Bypass Certificate Pinning in modern Android application via custom Root CA

Author: Nghia Van Le - Sun* Cyber Security Research

TESTING PLATFORM

Host OS: Kali Linux 2019.4

Android Emulator: Using genymotion - Android 6.0 - API Level 23

Tested Device: Rooted Redmi Note 6 Pro – Android 8.1.0 – API Level 27

TOOLS and APPLICATION

Host OS Tools: adb, Burpsuite, MobSF, genymotion

Mobile Tools: Root Certificate Manager (ROOT)

Mobile Application: Airtable

Contents

I. Introduction

This document is intended to provide detailed instructions for bypass certificate pinning via custom Root CA. It covers all the required topics for understanding this method. The proof of concept will help visualize and perform bypass certificate pinning, **specially in modern applications now and in the future**.

II. About Certificate Pinning

By default, an application trusts all the CAs shipped with Operating System (pre-installed CAs), it is around 140 trusted root CA included now [1].

Even though there is only a very small possibility for this to happen: if any of these CAs issue a fraudulent certificate [2],the application is at risk of being hacked by Man-in-the-Middle attack [3]. In addition, the users could be compromised with a rogue certificate installed on their device through social engineering.

To prevent that, the developers has 2 options:

- Limit the set of certificates they accept by either limiting the set of CAs they trust.

- Implement certificate pinning.

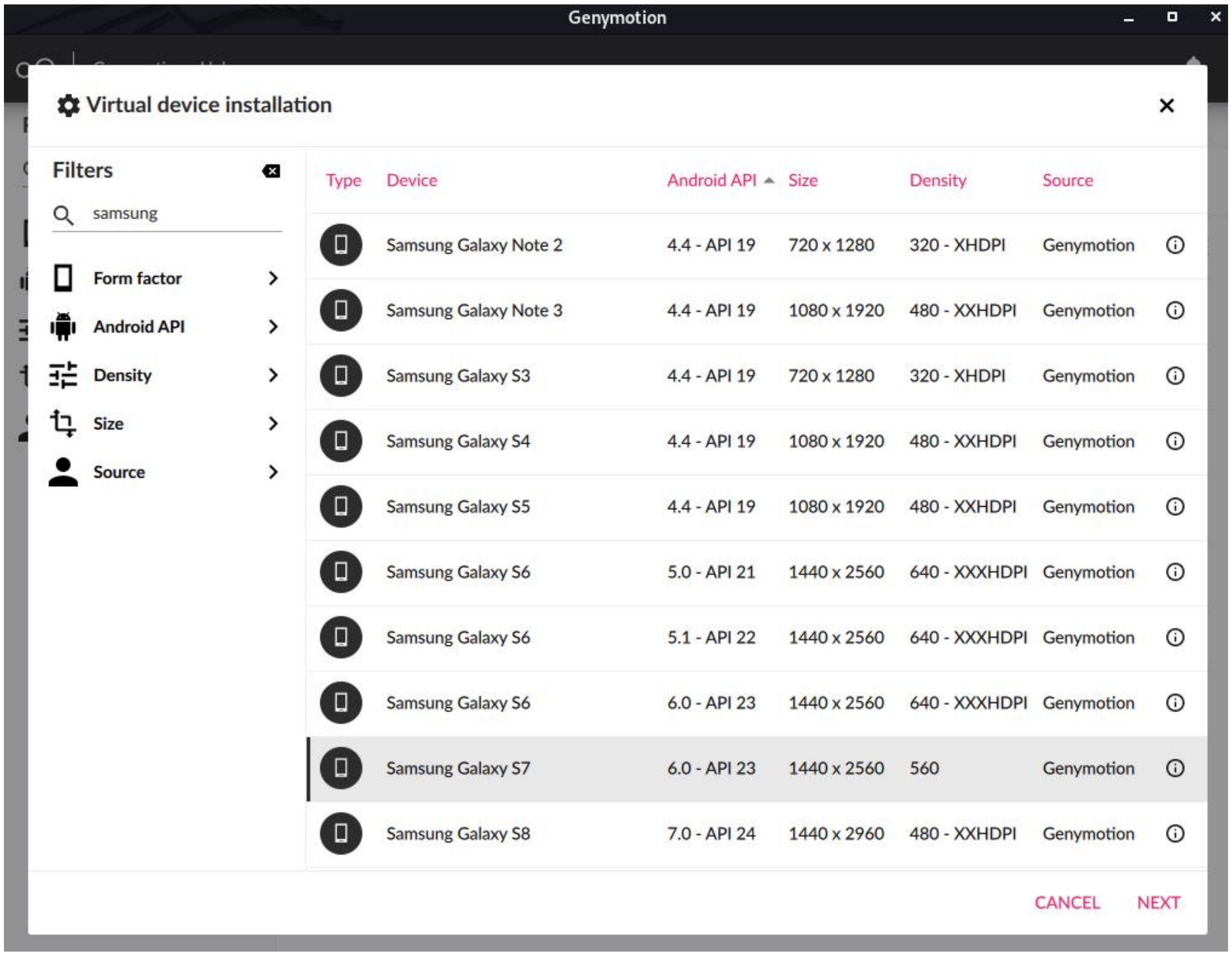
By enhanced security, the cost is negligible and easy to deal with, most developers choose certificate pinning for their applications. The developers embed (or pinning) a list of trustful certificates to their application during development, then use them to compare against the server certificates during runtime. In case of mismatch, the TCP connection will be disrupted, and no further user data will be sent to that server.

III. How does it affect us (Pentester)?

In the phase of static analysis, it has no effect on this process at all. But in dynamic analysis, it can be a huge problem. You cannot observe or intercept the request/response between the application and the server when they communicate with each other, even worse is that the application will not work. If you can't solve this problem, your pentesting process stops here.

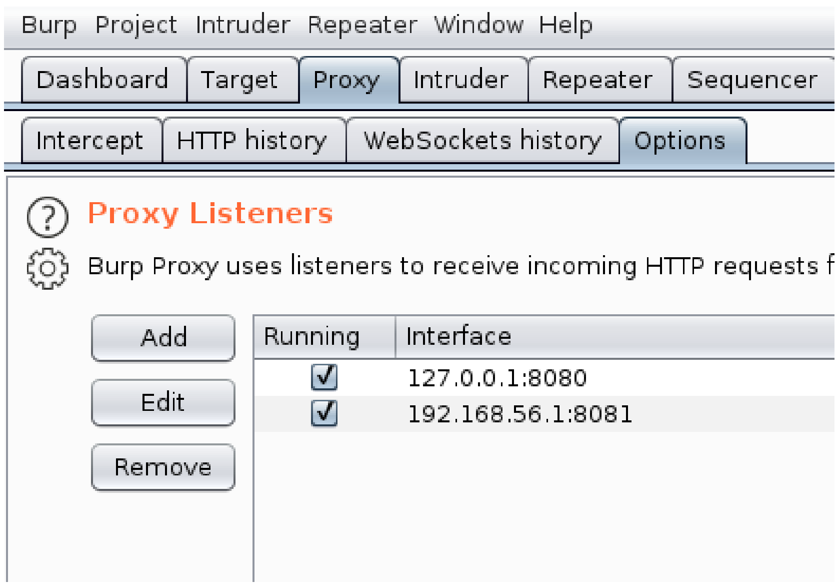
But in fact, we can easily bypass it at this very moment, following this instruction:

