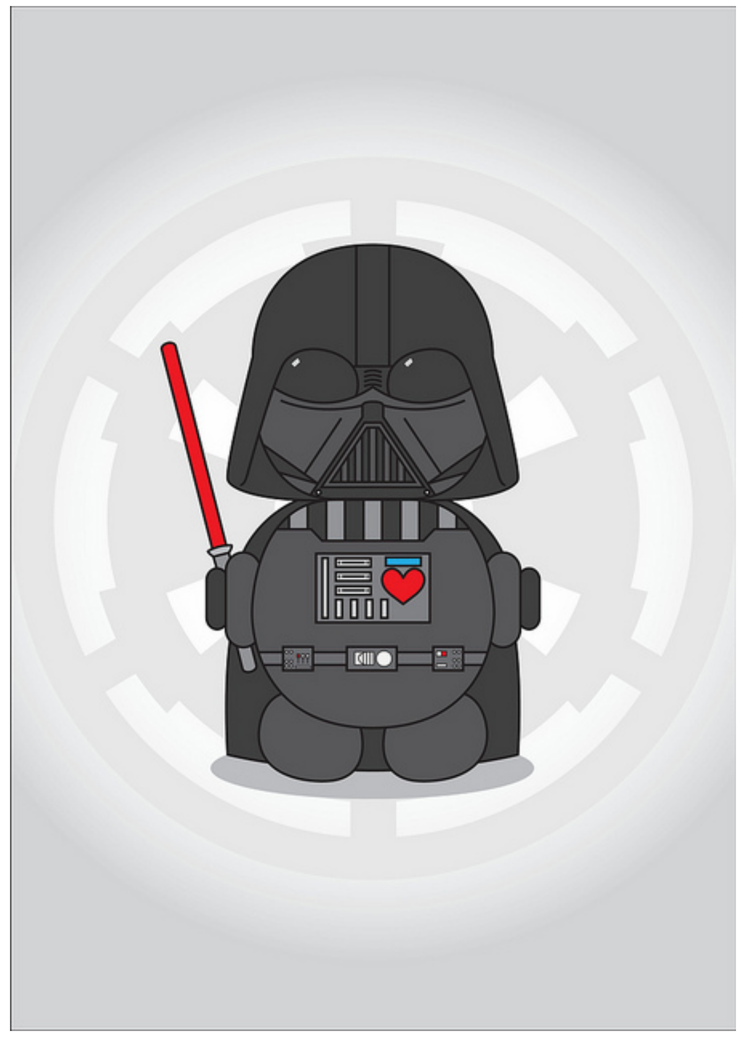
Tales from the dark side: developing SDKs at scale

