iSECpartners
part of nccgroup

Multi-User Mode Android

Exploring The Android 4.2.x Security Model

Jesse Burns

Confidence 2013

May, 29th Krakow ,Poland

Agenda

iSECpartners part of nccgroup

- Jelly Bean & It's Features
- Claiming and Disclaiming Multi-User Security
- Protections
  - What
  - How
  - Example improvements

iSECpartners
part of nccgroup

Jelly Bean and It's Features

- Android has weird naming
  - Names, Numbers and API Levels

Version Name	Numbers	API
Cupcake	1.5	3
Donut	1.6	4
Éclair	2.0	5
Éclair	2.0.1	6
Éclair	2.1	7
Froyo	2.2.x	8
Gingerbread	2.3	9

Version Name	Numbers	API
Gingerbread	2.3.3	10
Honeycomb	3.0	11
Honeycomb	3.1	12
Honeycomb	3.2	13
Ice Cream Sandwich	4.0.1	14
Ice Cream Sandwich	4.0.3	15
Jelly Bean	4.1	16
Jelly Bean	4.2	17

iSECpartners
part of nccgroup

Jelly Bean – 4.2 – API 17

- Not a new "version name"
- New features: especially security features

Application verification, More control of premium SMS, Always-on VPN, Certificate Pinning, Improved display of Android permissions, installd hardening, init script hardening, FORTIFY_SOURCE, ContentProvider default configuration, Cryptography, Security Fixes…

iSECpartners
part of nccgroup

Jelly Bean – 4.2 – API 17

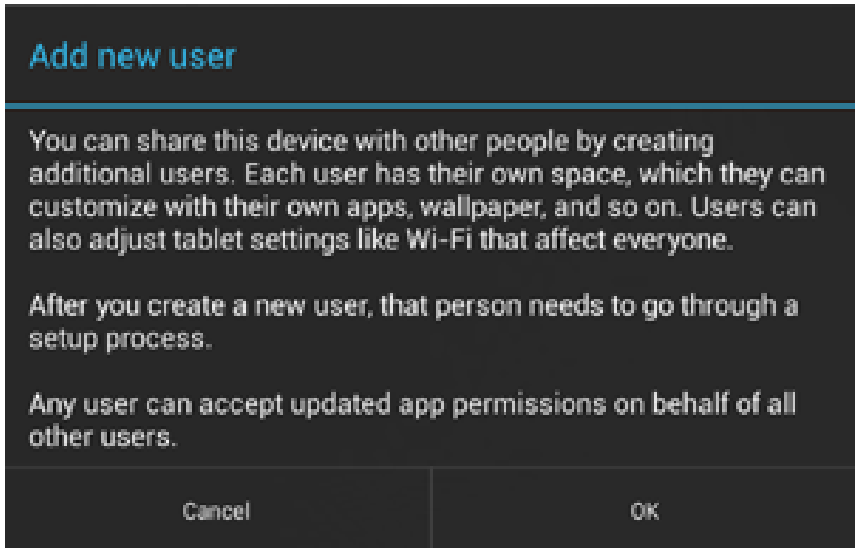
- Big Feature: **Multi-User Support** for tablets!
- Sounds security related!
- User's have different names, passwords, data
- New concept of an device "owner"
- Lots of security stuff under the covers
- Death to "World Read and Writeable"

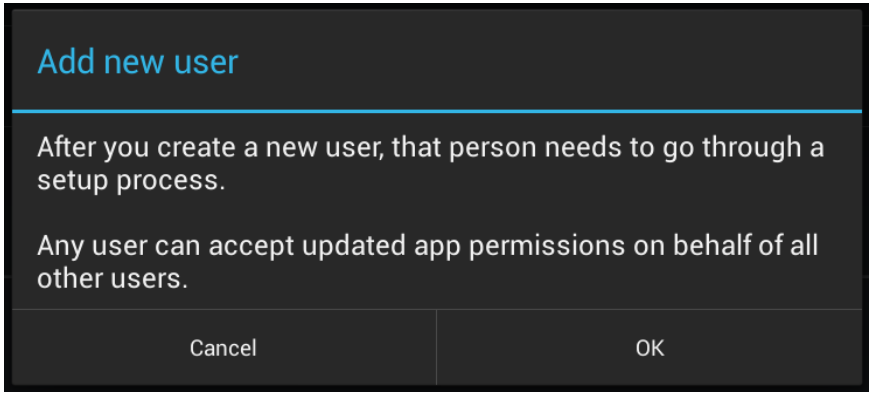
iSECpartners
part of nccgroup

Disclaiming Security

- This sounds like a security feature until you go to use it…

"As with any computer, you should share this tablet only with people you trust."





iSECpartners
part of nccgroup

Claiming Security

"No matter which of these APIs you use to save data for a given user, the data will not be accessible while running as a different user. From your app's point of view, each user is running on a completely separate device."

"As an app developer, there's nothing different you need to do in order for your app to work properly with multiple users on a single device."

http://developer.android.com/about/versions/android-4.2.html#MultipleUsers

iSECpartners
part of nccgroup

So which is it?