1. Using a device or an emulator with Android version 6.0 - API level 23 or below. I don't have any physical device with this Android version so I'm using an emulator, let's create it:



Type	Device	Android API	Size	Density	Source
	Samsung Galaxy Note 2	4.4 - API 19	720 x 1280	320 - XHDPI	Genymotion
	Samsung Galaxy Note 3	4.4 - API 19	1080 x 1920	480 - XXHDPI	Genymotion
	Samsung Galaxy S3	4.4 - API 19	720 x 1280	320 - XHDPI	Genymotion
	Samsung Galaxy S4	4.4 - API 19	1080 x 1920	480 - XXHDPI	Genymotion
	Samsung Galaxy S5	4.4 - API 19	1080 x 1920	480 - XXHDPI	Genymotion
	Samsung Galaxy S6	5.0 - API 21	1440 x 2560	640 - XXXHDPI	Genymotion
	Samsung Galaxy S6	5.1 - API 22	1440 x 2560	640 - XXXHDPI	Genymotion
	Samsung Galaxy S6	6.0 - API 23	1440 x 2560	640 - XXXHDPI	Genymotion
	Samsung Galaxy S7	6.0 - API 23	1440 x 2560	560	Genymotion
	Samsung Galaxy S8	7.0 - API 24	1440 x 2960	480 - XXHDPI	Genymotion

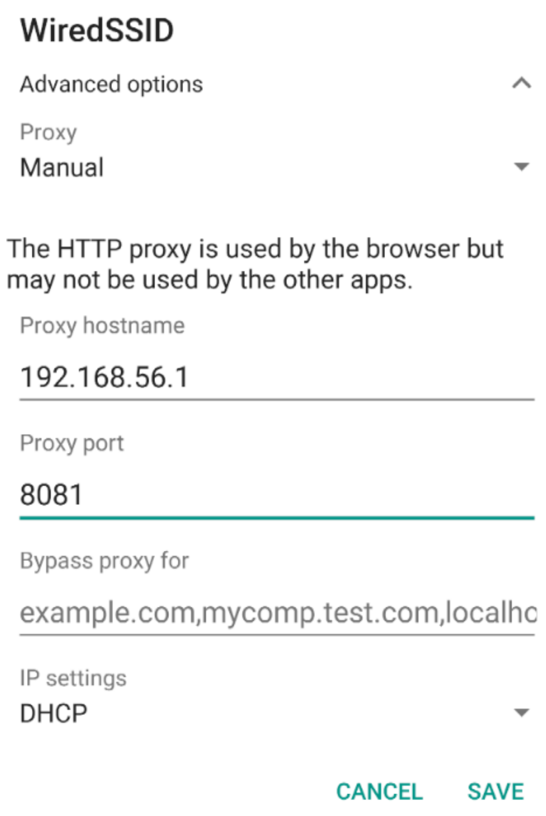
2. Configure emulator to work with Burpsuite's proxy server:


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer
Intercept HTTP history WebSockets history Options
Add
Edit
Remove

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests f

Running	Interface
✓	127.0.0.1:8080
✓	192.168.56.1:8081



WiredSSID

Advanced options

Proxy: Manual

The HTTP proxy is used by the browser but may not be used by the other apps.

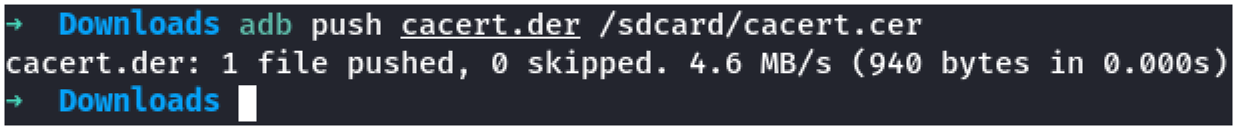
Proxy hostname: 192.168.56.1

Proxy port: 8081

Bypass proxy for: example.com,mycomp.test.com,localhc

IP settings: DHCP

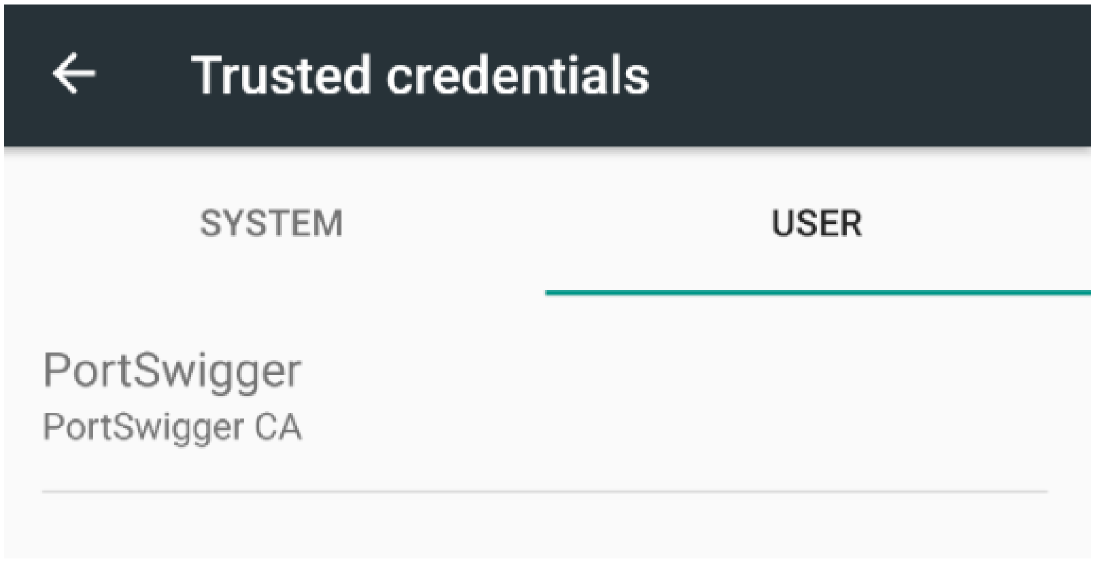
3. Push the Burp’s certificate and install on emulator:



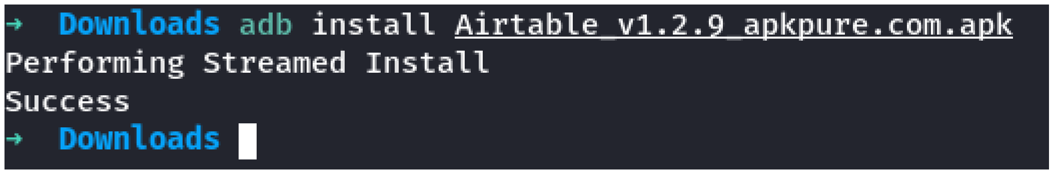
```
→  Downloads adb push cacert.der /sdcard/cacert.cer
cacert.der: 1 file pushed, 0 skipped. 4.6 MB/s (940 bytes in 0.000s)
→  Downloads 
```

Settings > Install certificates > Install certificates > Choose the Burp’s certificate > Create a PIN > Install the certificate and all done.