DroidCon Berlin, September 2017

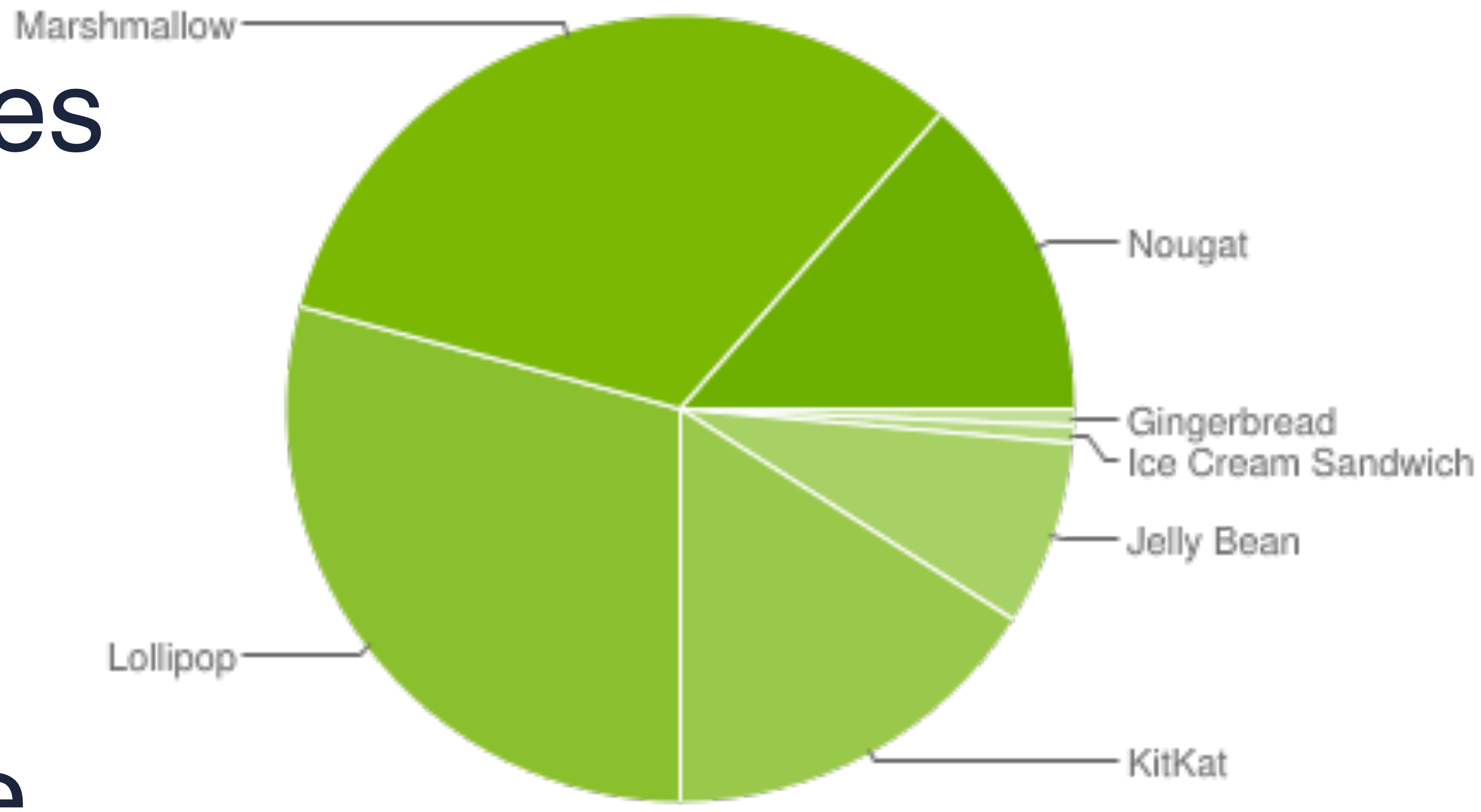
Kenneth Geisshirt
@kgeisshirt



http://jennandtonybot.blogspot.de/

A bit about Android statistics

- Pre-5.0 (API < 21): 25 % of active devices (August 2017)
- Android 4.3 and earlier: unsupported
- Android 4.4.4 receives security updates
  - vendors rarely update
- Your users are using old devices - unlike you 😱
- You will see bug reports from older devices



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

Bugs to cover?

- Cannot load `.so` file
- Encryption is failing
- Can’t find app’s directory
- Random crashes

Only seen on selected devices or under rare conditions

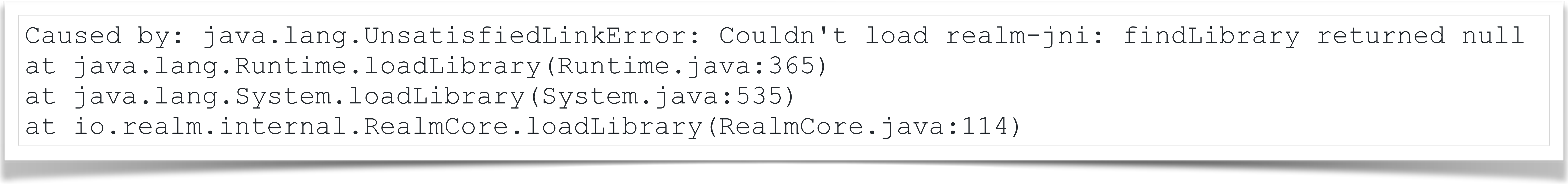
Cannot load `.so` file

The joy of native code

- C++ code is compiled and linked into shared objects (`.so` files)
- Loading is done by `Realm.init()`
- Loaded by app by calling `System.loadLibrary()`
- An APK contains `.so` files for all supported architectures
- During installation, only architecture specific `.so` files are copied

Loading `.so` files isn’t trivial

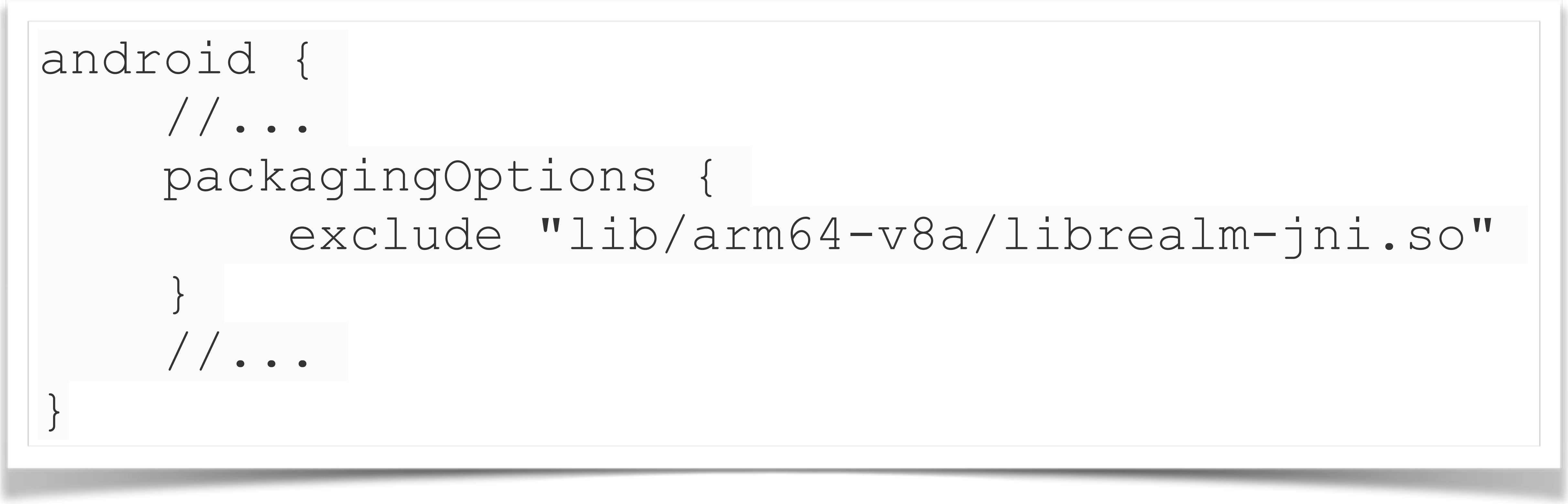
- Realm Java issue #1534 (October 2015)
- Android’s PackageManager will not always install `.so` files!
- **Solution**: use ReLinker (https://github.com/KeepSafe/ReLinker)
- All credit goes to KeepSafe for contributing ReLinker



