



MAIBORNWOLFF

Disclaimer:

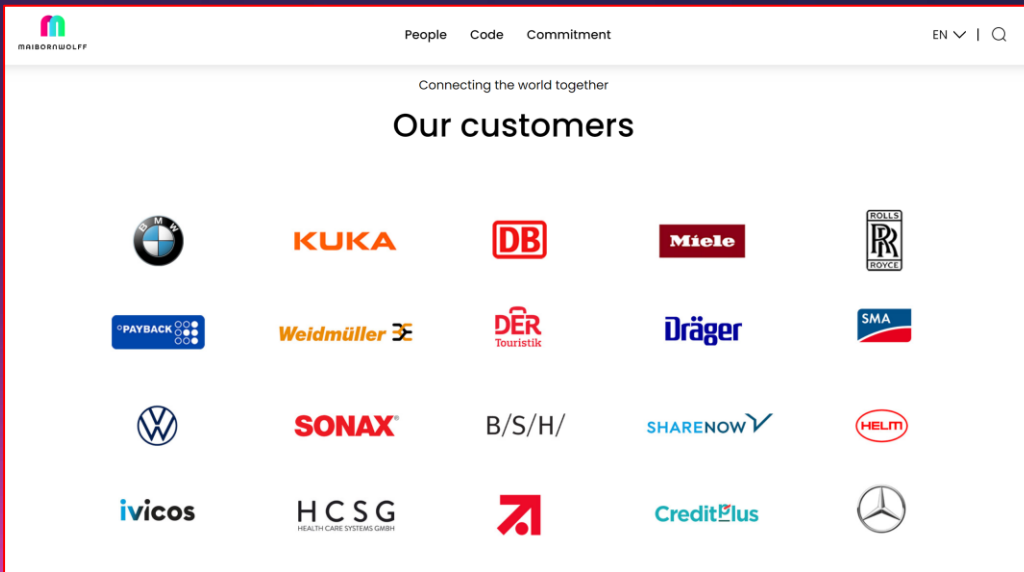
Please notice that any presented information may be used for legal and teaching/research purposes only. Please observe the laws and reengineering/interception is not legal in most countries without permission.

Dr. Nils T. Kannengießer



Senior Security Consultant
MaibornWolff GmbH, Augsburg

Consulting/Software Engineering company



Great Place To Work
12th year



Munich
Augsburg
Frankfurt
Darmstadt
Berlin
Hamburg
Tunis, Tunisia
Valencia, Spain



High investment in R&D



Different nationalities

Benefits include, e.g.

- Parental leave
- 30 days vacation
- Sabbatical (team approval)
- Home Office up to 100% (team approval)
- Flexible working time (team approval)
- Training budget 1.5x monthly salary (IT topics preferred, but also languages etc.)
- Lots of company events, e.g. sailing challenge Valencia to Tunis this fall
- ...

An Introduction into the Interception of TLS/HTTPS on Android

Android Reengineering

- Quick Intro -

Android Reengineering Basics

The smali language (assembly) is generated by a tool called apktool to allow easy modifications and recompilations.

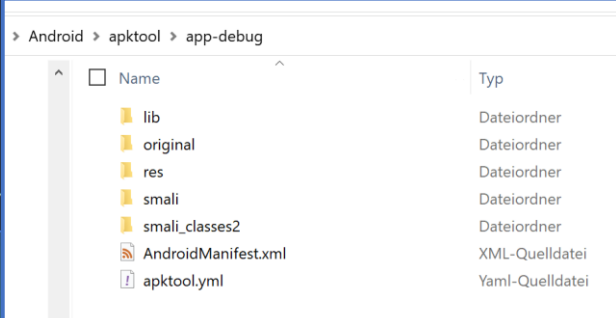
Example

Java:

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    Button button = (Button) findViewById(R.id.button);
```

smali:

```
.method protected onCreate(Landroid/os/Bundle;)V  
    .locals 3  
    .param p1, "savedInstanceState"    # Landroid/os/Bundle;  
  
    .line 20  
    invoke-super {p0, p1}, Landroidx/appcompat/app/AppCompatActivity;->onCreate(Landroid/os/Bundle;)V  
  