Now in trusted credentials, in the USER tab, you will see the PortSwigger CA beside SYSTEM root CA:

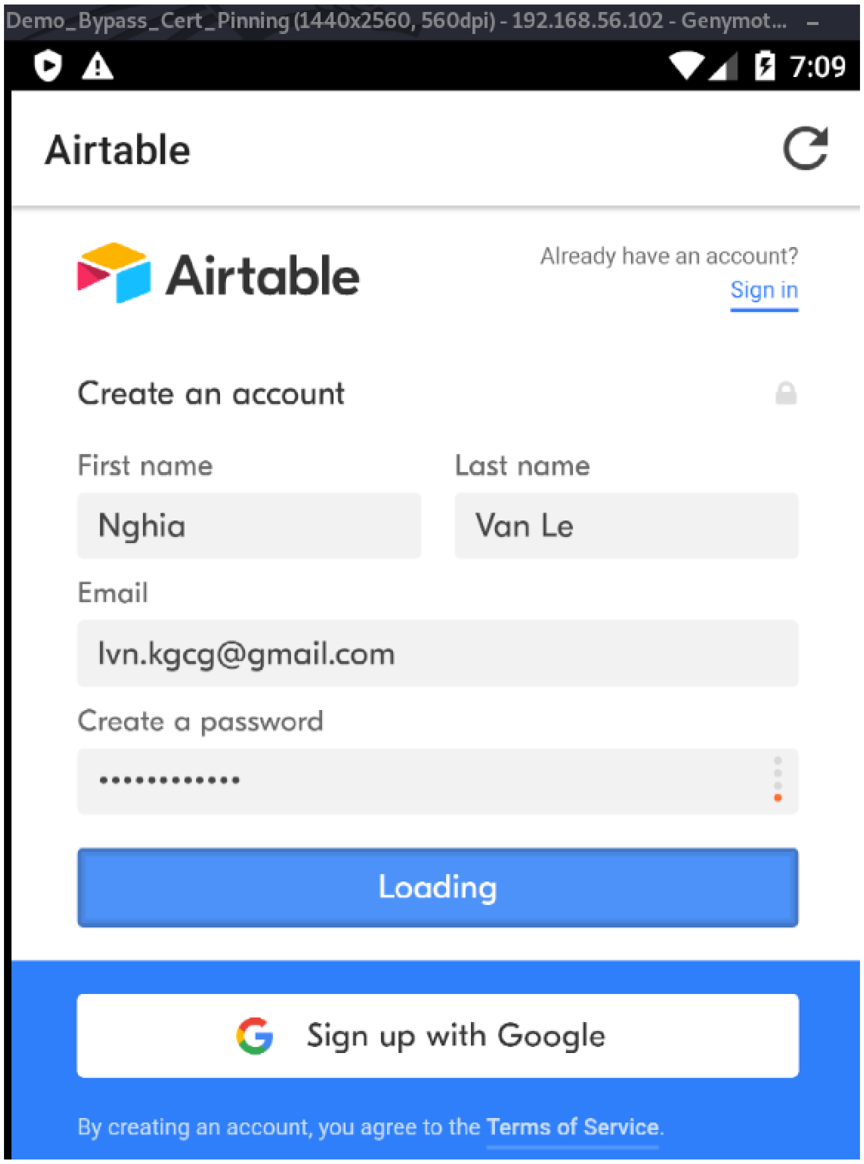


4. Install your application you like to pentesting on this emulator, in this case I’m using the Airtable application, they have a bug bounty program on Hackerone [4], so this is legal:



```
→  Downloads adb install Airtable_v1.2.9_apkpure.com.apk
Performing Streamed Install
Success
→  Downloads 
```

5. Now the app is running and you can fully intercept all the requests and responses between the Airtable and its server.


Demo_Bypass_Cert_Pinning (1440x2560, 560dpi) - 192.168.56.102 - Genymot...

7:09

Airtable

Airtable

Already have an account?

