/protoburp.py", line 383, in importProtoFiles
    module = compile_and_import_proto(selectedFile)
File "/root/Burp/bapps/bdb8c70d3f1b74679b2996e0d3d36e81a/protoburp.py", line 471, in compile_and ImportProtoFiles
    return compile_and_import_proto(module)
File "/root/jython/jython-standalone-2.7-b3.jar/Lib/importlib/__init__.py", line 37, in import_module
    __import__(name)
File "/tmp/tmpWH1PfZ/checkin_pb2.py", line 9, in <module>
ImportError: cannot import name symbol_database
```
Load .proto Files (1-by-1 or All)

• Tweaking .proto files before loading them into the Protobuf Decoder extension
  – Dependencies (import statements)
Deserialize As...

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw</td>
<td>Params</td>
</tr>
<tr>
<td>Headers</td>
<td>Hex</td>
</tr>
<tr>
<td>Protobuf</td>
<td></td>
</tr>
</tbody>
</table>

```
stat {
  tag: ""
}
```

Default matching type - auto
(or select it manually)

```
checkin {
  build {
  
  }
}
fragment: 0
```

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw</td>
<td>Headers</td>
</tr>
<tr>
<td>Hex</td>
<td>Protobuf</td>
</tr>
</tbody>
</table>

```
statsOk: true
timeMsec: 1424437201284
```

Load .proto

Deserialize As...

- AndroidCheckinProto
- AndroidCheckinResponse
- DeviceConfigurationProto
- AndroidEventProto
- AndroidBuildProto
- AndroidStatisticProto
- AndroidCheckinRequest
- AndroidIntentProto
- GservicesSetting
Recovering the .proto files

• The hard way... reverse engineering
• Extract .proto files from Elf binaries (June 2012)
  – Using metadata added by protoc (serialized .proto file)
  – “Reverse-engineering of protobuf-based applications”
    • https://www.sysdream.com/reverse-engineering-protobuf-apps
• Extract .proto files from Android APKs (Sep 2012)
  – APKs use Micro-protobuf (vs. protoc, removes metadata)
  – Automated through Androguard (PoC) – androproto.py
    • https://github.com/egirault/googleplay-api/blob/master/androguard/androproto.py
  – “Reversing Google Play and Micro-Protobuf applications”
    • http://www.segmentationfault.fr/publications/reversing-google-play-and-micro-protobuf-applications/
Request: /checkin

- checkin.proto: message AndroidCheckinRequest {}

  message AndroidCheckinRequest {
    optional AndroidCheckinProto checkin = 4;
    optional int32 fragment = 20;
  }

- logs.proto: message AndroidCheckinProto {}

  message AndroidCheckinProto {
    optional AndroidBuildProto build = 1;
  }

  message AndroidBuildProto {
    optional string id = 1;
  }
Android 4.4.4 (Default) – GSF (1/4)

- GPS 4.3.23 & GSF 4.4.4-...
Android 4.4.4 (Default) – GSF (2/4)
Android 4.4.4 (Default) – GSF (3/4)
public Logs.AndroidBuildProto getBuild()
{
    return this.build;
}
Android 4.4.4

- GPS 6.5.99+

Use previous non-obfuscated old code from GSF or GPS 😊
Exploiting Protobufs

- Web-app pen-tests & research
- Mobile apps pen-tests & research
- Google ‘Security Reward Programs” (bug bounty)
  - Web apps & Mobile apps (January 2015)
- New Protobuf Decoder Burp Extension
  - https://github.com/dinosec
  - Installation guide at DinoSec Blog (*in ≈ one month*)

Second, also starting today, all mobile applications officially developed by Google on Google Play and iTunes will now be within the scope of the Vulnerability Reward Program.

CERTIFICATE PINNING
OTA Update Checks

• MitM HTTPS traffic
  – Does not work, even with an imported root CA in the TrustStore
    • It is using certificate pinning
      – https://android.clients.google.com
      – https://play.googleapis.com

• Hard-coded inside Android / GPS

Certificate Pinning. Android 4.4 detects and prevents the use of fraudulent Google certificates used in secure SSL/TLS communications.

https://source.android.com/devices/tech/security/enhancements/enhancements44.html
Traffic capture: Check for update
GPS Native Libraries (Future Research...)

# cd /data/data/com.google.android.gms
# ls -l
...
...

# ls -l /data/app-lib/com.google.android.gms-2

-rwxr-xr-x system  system  517592 2008-09-21 18:13  libAppDataSearch.so
-rwxr-xr-x system  system  452308 2008-09-21 17:13  libWhisper.so
-rwxr-xr-x system  system  13500  2008-09-21 18:13  libconscrypt_gmscore_jni.so
-rwxr-xr-x system  system  13496  2008-09-21 18:13  libgames_rtmp_jni.so
-rwxr-xr-x system  system  181616 2008-09-21 18:13  libgcastv2_base.so
-rwxr-xr-x system  system  194000 2008-09-21 18:13  libgcastv2_support.so
-rwxr-xr-x system  system  304752 2008-09-21 18:13  libgms-ocrclient.so
-rwxr-xr-x system  system  1626612 2008-09-21 18:13  libgmscore.so
-rwxr-xr-x system  system  34188  2008-09-21 18:13  libjgcastservice.so
-rwxr-xr-x system  system  1239520 2008-09-21 18:13  libsslwrapper_jni.so
Android Internals

GETTING THE CODE
How To Get APK Files? (1/2)

- Get list of installed packages
  ```shell
  $ pm list packages
  $ pm path com.google.android.gsf
  package:/system/priv-app/GoogleServicesFramework.apk
  ```

- Get APK file location by package name
  ```shell
  $ ls -ld /system/app
  drwxr-xr-x  root  root  2015-01-01 21:55 app
  $ ls -ld /system/priv-app
  drwxr-xr-x  root  root  2014-06-13 09:06 priv-app
  $ ls -ld /data/app
  drwxrwx--x  system system 2015-01-01 13:41 app
  ```

- Get APK (without root, if you know its name: /data)
  ```shell
  C:\> adb pull /system/priv-app/GoogleServicesFramework.apk
  ```
How To Get APK Files? (2/2)

• Or from: /data/system/packages.xml (as root)
  – <item name="com.google.android.gms.permission.CHECKIN_NOW"
     package="com.google.android.gms" protection="2" />
  – (GPS v4.3.x): /system/priv-app/PrebuiltGmsCore.apk

$ ls -ld /data/system
drwxrwxr-x system system 2015-01-01 21:20 system

$ ls -l /data/system
...  
-rw-rw---- system package_info 10827 ... packages.list
-rw-rw---- system system 219110 ... packages.xml

$ pm path com.google.android.gms
package:/data/app/com.google.android.gms-2.apk

C:\> adb pull /data/app/com.google.android.gms-2.apk
Source code or reverse engineer (RE)

CODE ANALYSIS
DEX2JAR...
jd-gui: Obfuscation
jd-gui: Decompilation Failed
jadx: Decompilation
Inspecting the GSF APK...

- 48h (172800000L ms)
  - com.google.android.gsf/.update.
    SystemUpdateService