    .line 21  
    const v0, 0x7f0a001c  
  
    invoke-virtual {p0, v0}, Lcom/example/mw/MainActivity;->setContentView(I)V  
  
    .line 23  
    const v0, 0x7f070051  
  
    invoke-virtual {p0, v0}, Lcom/example/mw/MainActivity;->findViewById(I)Landroid/widget/Button;
```

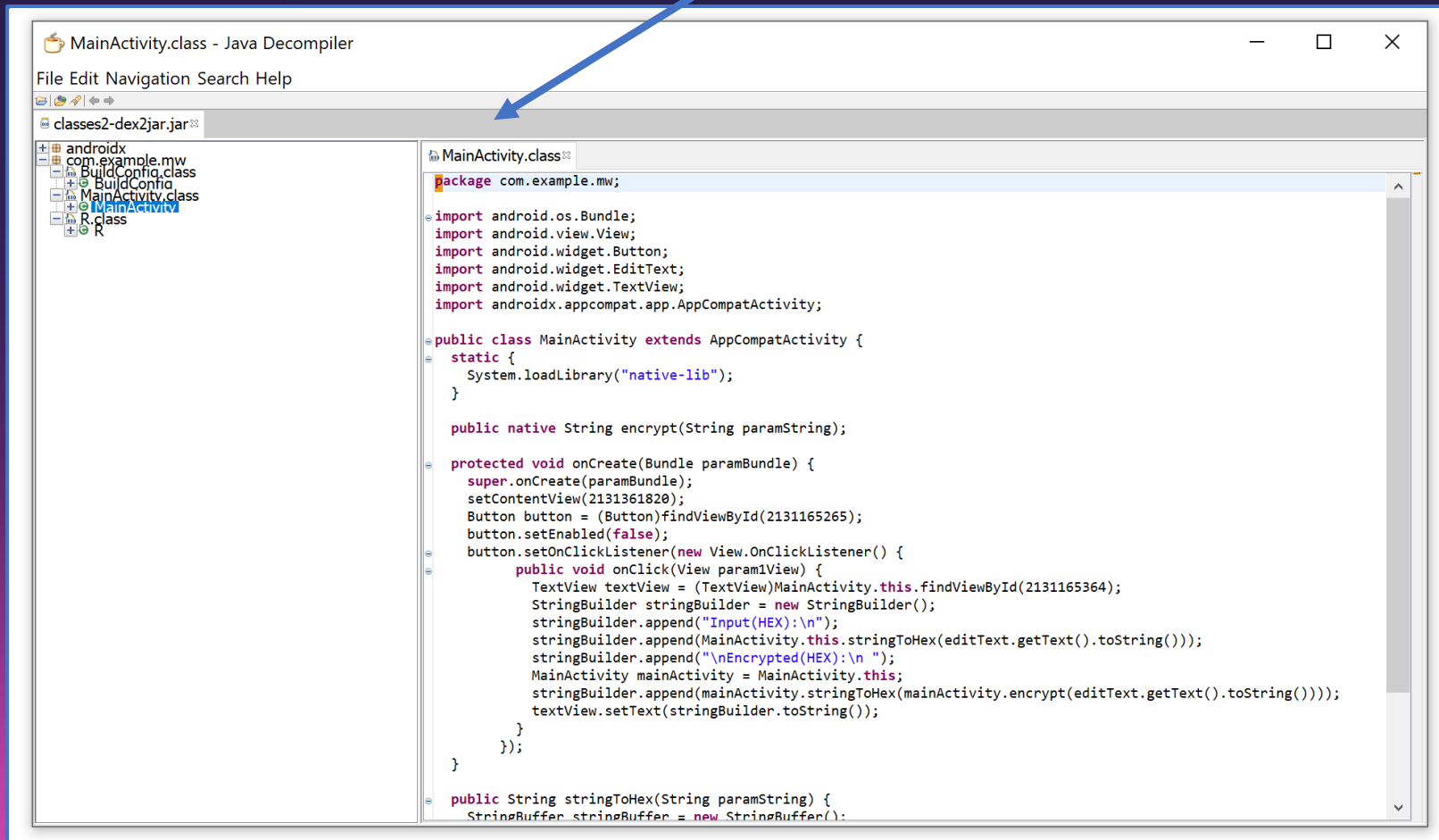


» Android » apktool » app-debug	
<input type="checkbox"/> Name	Type
lib	Dateiordner
original	Dateiordner
res	Dateiordner
smali	Dateiordner
smali_classes2	Dateiordner
AndroidManifest.xml	XML-Quelldatei
apktool.yml	Yaml-Quelldatei

Android Reengineering Basics

Besides using the apktool and smali, we can convert the file for the Dalvik/ART-VM from *.dex to *.jar to use any existing Java Decompiler.

For instance, JD-GUI ...



Android Reengineering Basics

Furthermore, we can decompile native code using IDA (\$\$\$) and NSA's Ghidra (free).



Ghidra graphic source: <https://ghidra-sre.org/>