Sign in

Create an account

First name: Nghia

Last name: Van Le

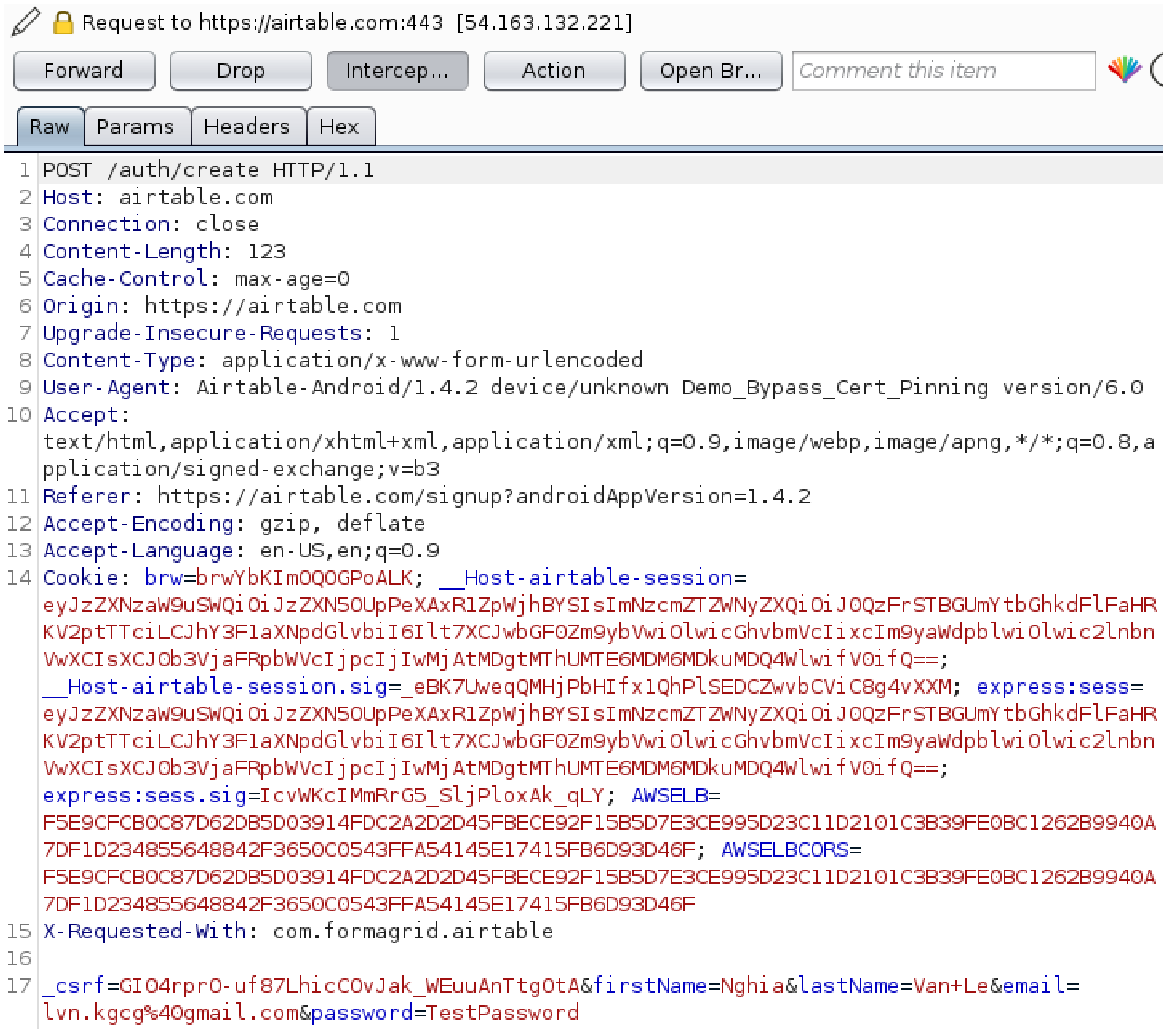
Email: lvn.kgcg@gmail.com

Create a password: ••••••••••••

Loading

Sign up with Google

By creating an account, you agree to the Terms of Service.



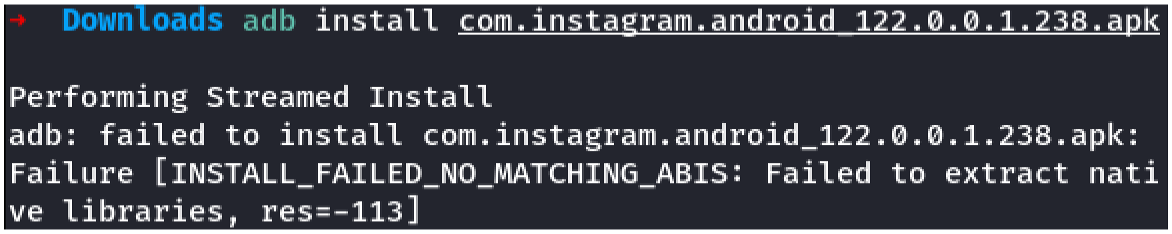
```
POST /auth/create HTTP/1.1
Host: airtable.com
Connection: close
Content-Length: 123
Cache-Control: max-age=0
Origin: https://airtable.com
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Airtable-Android/1.4.2 device/unknown Demo_Bypass_Cert_Pinning version/6.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,a
pplication/signed-exchange;v=b3
Referer: https://airtable.com/signup?androidAppVersion=1.4.2
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: brw=brwYbKImOQOGPoALK; __Host-airtable-session=
eyJzZXNzaW9uSWQiOiJzZXN5OUpPeXAxR1ZpWjhBYSIsImNzcmZTZWNyZXQiOiJ0QzFrSTBGUmYtbGhkdFlFaHR
KV2ptTTciLCJhY3FlaXNpdGlvbiI6Ilt7XCJwbGF0Zm9ybVwiOlwicGhvbmVcIixcIm9yaWdpblwiOlwic2lnbn
VwXCIsXCJ0b3VjaFRpbWVcIjpcIjIwMjAtMDgtMThUMTE6MDM6MDkuMDQ4WlwifV0ifQ==;
__Host-airtable-session.sig=_eBK7UweqQMHjPbHIfx1QhPlSEDCZwvbCViC8g4vXXM; express:sess=
eyJzZXNzaW9uSWQiOiJzZXN5OUpPeXAxR1ZpWjhBYSIsImNzcmZTZWNyZXQiOiJ0QzFrSTBGUmYtbGhkdFlFaHR
KV2ptTTciLCJhY3FlaXNpdGlvbiI6Ilt7XCJwbGF0Zm9ybVwiOlwicGhvbmVcIixcIm9yaWdpblwiOlwic2lnbn
VwXCIsXCJ0b3VjaFRpbWVcIjpcIjIwMjAtMDgtMThUMTE6MDM6MDkuMDQ4WlwifV0ifQ==;
express:sess.sig=IcvWKcIMmRrG5_SljPloxAk_qLY; AWSELB=
F5E9CFCB0C87D62DB5D03914FDC2A2D2D45FBECE92F15B5D7E3CE995D23C11D2101C3B39FE0BC1262B9940A
7DF1D234855648842F3650C0543FFA54145E17415FB6D93D46F; AWSELBCORS=
F5E9CFCB0C87D62DB5D03914FDC2A2D2D45FBECE92F15B5D7E3CE995D23C11D2101C3B39FE0BC1262B9940A
7DF1D234855648842F3650C0543FFA54145E17415FB6D93D46F
X-Requested-With: com.formagrid.airtable

_csrf=GIO4rprO-uf87LhicCOvJak_WEuuAnTtgOtA&firstName=Nghia&lastName=Van+Le&email=
lvn.kgcg%40gmail.com&password=TestPassword
```

This is the easiest way, but personally I think this method won't work in the near future.

Root cause:

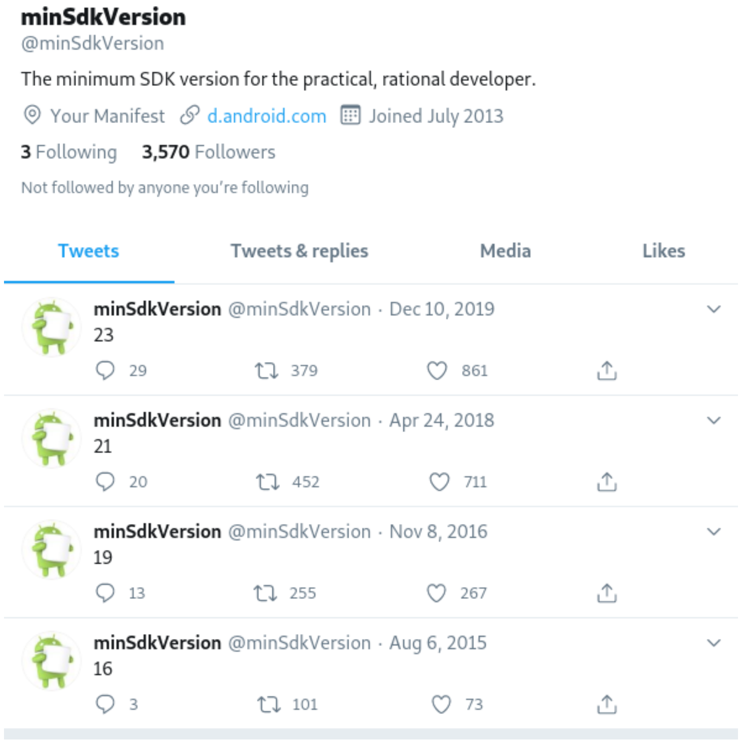
The first reason is this method only works on devices/emulators with Android version 6.0 - API level 23 or below. This is because of “Apps that target API Level 24 and above no longer trust user or admin-added CAs for secure connections, by default” [5].