```java
public static void processUpdateLock(Context paramContext, Intent paramIntent) {
    if (paramIntent == null) {
        return;
    }
    if (paramIntent.getBooleanExtra("nowisconvenient", false)) {
        sUpdatesLockedUntil = 0L;
        paramContext.startService(new Intent(paramContext, SystemUpdateService.class));
        return;
    }
    sUpdatesLockedUntil = 172800000L + paramIntent.getLongExtra("timestamp", 0L);
}
```
INSPECTING OTA UPDATE FILES
OTA Update Packages (Signed)

• OTA updates are verified (twice) using digital certificates
  – OTA update packages crypto signed using an RSA private key
    • /system/etc/security/otacerts.zip
      – Certificates whose public keys will be accepted
      – Package signed by the corresponding private key
    • By RecoverySystem.verifyPackage(...)
      – Before booting into recovery to actually apply the update
    • By the installer, after rebooting into the recovery image
      – It uses an embedded public key (“/res/keys”)

• Preventing OTA updates by renaming otacerts.zip
  – The OTA update package might be re-downloaded again... 😞
  # mount -o remount,rw /system
  # cd /system/etc/security
  # mv otacerts.zip otacerts.zip.freeze
OTA Update Package Signature (1/3)

• Trust: /system/etc/security/otacerts.zip
  – E.g. /vendor/google/security/hammerhead
    • releasekey.x509.pem (single root CA cert - PEM)

• OTA update package = custom ZIP file
  – Updated files & scripts to apply them (verifications)
  – /META-INF/com/android/otacert
    • OTA update signing certificate (PEM format)
  – Custom ZIP trailer (by Android signapk tool)

Android code signing
OTA Update Package Signature (2/3)

• Custom ZIP trailer
  – Comment size (2 bytes) + ZIP comment (null-terminated string, 18 bytes) + binary signature block + final 6-byte record (signature offset)

• Script to split OTA update package file
  – Input: OTA update package file (ZIP)
  – Output: signature trailer file & signed block file (ZIP)
  – OTA split script + openssl to verify it
OTA Update Package Signature (3/3)

```
$ ls -l
-rw-r--r-- 1 root root 2571501 Feb  3 16:03
c08cce45be6d759d6fd29480f01128b18f7d4a11.signed-hammerhead-KTU84P-from-KTU84M.c08cce45.zip
-rw-r----- 1 root root 1229 Feb  2 11:13 releasekey.x509.pem
-rwxr-xr-x 1 root root 1300 Feb  2 12:03 split_ota_update_file.py
```

```
$ ./split_ota_update_file.py
c08cce45be6d759d6fd29480f01128b18f7d4a11.signed-hammerhead-KTU84P-from-KTU84M.c08cce45.zip ota.signature ota.signed
File size: 2571501
Signature offset: 1352
Signature size: 1346
Comment length: 18
Comment: 'signed by SignApk' (+null)
Comment (hex): 7369676e6564206279205369676641706b00
```