```
1 #include <jni.h>
2 #include <stdio.h>
3 #include <iostream>
4 #include <android/log.h>
5
6 #define APPNAME "MyApp"
7
8 extern "C" JNIEXPORT jstring JNICALL
9
10 Java_com_example_mw_MainActivity_encrypt(
11     JNIEnv* env,
12     jobject obj,
13     jstring str
14 ) {
15     jsize len = env->GetStringUTFLength(str);
16     char *c_msg = nullptr;
17     c_msg = (char *) env->GetStringUTFChars(str, nullptr);
18
19     for (int i = 0; i < len; ++i) {
20         __android_log_print(ANDROID_LOG_VERBOSE, APPNAME, "The value %c", c_msg[i]);
21         c_msg[i] = (c_msg[i] ^ 'B');
22         __android_log_print(ANDROID_LOG_VERBOSE, APPNAME, "The XOR value %c", c_ms
23     }
24
25
26
```

docs.oracle.com/javase/7/docs/technotes/guides/jni/spec/functions.h...

getstringut 5/10

RETURNS:
Returns a Java string object, or NULL if the string cannot be constructed.

THROWS:
OutOfMemoryError: if the system runs out of memory.

GetStringUTFLength
jsize GetStringUTFLength(JNIEnv* env, jstring string);
Returns the length in bytes of the modified UTF-8 representation of a string.

LINKAGE:
Index 168 in the JNIEnv interface function table.

PARAMETERS:
env: the JNI interface pointer.
string: a Java string object.

RETURNS:
Returns the UTF-8 length of the string.

GetStringUTFChars
const char* GetStringUTFChars(JNIEnv* env, jstring string, jboolean* isCopy);
Returns a pointer to an array of bytes representing the string in modified UTF-8 encoding. This array is valid until it is released by ReleaseStringUTFChars().
If isCopy is not NULL, then *isCopy is set to JNI_TRUE if a copy is made; or it is set to JNI_FALSE if no copy is made.