I know that not too many physical devices are still running android 6. In the case of insufficient facilities, you can use an emulator. But lots of applications do not allow installation on virtualized devices, you can bypass it by some method, but it makes things more complicated.



```
➜  Downloads adb install com.instagram.android_122.0.0.1.238.apk

Performing Streamed Install
adb: failed to install com.instagram.android_122.0.0.1.238.apk:
Failure [INSTALL_FAILED_NO_MATCHING_ABIS: Failed to extract nati
ve libraries, res=-113]
```

Second reason and most importantly: in Android software development, the minSdkVersion is increasing every year. Android 6 - API level 23 was released in October 2015, it's been over 4 years.



minSdkVersion
@minSdkVersion
The minimum SDK version for the practical, rational developer.
Your Manifest d.android.com Joined July 2013
3 Following 3,570 Followers
Not followed by anyone you're following

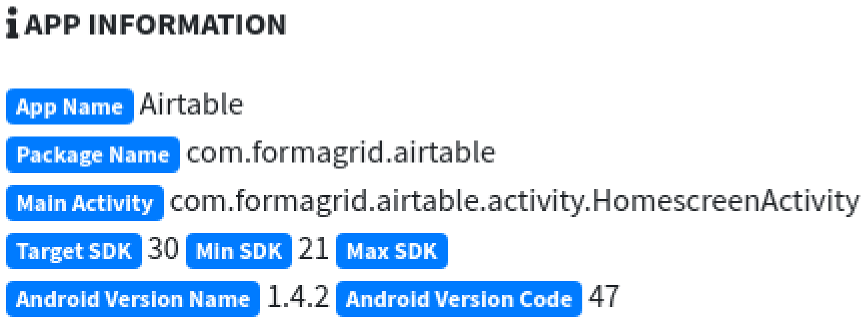
minSdkVersion @minSdkVersion · Dec 10, 2019
23
29 379 861

minSdkVersion @minSdkVersion · Apr 24, 2018
21
20 452 711

minSdkVersion @minSdkVersion · Nov 8, 2016
19
13 255 267

minSdkVersion @minSdkVersion · Aug 6, 2015
16
3 101 73

In the application we installed above (Airtable), it has the minSDK version 21. So the method above is still working. I tested some other popular apps, most of them have the minSdk version 21 as well. But just in the next few years, it will change.



APP INFORMATION

App Name Airtable
Package Name com.formagrid.airtable
Main Activity com.formagrid.airtable.activity.HomescreenActivity
Target SDK 30 Min SDK 21 Max SDK
Android Version Name 1.4.2 Android Version Code 47

IV. Bypass Certificate Pinning via custom Root CA

Let's say you don't have a device with Android 6 or lower, or that the app doesn't allow installation on devices with API level 23 or below. How can you dynamic penetration testing this app?

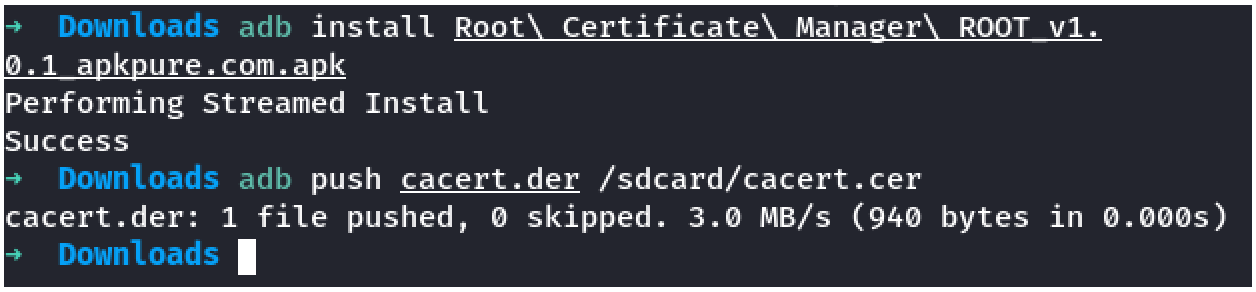
This method will remove every obstacle in the way, or I can say in the future way.

By pentesting on a rooted device, you can manually install a Root CA on your phone. Then you can intercept all the requests and responses easily, like the way I just mentioned.

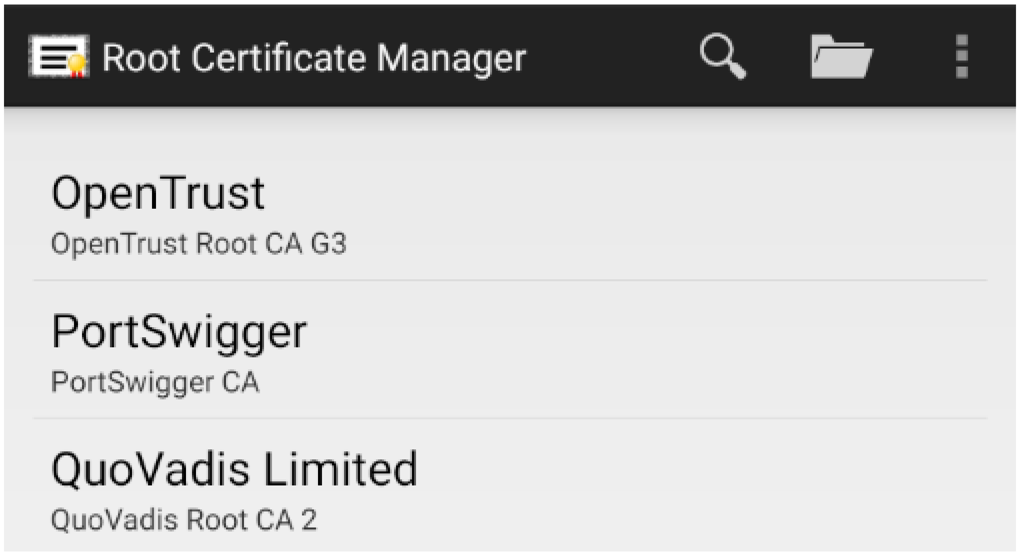
But you cannot use the existing Burp's CA certificate, let me show you:

1. First, I'll install the **Root Certificate Manager ROOT** application and push the Burp's CA certificate to my device: Redmi Note 6 Pro - running Android 8.1 – API level 27.