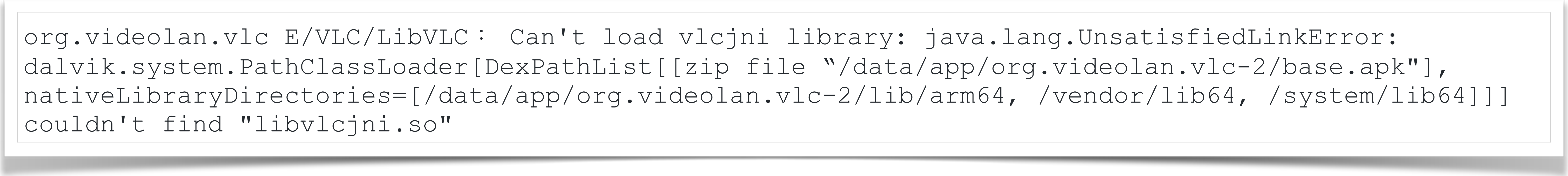
```
Caused by: java.lang.UnsatisfiedLinkError: Couldn't load realm-jni: findLibrary returned null
at java.lang.Runtime.loadLibrary(Runtime.java:365)
at java.lang.System.loadLibrary(System.java:535)
at io.realm.internal.RealmCore.loadLibrary(RealmCore.java:114)
```

More .so issues

- Realm Java issue #1640 (October 2015)
- Mixing 32 bit and 64 bit will not work
- **Solution**: exclude 64 bit Realm
- Known trouble-makers
  - Parallel Space, RenderScript, Unity3D



```
android {
    //...
    packagingOptions {
        exclude "lib/arm64-v8a/librealm-jni.so"
    }
    //...
}
```

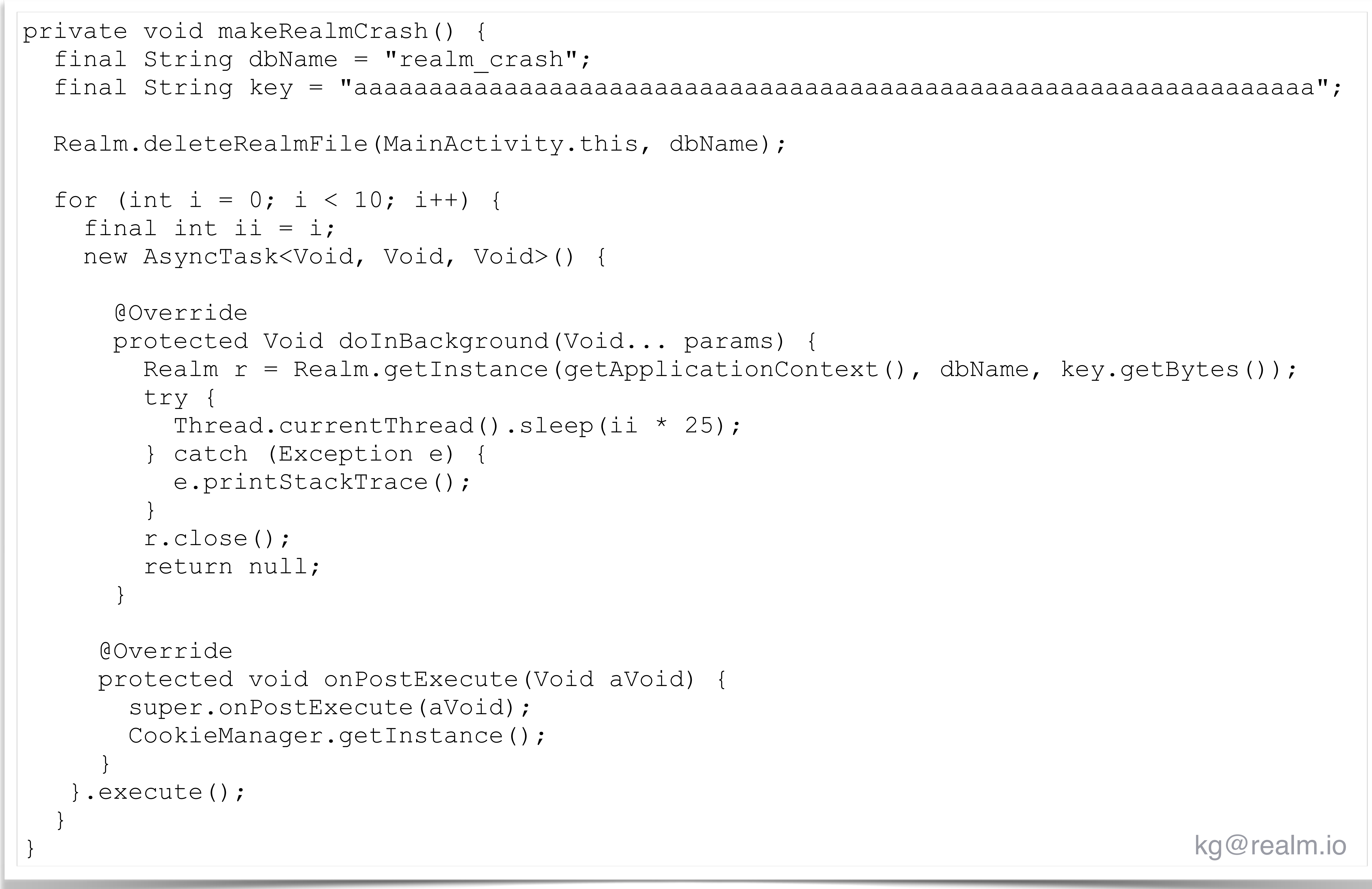


```
org.videolan.vlc E/VLC/LibVLC: Can't load vlcjni library: java.lang.UnsatisfiedLinkError:
dalvik.system.PathClassLoader[DexPathList[[zip file "/data/app/org.videolan.vlc-2/base.apk"],
nativeLibraryDirectories=[/data/app/org.videolan.vlc-2/lib/arm64, /vendor/lib64, /system/lib64]]]
couldn't find "libvlcjni.so"
```