```
Decompile: Java_com_example_mw_MainActivity_encrypt - (libnative-lib2.so)
1 void Java_com_example_mw_MainActivity_encrypt(JNIEnv *param_1,undefined4 param_2,_jstring *param_3)
2
3
4 {
5     int iVar1;
6     char *pcVar2;
7     undefined4 uVar3;
8     int local_28;
9
10    iVar1 = _JNIEnv::GetStringUTFLength(param_1,param_3);
11    pcVar2 = (char *)_JNIEnv::GetStringUTFChars(param_1,param_3,(uchar *)0x0);
12    local_28 = 0;
13    while (local_28 < iVar1) {
14        uVar3 = __android_log_print(2,"MyApp","The value %c",pcVar2[local_28]);
15        pcVar2[local_28] = pcVar2[local_28] ^ 0x4e;
16        __android_log_print(2,"MyApp","The XOR value %c",pcVar2[local_28],uVar3);
17        local_28 = local_28 + 1;
18    }
19    _JNIEnv::NewStringUTF(param_1,pcVar2);
20    return;
21 }
22
```

Notice: I used a different lib with key ,N' (0x4E) in Ghidra

Android Reengineering Basics

If this was too fast for you now, I'd like to pinpoint you towards a dedicated talk on it that I gave at the Technical University of Valencia last year (Spanish intro, English talk):



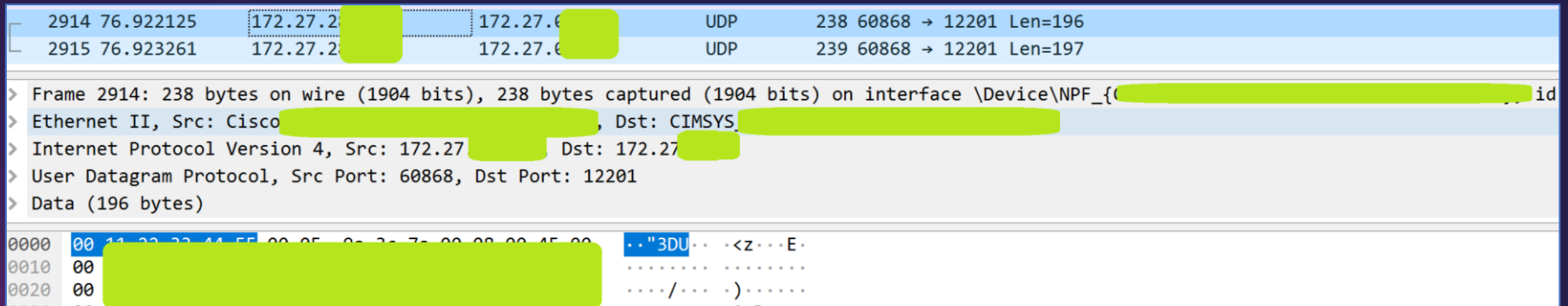
<https://www.youtube.com/watch?v=CsrBm0KbPsg>

Interception

... of secured channels

Interception - Intro