2. Burp's CA certificate installation successfully via Root Certificate Manager :



```
→  Downloads adb install Root\ Certificate\ Manager\ ROOT_v1.
0.1_apkpure.com.apk
Performing Streamed Install
Success
→  Downloads adb push cacert.der /sdcard/cacert.cer
cacert.der: 1 file pushed, 0 skipped. 3.0 MB/s (940 bytes in 0.000s)
→  Downloads
```



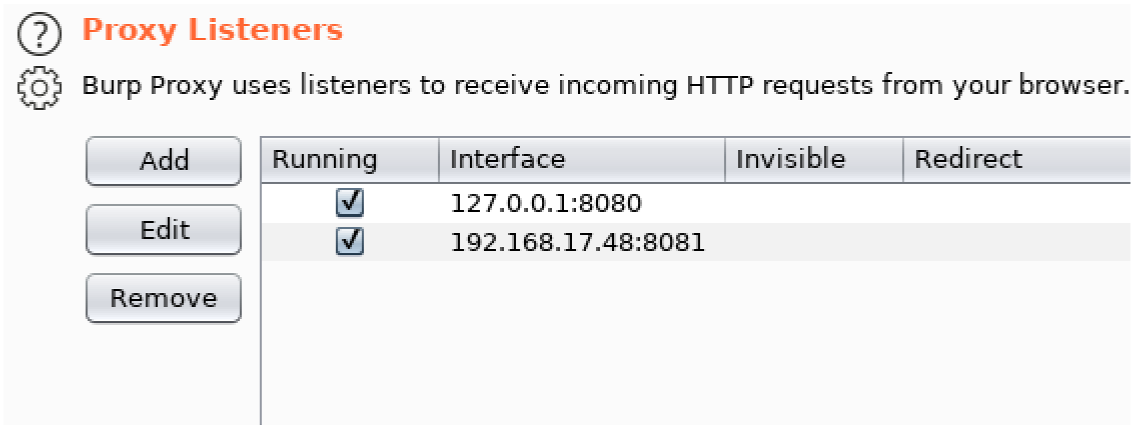
Root Certificate Manager

OpenTrust
OpenTrust Root CA G3

PortSwigger
PortSwigger CA

QuoVadis Limited
QuoVadis Root CA 2

3. Configure device to work with Burpsuite's proxy server:

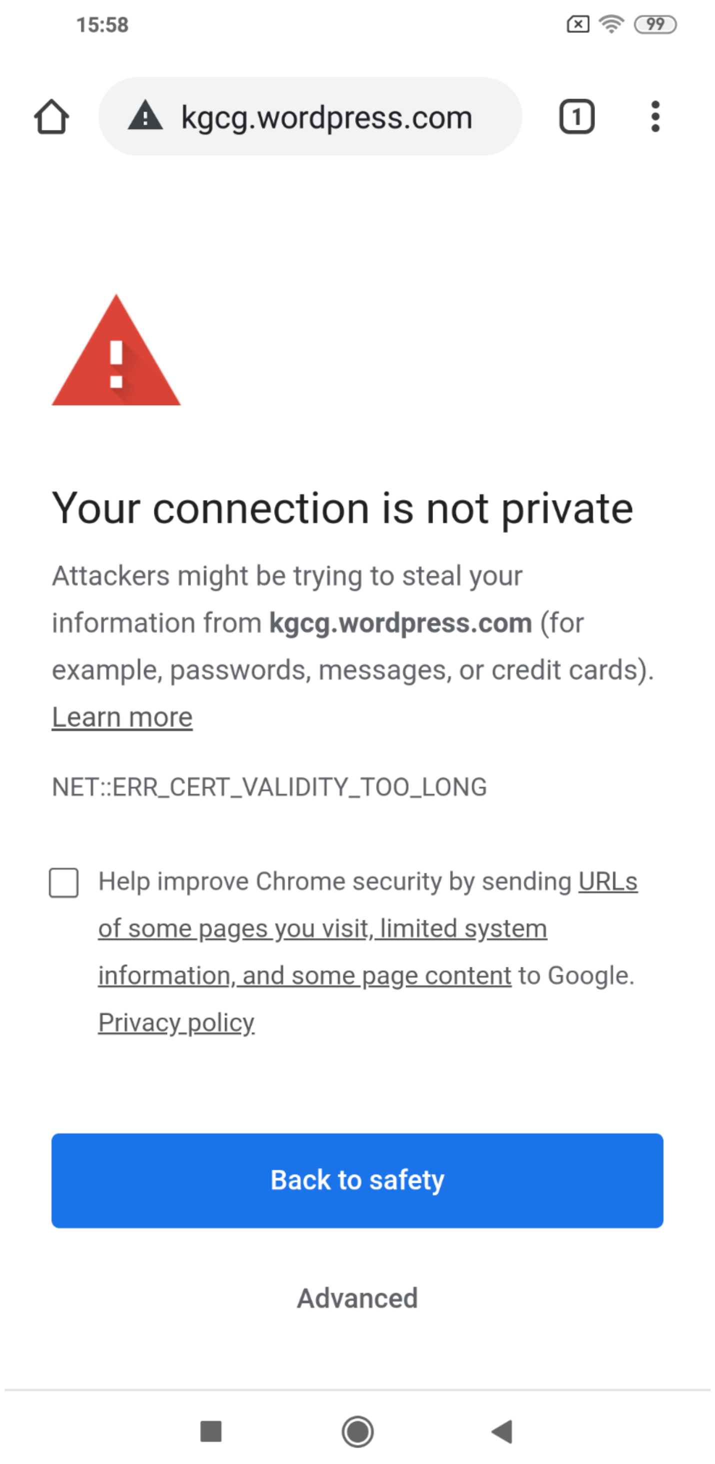


Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser.

Running	Interface	Invisible	Redirect
✓	127.0.0.1:8080		
✓	192.168.17.48:8081		

But when I try to access any HTTPs website in browser, I get the following error:
NET::ERR_CERT_VALIDITY_TOO_LONG



15:58

kgcg.wordpress.com

Your connection is not private

Attackers might be trying to steal your information from **kgcg.wordpress.com** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_VALIDITY_TOO_LONG

☐ Help improve Chrome security by sending URLs of some pages you visit, limited system information, and some page content to Google. Privacy policy

Open the app and it’s totally blank:

16:02
Airtable

The root cause is Burpsuite’s CA certificate validity too long and regenerating the certificate could not solve the problem.