Encryption is failing

Encryption is failing

- Realm Java issue #1008 (April 2015)
- Signal 11 (segmentation fault)
- CookieManager + encrypted Realms
- Affects Android 5.0.2 and 5.1
- Realm’s first encryption implementation was using signals



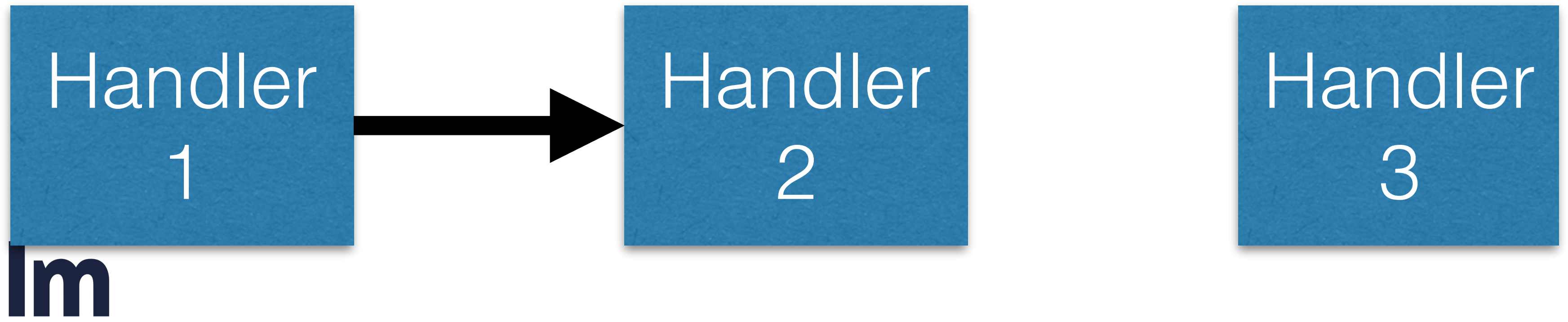
```
private void makeRealmCrash() {
  final String dbName = "realm_crash";
  final String key = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";

  Realm.deleteRealmFile(MainActivity.this, dbName);

  for (int i = 0; i < 10; i++) {
    final int ii = i;
    new AsyncTask<Void, Void, Void>() {

      @Override
      protected Void doInBackground(Void... params) {
        Realm r = Realm.getInstance(getApplicationContext(), dbName, key.getBytes());
        try {
          Thread.currentThread().sleep(ii * 25);
        } catch (Exception e) {
          e.printStackTrace();
        }
        r.close();
        return null;
      }

     @Override
     protected void onPostExecute(Void aVoid) {
       super.onPostExecute(aVoid);
       CookieManager.getInstance();
     }
   }.execute();
  }
}
```

realm

kg@realm.io

Using `signal(3)` requires discipline

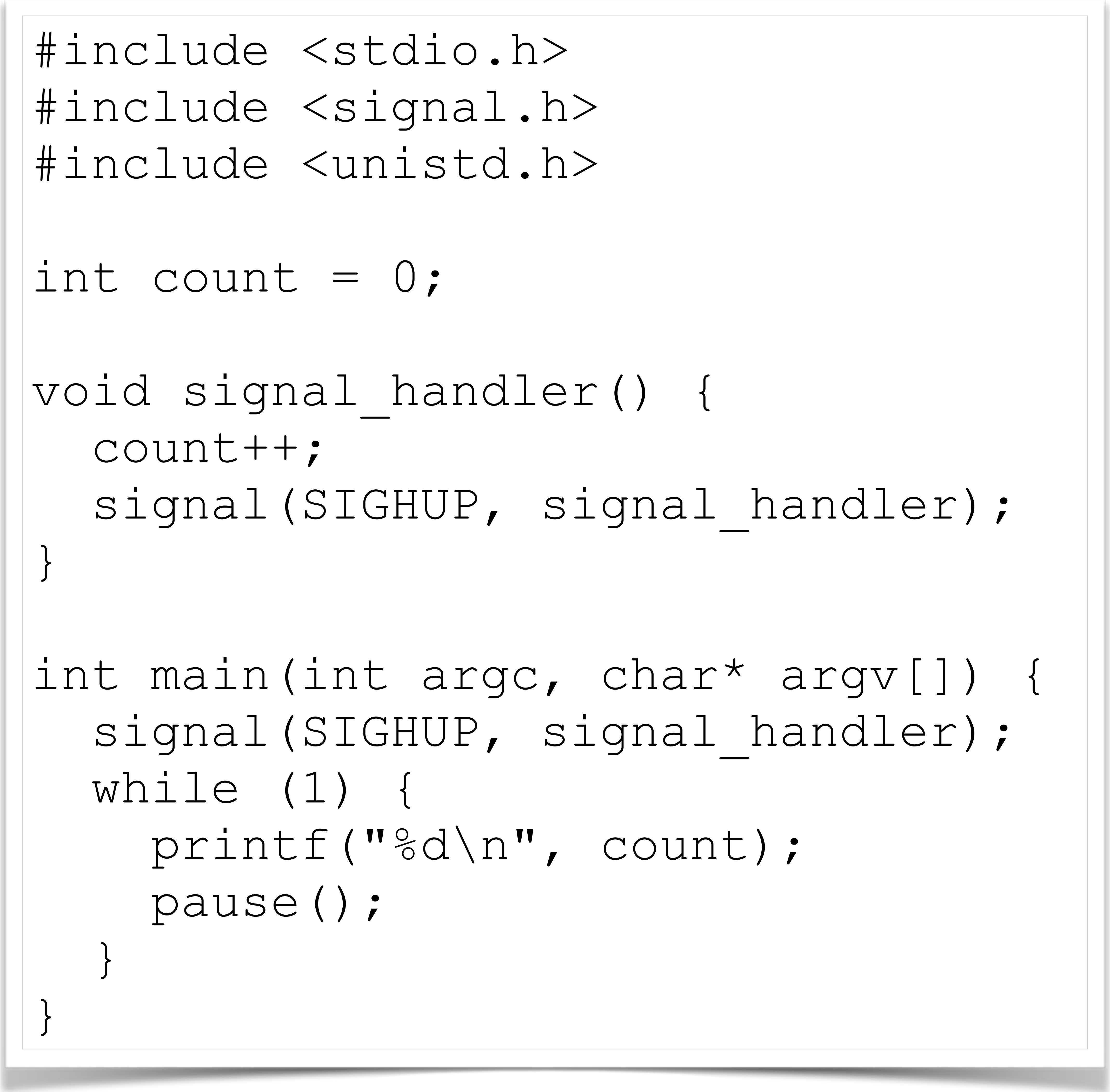
- Signal handler must pass on signals
- WebView 40 does not!
  - https://bugs.chromium.org/p/chromium/issues/detail?id=476831
  - Fixed by Google within 1 month
- Not passing on the signal, Realm ended up with corrupted memory
- We rewrote our encryption layer