```
$ openssl smime -verify -in ota.signature -inform DER -content
ota.signed -noverify releasekey.x509.pem > /dev/null
Verification successful
```
OTA Updates: Download & Verification

• Stored in the “/cache” partition
  – update.zip

• OTA update scripts
  – Verify versions match
    • Device and OS version & Update version
  – Verify SHA-1 hashes of files to be replaced

• Logs
  – last_install
Frozen: “Por primera vez en años...”
ANDROID UPDATE PROCESS
MitM Attack Tool

• Last year... (2014)
  – iOSTIA: “iOS Tengo Indicios de Actualizaciones”
    • iProxy (in videos)

• This year... (2015)
  – heladOS
    “hoy enfriamos las actualizaciones de OS”
  – Python MitM proxy
    • Manipulate Android OTA update downloads
Downloading OTA Updates

• OTA update download request: HTTP
  – Multiple HTTP redirections
• MitM HTTP traffic
  – Delay immediate future updates...
  – If HTTPS traffic is intercepted
    • ‘Check now’ is blocked (certificate pinning)
• Automated attacks
  – ‘User-Agent’ accurate fingerprinting

User-Agent: Android-Checkin/2.0 (hammerhead KTU84P); gzip
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.3; Nexus 7 Build/KTU84L)
User-Agent: Dalvik/2.0.0 (Linux; U; Android 4.4.4; Nexus 5 Build/KTU84P)
OTA Downgrades (1/2)

• Device is running 4.4.3
  – OTA update available: 4.4.4
  – What if device gets 4.4.2 (from 4.4)?

• Recovery image
  – “Installing system update...”: Error
OTA Downgrades (2/2)

• OTA update package (ZIP)
  – META-INF/com/google/android/updater-script
  – E.g. OTA update from 4.4.3 to 4.4.4
  • Device running 5.0.1

```sh
file_getprop("/system/build.prop", "ro.build.fingerprint") ==
"google/hammerhead/hammerhead:4.4.3/KTU84M/1158763:user/release-keys" ||
file_getprop("/system/build.prop", "ro.build.fingerprint") ==
"google/hammerhead/hammerhead:4.4.4/KTU84P/1227136:user/release-keys" ||
abort("Package expects build fingerprint of ... or ... ; this device has " +
getprop("ro.build.fingerprint") + ".");

getprop("ro.product.device") == "hammerhead" || abort("This package is for
"hammerhead" devices; this is a "" + getprop("ro.product.device") + ",");
```

```
# getprop ro.build.fingerprint
google/nakasi/grouper:4.4.3/KTU84L/1148727:user/release-keys
```
Android OTA Update Attacks

• OTA update freeze
  – Same (vulnerable) version

• OTA bad updates
  – E.g. Android 5.0.1 (LRX22C)

<table>
<thead>
<tr>
<th>VERSION</th>
<th>BUILD</th>
<th>BOOTLOADER</th>
<th>BASEBAND</th>
<th>IMAGES</th>
<th>DELTAS</th>
<th>NOTES</th>
</tr>
</thead>
</table>
| 5.0.1   | LRX22C | HHZ12d     | 2.0.50.2.22 | Images | LRX22C from KTU84P
          |        |            |          |        | LRX22C from KTU84O
          |        |            |          |        | LRX22C from LRX210 (#1)
          |        |            |          |        | LRX22C from LRX210 (#2)
          |        |            |          |        | LRX22C from LRX210 (#3) |

5. Three versions of the LRX22C from LRX210 OTA exist, presumably due to people getting failures while updating because of the baseband. The one most likely to flash for anyone is #3.

The Decalogue

SOLUTIONS & CONCLUSIONS
Solution(S)

https://
El refranero español 🇪🇸

“Cuando las tecnologías de tu vecino veas peligrar...
...pon las tuyas a actualizar”
The Update Decalogue (1/2)

• Checking for updates
  – Trusted communication
    • Authenticated and encrypted
      – HTTPS + Certificate pinning
    – Verify integrity of contents
      • Timestamps
    – Verify time references
    – Register update for verification (next step)
  – Anti-DoS
    • If response not received in 10 days, e.g. suggest user to check manually
The Update Decalogue (2/2)

• Downloading and applying updates
  – Trusted communication
    • Authenticated and encrypted
      – HTTPS + Certificate pinning
  – Verify integrity of contents
    • Timestamps
  – Verify time references
  – Authorize update / downgrade (optional)
  – Verify integrity between check & download
    • Different versions? Downgrade?
  – Verify integrity between download & apply
    • Different versions? Downgrade?
  – Anti-DoS
    • If download not obtained in 10 days, e.g. suggest user to download manually
Credits

- Produced by: Raúl Siles

- Directed by: Mónica Salas
- E & E

- Casting by: Google

- Tools: Marcin

- Music by: Siletes

- Costume Designer: DinoSec
Questions?