Solution:

According to the Postwigger pages [6] I can import my custom CA certificate and they also have a brief guide. But here is the full tutorial to help you during dynamic pentesting the apps:

1. Create a folder

```
mkdir cert && cd cert
```

2. Install openssl

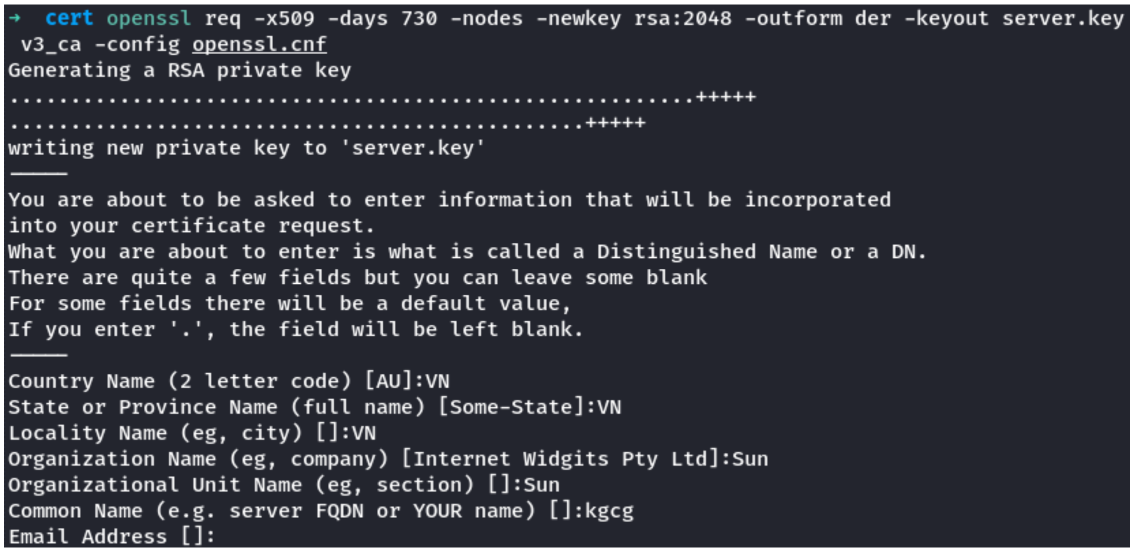
```
sudo apt-get install openssl
```

3. Find the default openssl config file and copy the default openssl.cnf

```
cp /etc/ssl/openssl.cnf ./
```

4. Create a private key, the "days" value is 730 means it < 2 year of validity. Then fill out some fields:

```
openssl req -x509 -days 730 -nodes -newkey rsa:2048 -outform der -keyout server.key -out ca.der -extensions v3_ca -config openssl.cnf
```



```
→  cert openssl req -x509 -days 730 -nodes -newkey rsa:2048 -outform der -keyout server.key
 v3_ca -config openssl.cnf
Generating a RSA private key
.......................................................+++++
...............................................+++++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:VN
State or Province Name (full name) [Some-State]:VN
Locality Name (eg, city) []:VN
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Sun
Organizational Unit Name (eg, section) []:Sun
Common Name (e.g. server FQDN or YOUR name) []:kgcg
Email Address []:
```

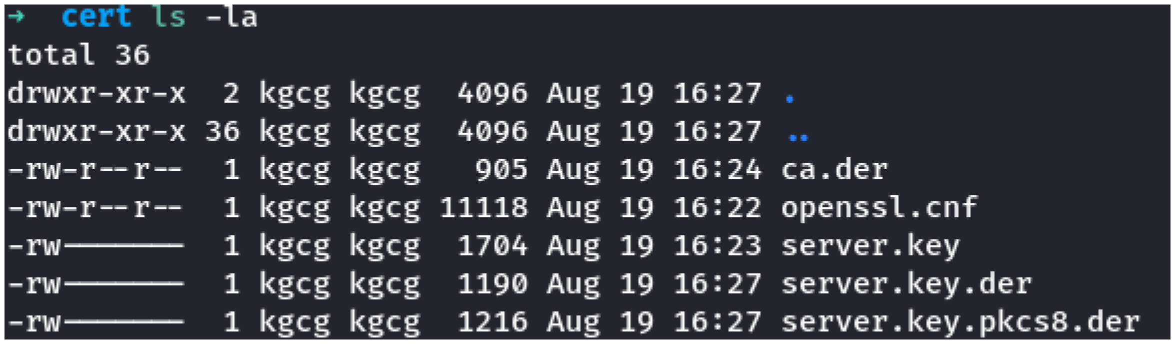
5. Convert to der format:

```
openssl rsa -in server.key -inform pem -out server.key.der -outform der
```

6. Convert key to pkcs8 format:

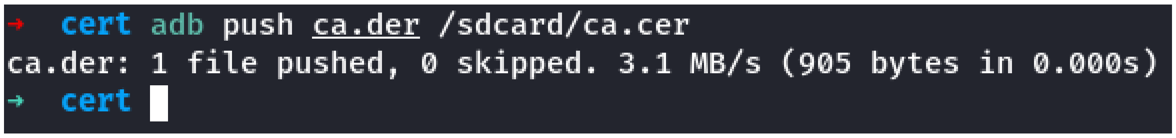
openssl pkcs8 -topk8 -in server.key.der -inform der -out server.key.pkcs8.der -outform der -nocrypt

Now we had 5 files in our **cert** folder:

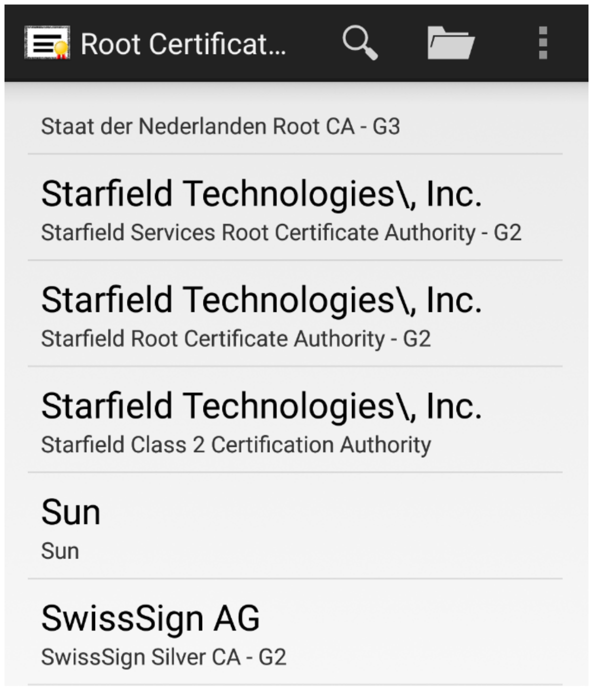


```
➜  cert ls -la
total 36
drwxr-xr-x  2 kgcg kgcg  4096 Aug 19 16:27 .
drwxr-xr-x 36 kgcg kgcg  4096 Aug 19 16:27 ..
-rw-r--r--  1 kgcg kgcg   905 Aug 19 16:24 ca.der
-rw-r--r--  1 kgcg kgcg 11118 Aug 19 16:22 openssl.cnf
-rw-------  1 kgcg kgcg  1704 Aug 19 16:23 server.key
-rw-------  1 kgcg kgcg  1190 Aug 19 16:27 server.key.der
-rw-------  1 kgcg kgcg  1216 Aug 19 16:27 server.key.pkcs8.der
```