Interception is required quite often. My most recent case was the debugging of a reporting library. A handy tool is called **WireShark** and allows you to review communications, understand protocols, fix issues or use it for reengineering purposes.



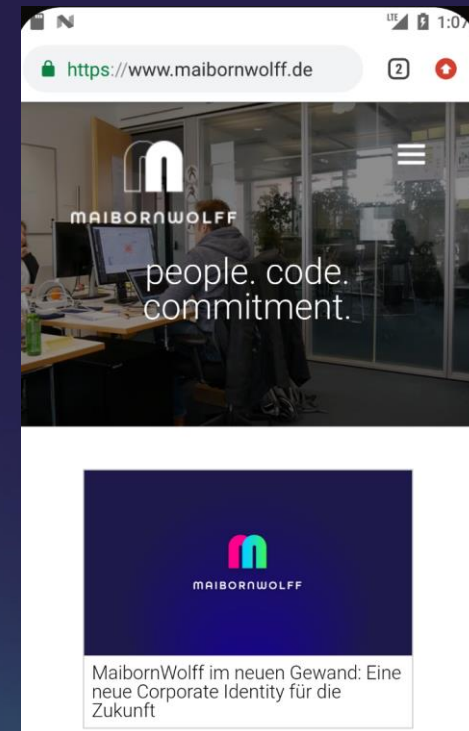
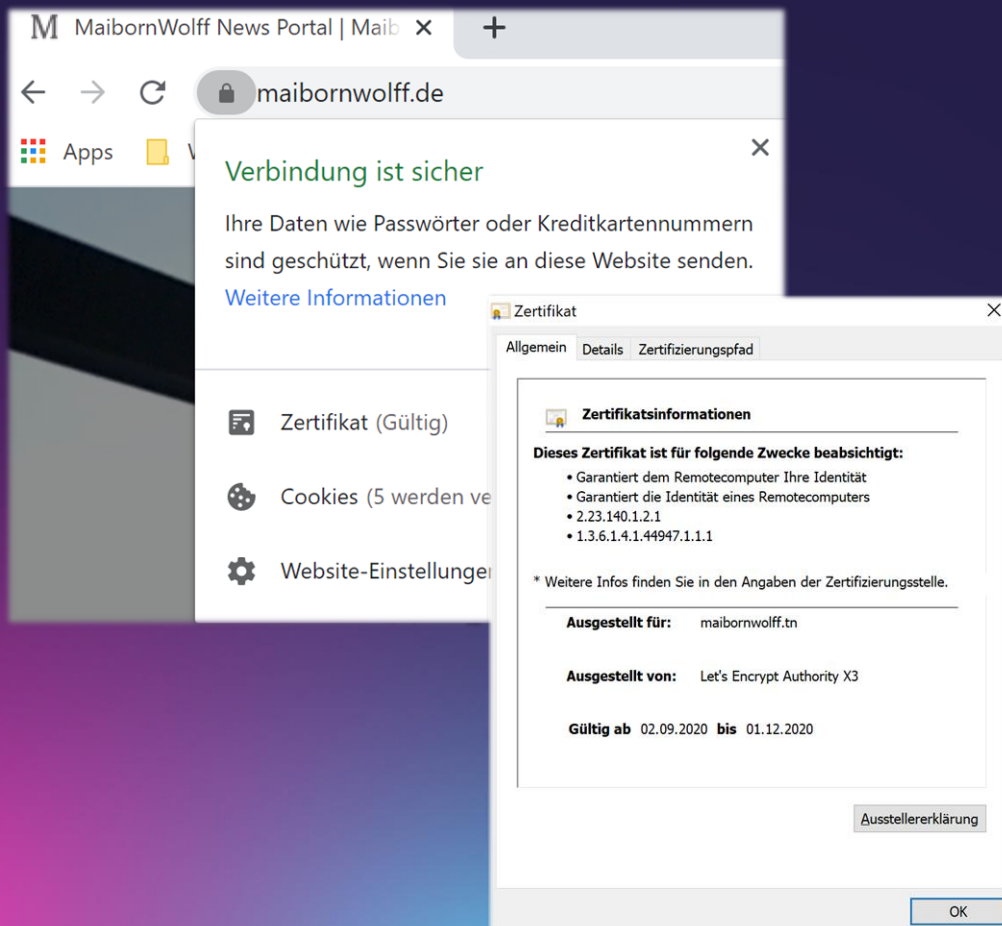
Here it was UDP and unencrypted.... But what about Android and encrypted connections?

Interception - Intro

In general, websites and communication can and should be encrypted. Nowadays, there are even free Certificate Authorities (CAs) like „Let's encrypt“ and it's the default.

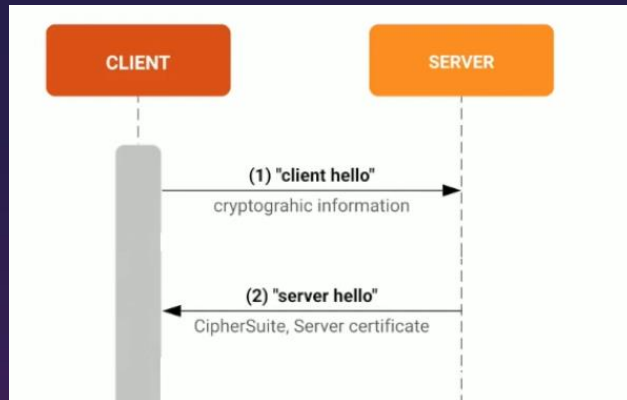


Graphic source: <https://letsencrypt.org/>

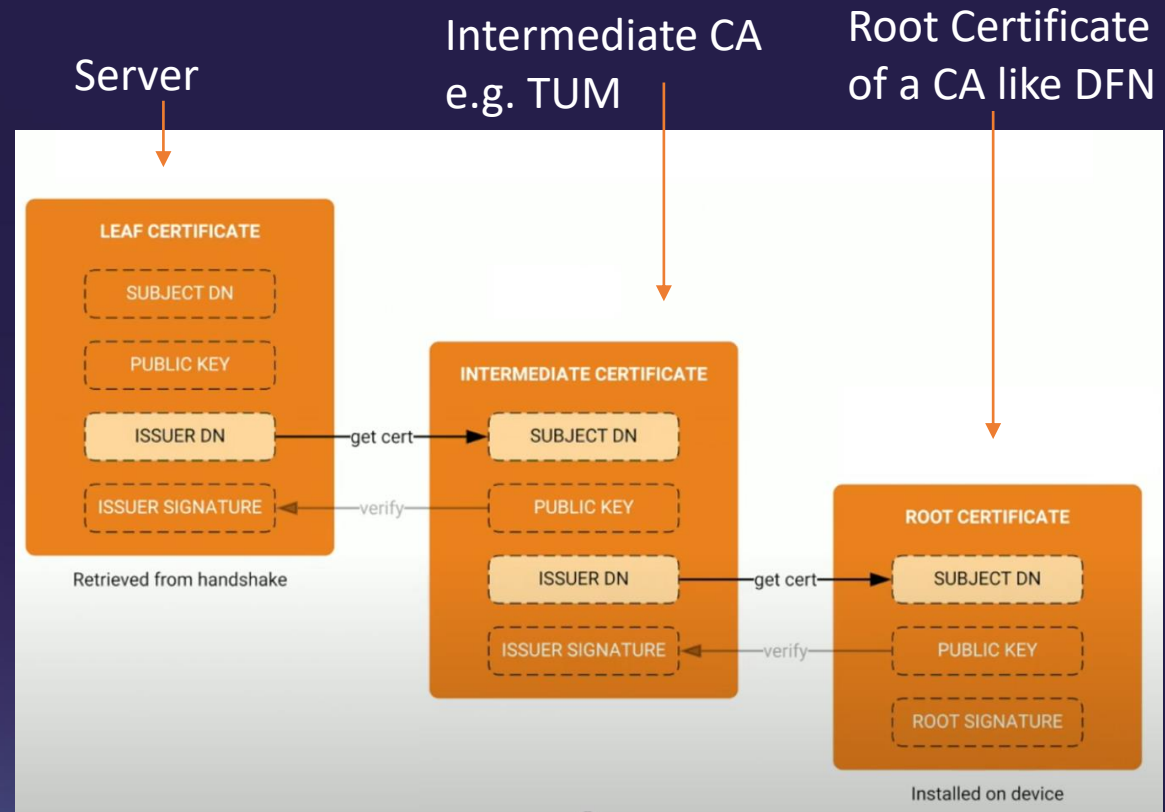


Interception – Background information

Let's take a quick look how that works...



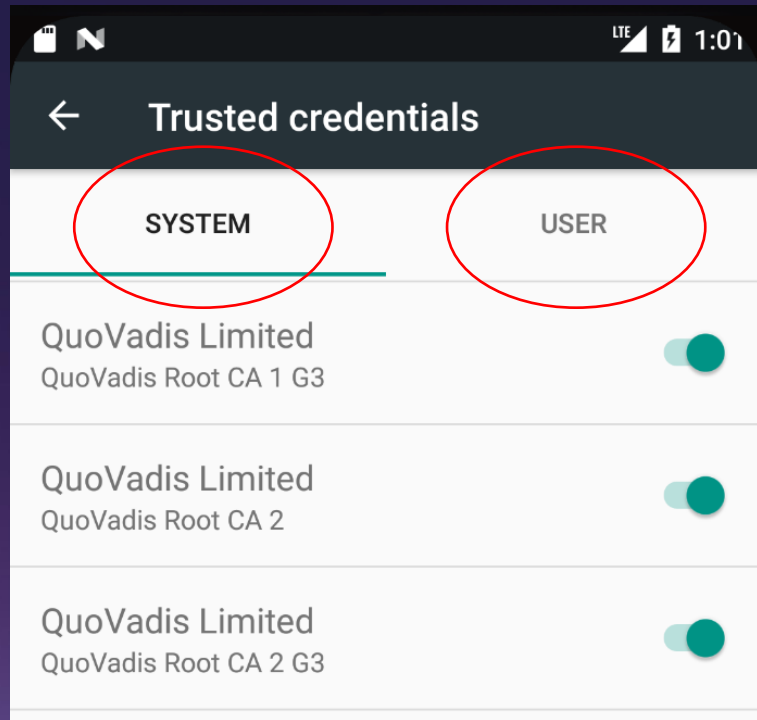
Verification?



Source of pictures and great introduction: <https://www.youtube.com/watch?v=c6tzvZDT5Is>

Interception – The problem

On Android, you have the trusted certificates in the system configuration. Nowadays, divided into system and user certificates.



Apps trust those system certificates and their issued certificates.

Notice: Until Android 7, apps even accepted user defined certificates and its childs. A huge security issue that got fixed. Nevertheless, as reengineers we need exactly that. **How 😊 ?**

Interception – the default way for developers/reengineers

So assuming we want to allow that an app accepts user defined certificates the typical (reengineering) way would be to add the required configuration.

6. Now inside your project create xml file:

res/xml/network_security_config.xml

Add this code:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <network-security-config>
3   <debug-overrides>
4     <trust-anchors>
5       <!-- Trust user added CAs while debuggable only -->
6       <certificates src="user" />
7       <certificates src="system" />
8     </trust-anchors>
9   </debug-overrides>
10  <base-config cleartextTrafficPermitted="true">
11    <trust-anchors>
12      <certificates src="system" />
13    </trust-anchors>
14  </base-config>
15  <domain-config>
16    <!-- Make sure your URL Server here -->
17    <domain includeSubdomains="true">your_production_domain</domain>
18    <trust-anchors>
19      <certificates src="user"/>
20      <certificates src="system"/>
21    </trust-anchors>
22  </domain-config>
23 </network-security-config>
```