- Confusing – is this a security barrier?
- Till I mentioned a few problems
  - Fixed things I reported, wanted more
  - Clearly part of their security model
  - I didn't notice the fixes for awhile – 2 day fix latency, in code only apparent when 4.2.2 was open sourced
- Guests trust owners, owners trust guests less
- Real, but tablet only for now

iSECpartners
part of nccgroup

What is protected

- Processes and UIDs – the model has changed!
- "Activity Manager" arbitrated IPC
- New Permissions, command parameters and APIs
- Storage spaces for apps, isolation with namespaces!
- Even "sdcard" (no 4.2.x devices with real sdcards)
- The "Owner", from guests - generally
- Certain System Settings
  - New read only "Global" settings
  - Apps loosing ability to write

iSECpartners
part of nccgroup

What isn't protected

- Anything that's device wide:
  - Calling or messaging on a phone (hence tablet only!)
  - System Logs: note 4.1 READ_LOGS protection upgrade
  - System services state, e.g. clipboard changes
  - Shared System Settings that weren't migrated
    - Configurations like Paired Bluetooth, WiFi, Unknown Sources
- The idea of an untrusted user now exists outside the lock screen
- Guests from owners

iSECpartners
part of nccgroup

Protections Explored UIDs

- Android Apps have their own UIDs
- This is now one UID per app, per user
- So each app on a device with 3 users, runs with 3 UIDs

```
u10_a18   1054  125   855956 40280 ffffffff 00000000 S com.android.launcher
root      1212  2     0      0     ffffffff 00000000 S kbase_event
u14_a18   1347  125   858640 42548 ffffffff 00000000 S com.android.launcher
```

iSECpartners
part of nccgroup

Protections Explored UIDs

- Comment explaining naming in bionic/libc/bionic/stubs.cpp

```
Translate a user/group name to the corresponding user/group id.

// u0_a1234 -> 0 * AID_USER + AID_APP + 1234
// u2_i1000 -> 2 * AID_USER + AID_ISOLATED_START + 1000
// u1_system -> 1 * AID_USER + android_ids['system']
// returns 0 and sets errno to ENOENT in case of error
```

- Gist is – UID space divided into ranges

Protections Explored UIDs

- Android already used UID / GID ranges
- Service Isolation (new from 4.1) extended this
  - When android:isolatedProcess="true"
  - Runs with unique, unprivileged UID 99000 to 99999.

system/core/include/private/android_filesystem_config.h

```
AID_ISOLATED_START 99000
/* start of uids for fully isolated sandboxed processes */
AID_ISOLATED_END   99999
/* end of uids for fully isolated sandboxed processes */
```

iSECpartners
part of nccgroup

Protections Explored UIDs

- System services – not per-user
- But "launcher" might run as:

  U0_a18, u10_a18, and u14_a18
- Three users 0, 10, and 14, one app – a18, three UIDs:
  - 10018, 1010018, 1410018
- Four GIDs – 10018, 1010018, 1410018 & 50018 all_a18
  - That ALL GID (secondary group) isn't used much
- Android already used UID / GID ranges

iSECpartners
part of nccgroup

Activity Manager arbitrated IPC

- New Perms: INTERACT_ACROSS_USERS_FULL & INTERACT_ACROSS_USERS & ACCESS_ALL_EXTERNAL_STORAGE & ACCESS_CONTENT_PROVIDERS_EXTERNALLY & MANAGE_USERS

- ProtectionLevels above Dangerous – all at least Signature – most available to shell.

iSECpartners part of nccgroup

Activity Manager arbitrated IPC

- Context has new methods:

sendBroadcastAsUser(Intent **intent**, UserHandle **user**);
sendBroadcastAsUser(Intent **intent**, UserHandle **user**, String receiverPermission);

- @hide methods too!

registerReceiverAsUser(BroadcastReceiver **receiver**, UserHandle user, IntentFilter filter, String broadcastPermission, Handler scheduler)

startActivityAsUser(Intent **intent**, UserHandle **user**)

startServiceAsUser(Intent **service**, UserHandle **user**)

stopServiceAsUser(Intent **service**, UserHandle **user**)

iSECpartners part of nccgroup

Protections Explored - Storage

- /data/users now contains per-user directories
  - 0 – the owner
  - 10, 11, etc. – the guests
- Each looks like /data/data (0 is just a symlink here)

```
drwxr-x--x u10_system u10_system        2013-03-29 23:37 android
drwxr-x--x u10_a36    u10_a36           2013-03-29 23:37 com.android.apps.tag
drwxr-x--x u10_a1     u10_a1            2013-03-29 23:37 com.android.backupconfirm
drwxr-x--x u10_bluetooth u10_bluetooth  2012-01-05 08:07 com.android.bluetooth
drwxr-x--x u10_a3     u10_a3            2013-03-29 23:39 com.android.browser
```