Handler 1

Handler 2

Handler 3

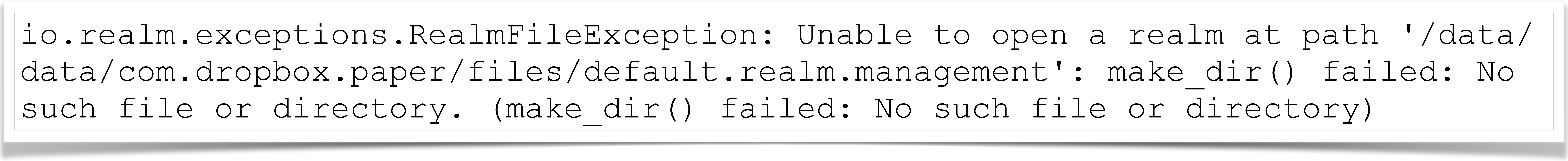


```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>

int count = 0;

void signal_handler() {
  count++;
  signal(SIGHUP, signal_handler);
}

int main(int argc, char* argv[]) {
  signal(SIGHUP, signal_handler);
  while (1) {
    printf("%d\n", count);
    pause();
  }
}
```

Cannot find app’s directory

When creating a Realm fails

- Realm Java issue #4493 (April 2017)
- Cannot create a Realm file: `make_dir()` is failing
- Sometimes `Context.getFilesDir()` returns `null`!



```
io.realm.exceptions.RealmFileException: Unable to open a realm at path '/data/
data/com.dropbox.paper/files/default.realm.management': make_dir() failed: No
such file or directory. (make_dir() failed: No such file or directory)
```

Known bug
and how to work around it

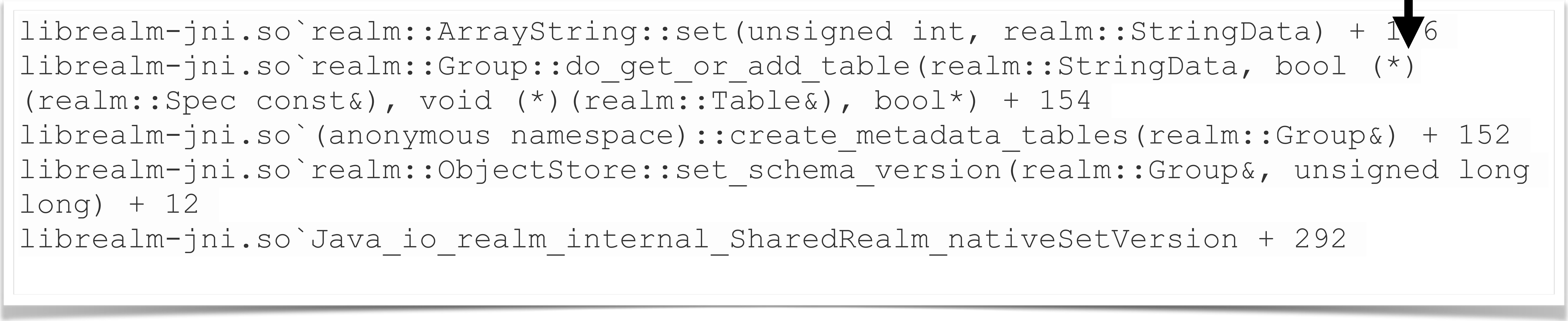
- Race condition in how directories/caches are created
- Bug fixed in Android 4.4
  - https://issuetracker.google.com/issues/36918154
  - June 2010 😈
- Realm’s work-around:
  - try creating directory multiple times (up to 200 ms = 12 frames)

Random crashes

A native crash

- Realm Java issue #3651(October 2016)
- Segmentation fault in `ArrayString::set()`

Remember this address



```
librealm-jni.so`realm::ArrayString::set(unsigned int, realm::StringData) + 1[illegible]6
librealm-jni.so`realm::Group::do_get_or_add_table(realm::StringData, bool (*)
(realm::Spec const&), void (*)(realm::Table&), bool*) + 154
librealm-jni.so`(anonymous namespace)::create_metadata_tables(realm::Group&) + 152
librealm-jni.so`realm::ObjectStore::set_schema_version(realm::Group&, unsigned long
long) + 12
librealm-jni.so`Java_io_realm_internal_SharedRealm_nativeSetVersion + 292
```

- `ArrayString::set()` is a key method within Realm Core - related to storing strings
- Highly tested method; Linux, OS X, iOS, Android, Windows