network_security_config.xml hosted with ❤ by GitHub

[view raw](#)

in AndroidManifest.xml:

```
<application
...
  android:networkSecurityConfig="@xml/network_security_config"
...>
```

A colleague of mine (Dani) illustrated to do the interception using postman here:

<https://dds861.medium.com/capture-android-request-response-calls-using-postman-efbda09b2317>



Interception – a different solution

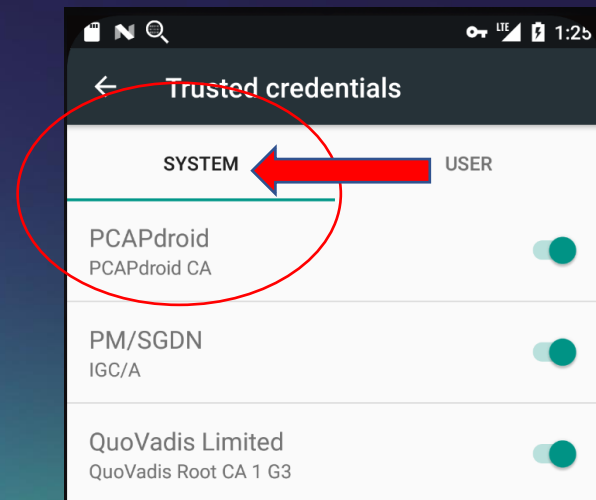
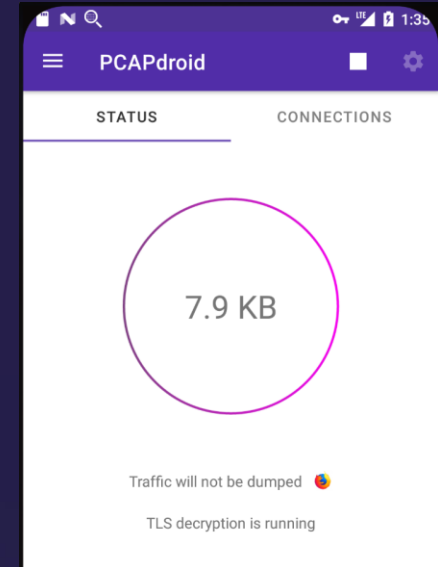
Magic word is: ROOT

F-Droid (third party market) and the tool called PCAPdroid.

It's very user-friendly and allows interception right away by redirecting all traffic through an internal VPN.

Nevertheless, its generated certificate gets installed in the user section by default. Modern apps don't like that anymore.

➔ Solution: root access!



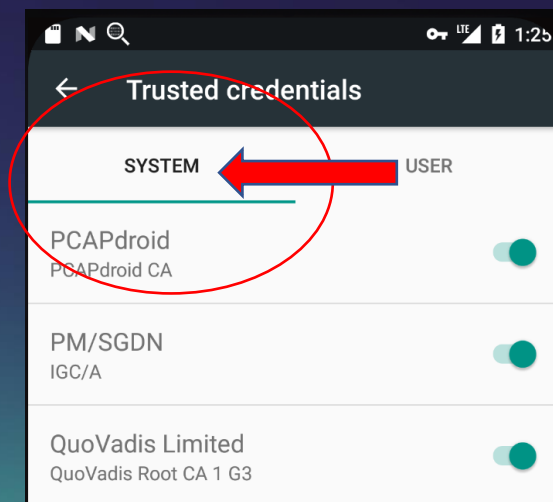
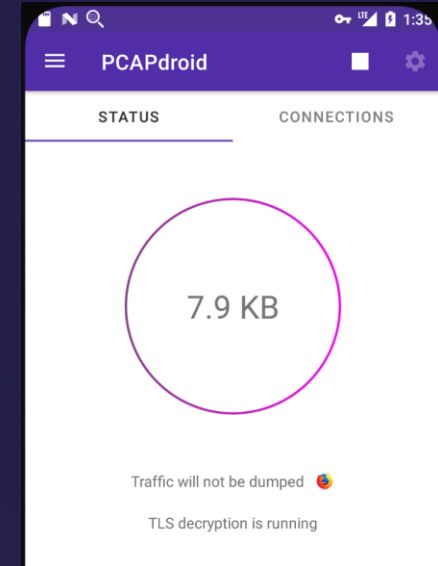
Interception – a different solution

A handy tool for demonstration purposes is the Android emulator with modified configuration.

Notice:

Further solutions include modifying apps to allow them to accept user defined certificates again (reengineering of apps).