- … but owned by the users' apps UIDs
- Explains MODE_WORLD* deprecation

iSECpartners
part of nccgroup

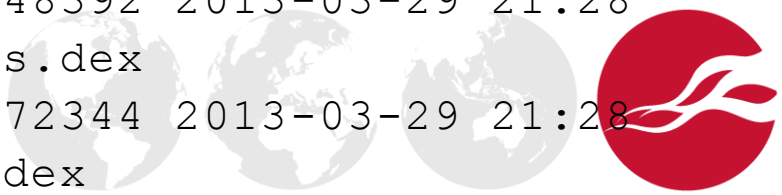
Protections Explored - Storage

- Applications are going to see their own local data, caches, databases etc! Without changing their code.
- This isolation is not namespace mount isolation though – other readable and writeable is accessible
- Unix permissions are the enforcement for app data
- Code like libraries still the same bytes / pages

iSECpartners
part of nccgroup

Protections Explored - Storage

```
-rw-r--r-- system   all_a41      60696 2013-03-29 23:40
data@app@jackpal.androidterm-1.apk@classes.dex
-rw-r--r-- system   all_a0       24592 2013-03-29 21:28
system@app@ApplicationsProvider.apk@classes.dex
-rw-r--r-- system   all_a1        9768 2013-03-29 21:28
system@app@BackupRestoreConfirmation.apk@classes.dex
-rw-r--r-- system   all_a2       16584 2013-03-29 21:28
system@app@BasicDreams.apk@classes.dex
-rw-r--r-- system   u0_a31002   566224 2013-03-29 21:28
system@app@Bluetooth.apk@classes.dex
-rw-r--r-- system   all_a3      912832 2013-03-29 21:28
system@app@Browser.apk@classes.dex
-rw-r--r-- system   all_a4      348392 2013-03-29 21:28
system@app@Calculator.apk@classes.dex
-rw-r--r-- system   all_a5      672344 2013-03-29 21:28
system@app@Calendar.apk@classes.dex
```



iSECpartners
part of nccgroup

External Storage Protections

- SD Card / External Storage is actually cooler
- Shared between apps – although permissions for read and write are being slowly retrofitted
- Zygote uses – Linux's Namespaces -- /sdcard
- Unshare() – gives per-process view of mounts
- Transparent to apps developers, each user has a unique view of the external storage

"Each user must have their own isolated primary external storage, and must not have access to the primary external storage of other users."

http://source.android.com/tech/storage/

iSECpartners
part of nccgroup

Protections Explored

- Processes and UIDs – the model has changed!
- External Storage isolation with namespaces!
- Even "sdcard" (any 4.2.x tablets with real sdcards?)
- New Permissions, command parameters and APIs
- The "Owner" – the first user
- Certain System Settings
  - New read only "Global" settings
  - Apps loosing ability to write

Trusting the Owner

iSECpartners part of nccgroup

- The owners of devices _own_ them
- They control the software, including the Trusted Code Base, they can root it, update it etc.
- They can physically mess with them, know the root decryption key, can access backups or whatever

- Users always trust the owner of a machine

Trusting the Owner

iSECpartners part of nccgroup

- Owners may activate USB debugging on their device
- Then they can see the system's logs, or run privileged commands like:

```
content query --uri
content://com.android.contacts/contacts/
--user 10
```

- This dumps other users contacts, note the "content" tool was updated for 4.2 to support users

Trusting the Guest

iSECpartners part of nccgroup

- On 4.2.1 – e.g. older unpatched Jelly Bean API 17
- Non-owners could enable USB debugging e.g. in a terminal (not an ADB shell)

```
am start -a android.settings
.APPLICATION_DEVELOPMENT_SETTINGS --user 10
```

- Then check the debugging, or set a backup password
- That's because this system UI is trusted
- Fixed in 4.2.2, the system doesn't want to trust guests

Modifying a System Service

iSECpartners part of nccgroup

- System Services needed tweaks for Multi-User
- ClipboardService – added:

```
private PerUserClipboard getClipboard() {
    return getClipboard(UserId.getCallingUserId());
}
private PerUserClipboard getClipboard(int userId) {
    synchronized (mClipboards) {
        Slog.i(TAG, "Got clipboard for user=" + userId);
        PerUserClipboard puc = mClipboards.get(userId);
        if (puc == null) {
            puc = new PerUserClipboard(userId);
            mClipboards.put(userId, puc);
        }
        return puc;
...
```

WiFi and Bluetooth

iSECpartners part of nccgroup

- Networking configuration is shared
- Not all devices automatically activate
- E.g.
  - Bluetooth speakers
    - Paired as guest
    - Work as owner
  - Bluetooth keyboard
    - Paired as guest
    - Visible to owner, but not auto-connecting

Settings Changes

iSECpartners part of nccgroup

- Settings.Secure and Settings.System
- Device wide settings are now R/O Global

- Programmatically changing not allowed
- UI based tweaks – like Settings.Secure – sometimes

iSECpartners
part of nccgroup

Thanks for coming!

Questions?

iSECpartners
part of nccgroup

UK Offices
Manchester - Head Office
Cheltenham
Edinburgh
Leatherhead
London
Thame

North American Offices
San Francisco
Atlanta
New York
Seattle

Australian Offices
Sydney

European Offices
Amsterdam - Netherlands
Munich – Germany
Zurich - Switzerland