Reproducing the crash

- Impossible to reproduce using emulator or OnePlus One
- Affected device is Samsung Galaxy Tab 3 Lite (SM-T111)
- Managed to find and buy used model
- No unit tests fail!
- Only introExample (smallest possible demo app) could reproduce crash
- Limited debugging capabilities on Android NDK

Temporary fix

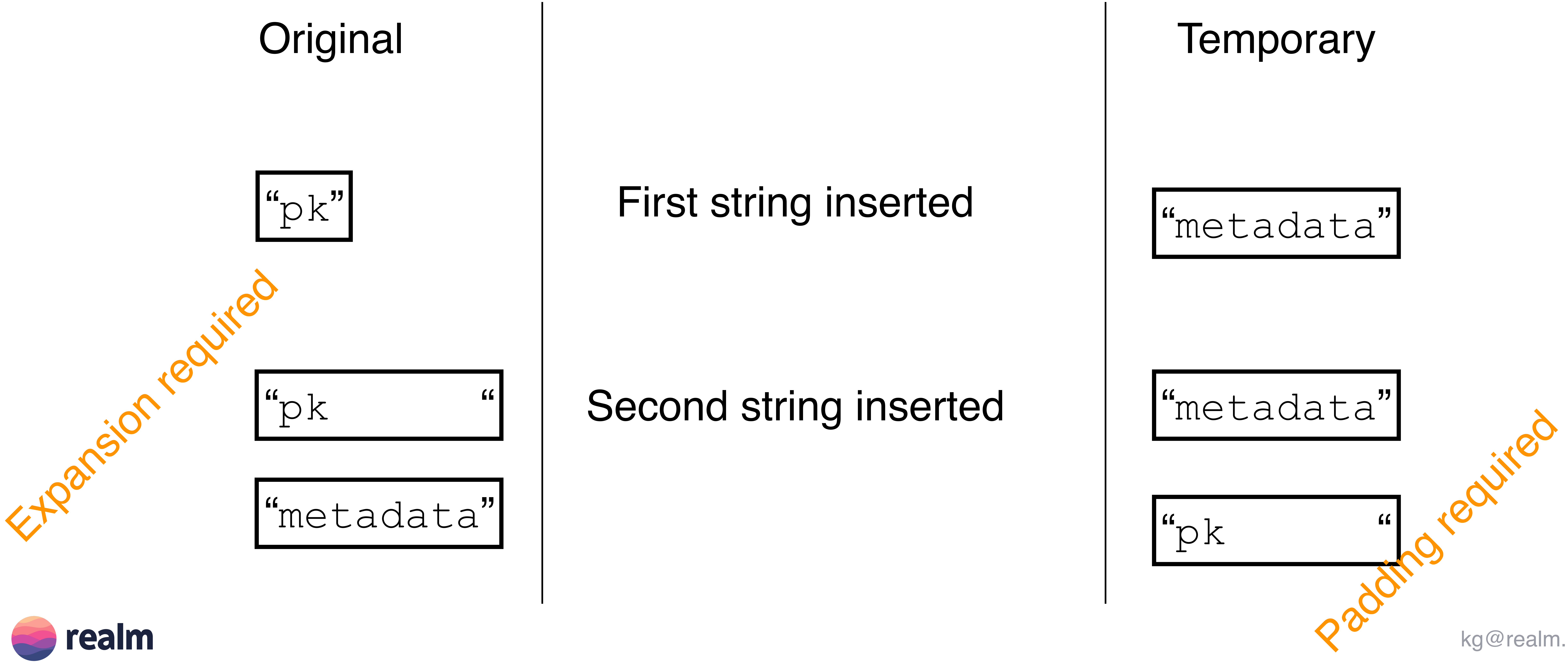
```
TableRef table = group.get_or_add_table(“pk”);
// adding columns and search index
table = group.get_or_add_table(“metadata”);
// adding columns and search index
```

↓

```
TableRef table = group.get_or_add_table(“metadata”);
// adding columns and search index
table = group.get_or_add_table(“pk”);
// adding columns and search index
```

- Stack trace includes `create_metadata_tables()`
- Called when Realm is created
- Only two strings are involved - first strings to be inserted