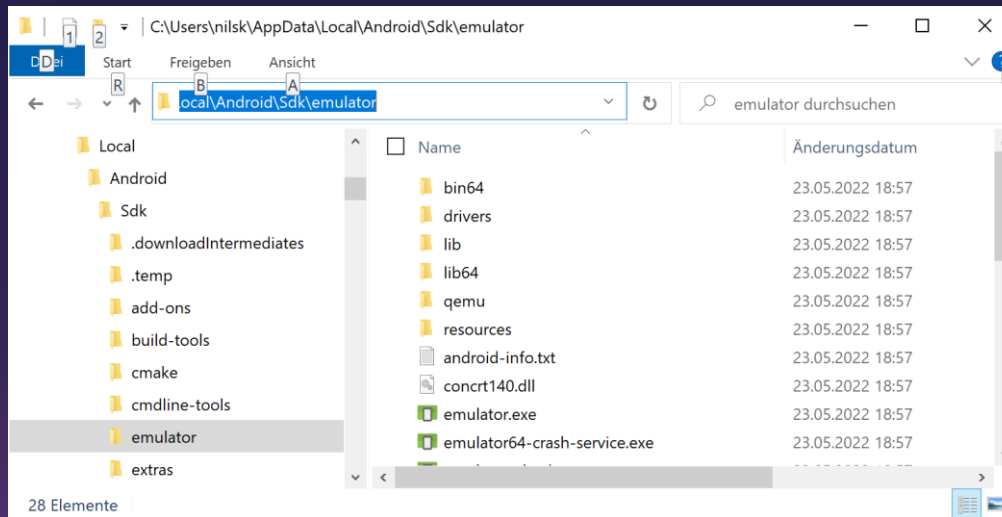
Also, running an Android emulator inside Android like „Virtual Android“ may be an option on real devices. Currently, VA does not allow system image modifications. Unfortunately, I received no reply by the authors on adding that functionality.



Interception – a step by step guide

Step by Step (after installing PCAPdroid and activating the deencryption module)

- 1) Create an AVD in the usual way (I choose Android 7 / API24 to avoid issues right away)
- 2) Discover your SDK and emulator directory (here Windows)



- 3) Get the AVD name:
`emulator.exe -list-avds`
- 4) Start the emulator with writeable system partition:
`emulator.exe -avd LIVE_HACKING_WORKSHOP -writable-system`
- 5) Make sure you have remounted writeable
`adb root` and `adb remount`

Interception

6) Get a shell

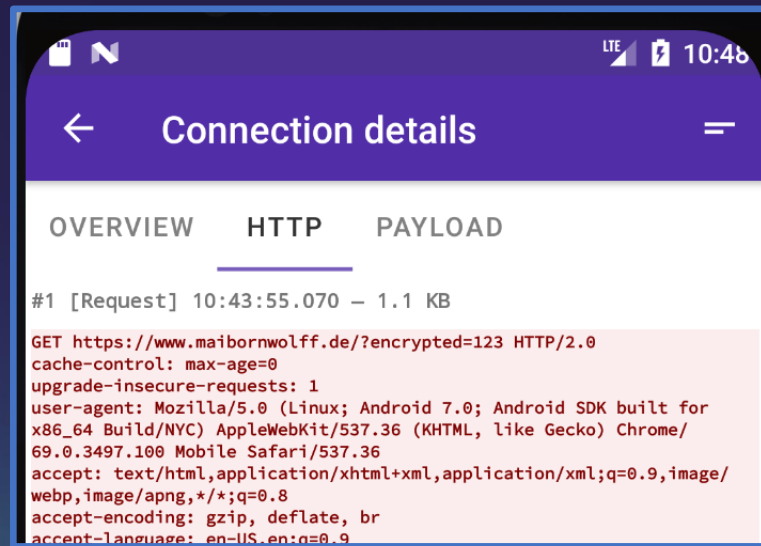
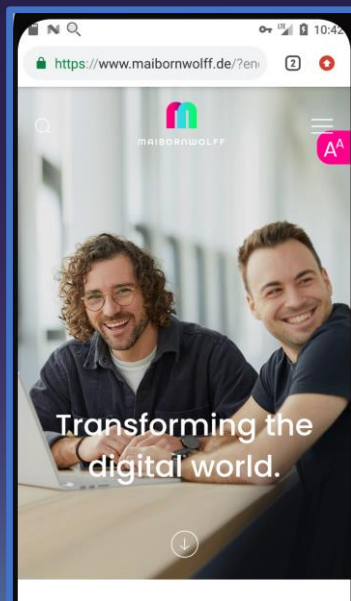
adb shell

7) Move the previous installed user certificate by PCAPdroid to the other system certificates

ls /data/misc/user/0/cacerts-added

```
mv /data/misc/user/0/cacerts-added/certname.0  
/system/etc/security/cacerts/
```

That's it ...



P.S. Did you know that governments can do that on the fly? Any CA in your browser...
HTTPS != SECURE

Deutschland

Spanien

Tunesien



Thanks for your attention.

Last but not least, I'd like to express that we are always looking for new talents.

Please feel free to take a look at open positions at our various branches across Germany, Europe and Africa.

<https://www.maibornwolff.de/jobs/>

We are also interested into university cooperations and invest a lot in R&D.

nils.kannengiesser@maibornwolff.de