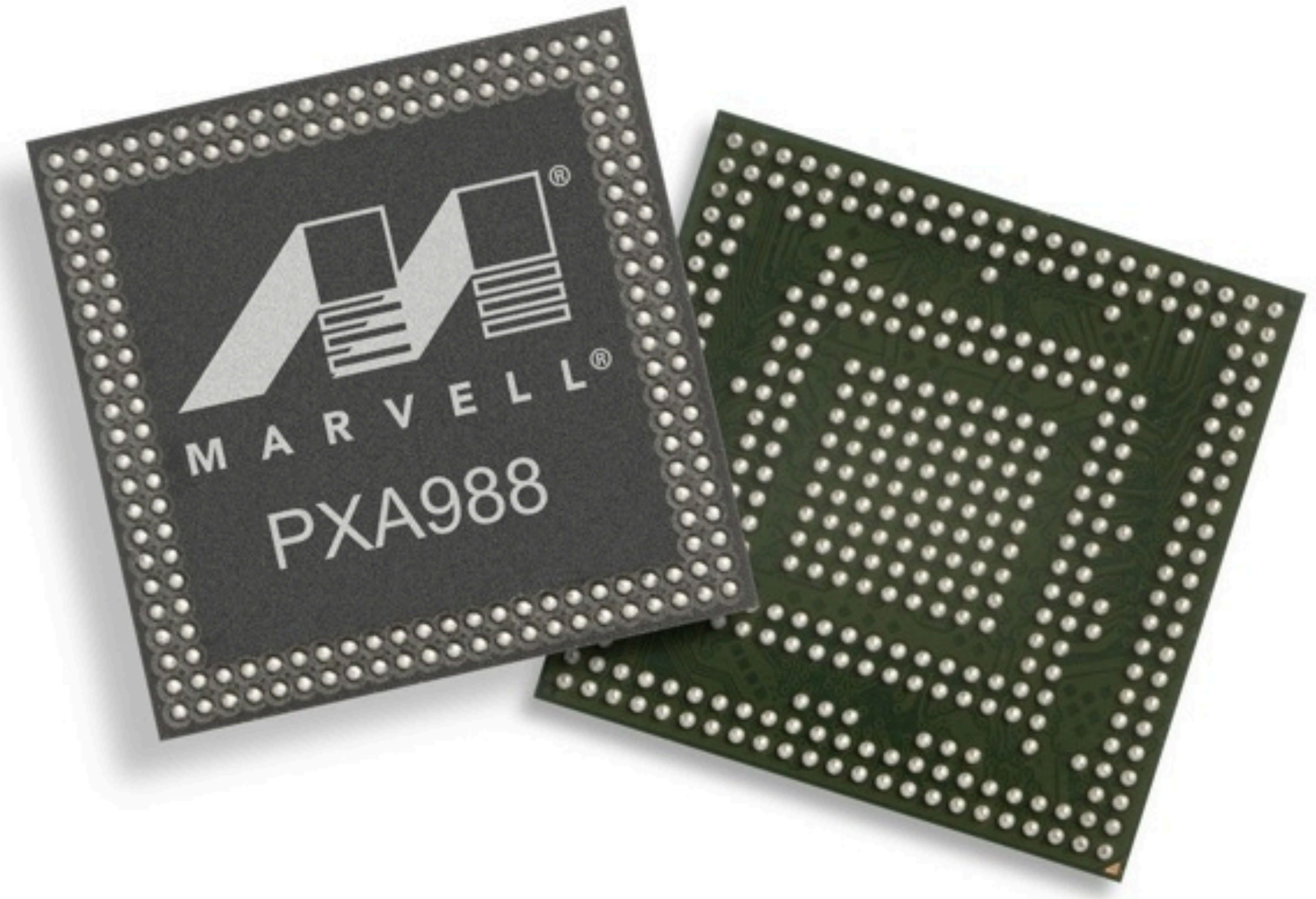
Insights from temporary fix



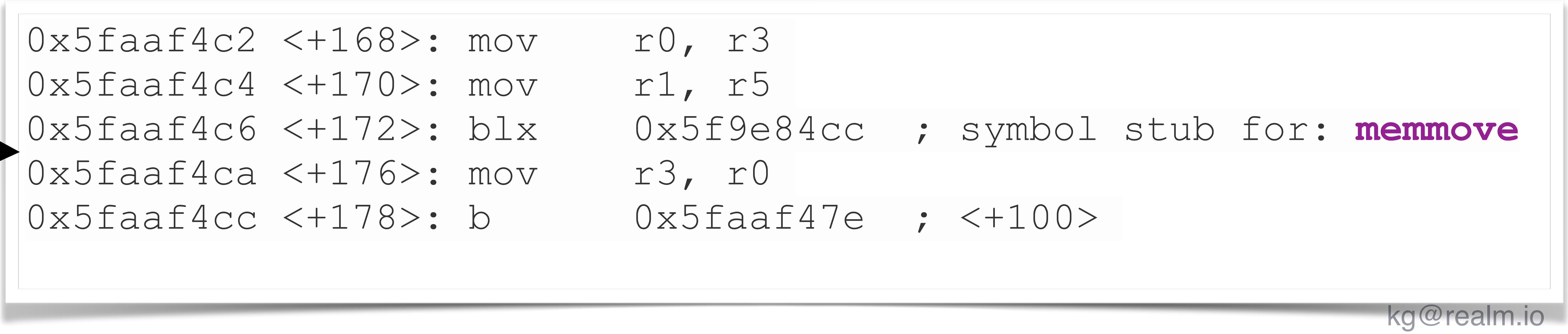
Original		Temporary
“pk”	First string inserted	“metadata”
“pk “ “metadata”	Second string inserted	“metadata” “pk “

Digging further

- Temporary fix helps in some cases
- Still segmentation fault in `ArrayString::set()`
- A lot of pointer arithmetics
- Calls to
  - `std::fill()` - padding
  - `std::copy_backward()` - expanding



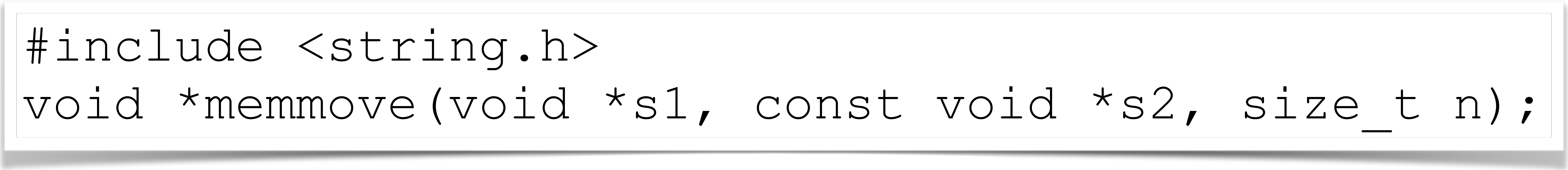
crash here
address 176



```
0x5faaf4c2 <+168>: mov    r0, r3
0x5faaf4c4 <+170>: mov    r1, r5
0x5faaf4c6 <+172>: blx    0x5f9e84cc   ; symbol stub for: memmove
0x5faaf4ca <+176>: mov    r3, r0
0x5faaf4cc <+178>: b      0x5faaf47e   ; <+100>
```

realm

memmove()



```
#include <string.h>
void *memmove(void *s1, const void *s2, size_t n);
```

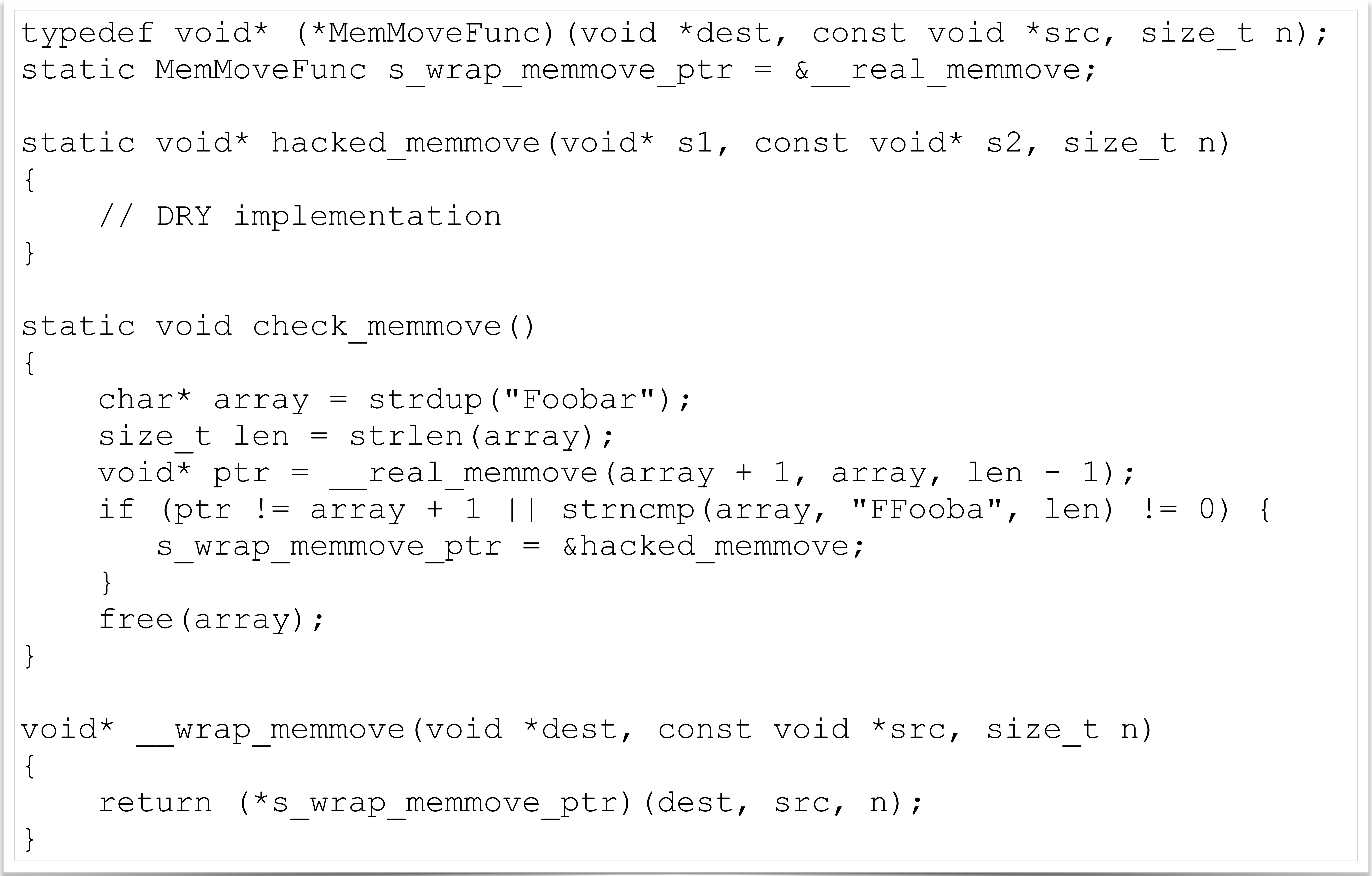
- Introduced in 4.3BSD Reno (1990)

- Bug in `memcpy()` found by ChengYi He (https://github.com/chengyihe)
- Bug reports at Qt and Unity3D
- Root cause: probably race condition in Linux kernel (http://lists.infradead.org/pipermail/linux-arm-kernel/2013-October/201893.html)

Ready for workaround

- Simple test case from Qt
- Rolling a new `memmove()`
- Using `memmove()` from D.R.Y. (https://github.com/dryc/libc11)
- Swapping gcc's builtin functions at link time
  - `-Wl,—wrap,memmove`

Blog post: https://academy.realm.io/posts/when-memmove-fails/



```
typedef void* (*MemMoveFunc)(void *dest, const void *src, size_t n);
static MemMoveFunc s_wrap_memmove_ptr = &__real_memmove;

static void* hacked_memmove(void* s1, const void* s2, size_t n)
{
    // DRY implementation
}

static void check_memmove()
{
    char* array = strdup("Foobar");
    size_t len = strlen(array);
    void* ptr = __real_memmove(array + 1, array, len - 1);
    if (ptr != array + 1 || strncmp(array, "FFooba", len) != 0) {
       s_wrap_memmove_ptr = &hacked_memmove;
    }
    free(array);
}

void* __wrap_memmove(void *dest, const void *src, size_t n)
{
    return (*s_wrap_memmove_ptr)(dest, src, n);
}
```

With enough users, your code will run on every Android version released.

You will be hit by old bugs.

Acknowledgements

- Analysing and testing `memcpy()`/`memmove()`: ChengYi He, GitHub user diegomontoya, and Jonas Bark
- Debugging `memmove()`: Finn Schiermer Andersen
- Wrapping `memmove()`: Mulong Chen
- Debugging and reimplementing encryption: Christian Melchior, Thomas Goyne, Mulong Chen
- Workaround for `getFilesDir()`: Christian Melchior