7. Push certificate to device and install it, I named it **Sun**:



```
➜  cert adb push ca.der /sdcard/ca.cer
ca.der: 1 file pushed, 0 skipped. 3.1 MB/s (905 bytes in 0.000s)
➜  cert
```



Staat der Nederlanden Root CA - G3

Starfield Technologies\, Inc.
Starfield Services Root Certificate Authority - G2

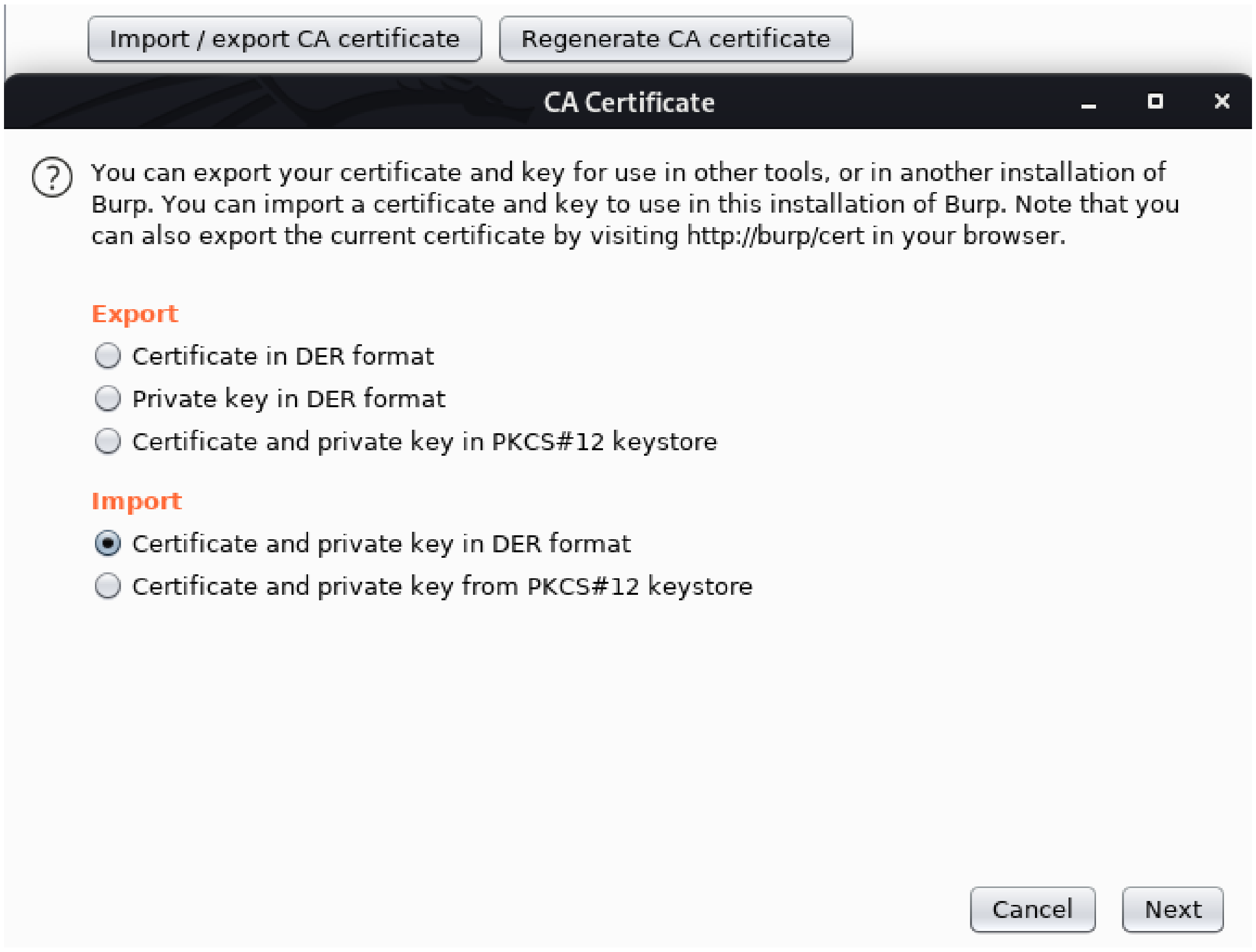
Starfield Technologies\, Inc.
Starfield Root Certificate Authority - G2

Starfield Technologies\, Inc.
Starfield Class 2 Certification Authority

Sun
Sun

SwissSign AG
SwissSign Silver CA - G2

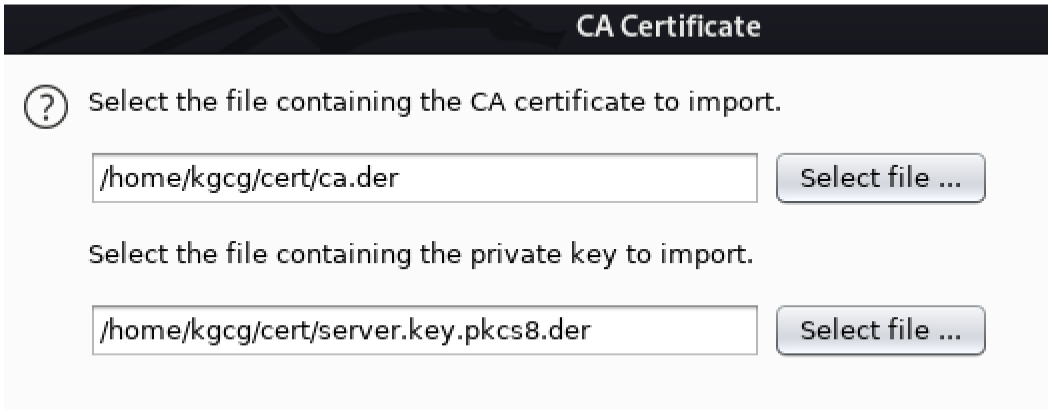
8. Importing these files to Burp's proxy server: "ca.der" and server.key.pkcs8.der:



You can export your certificate and key for use in other tools, or in another installation of Burp. You can import a certificate and key to use in this installation of Burp. Note that you can also export the current certificate by visiting http://burp/cert in your browser.

Export
- ( ) Certificate in DER format
- ( ) Private key in DER format
- ( ) Certificate and private key in PKCS#12 keystore

Import
- (●) Certificate and private key in DER format
- ( ) Certificate and private key from PKCS#12 keystore



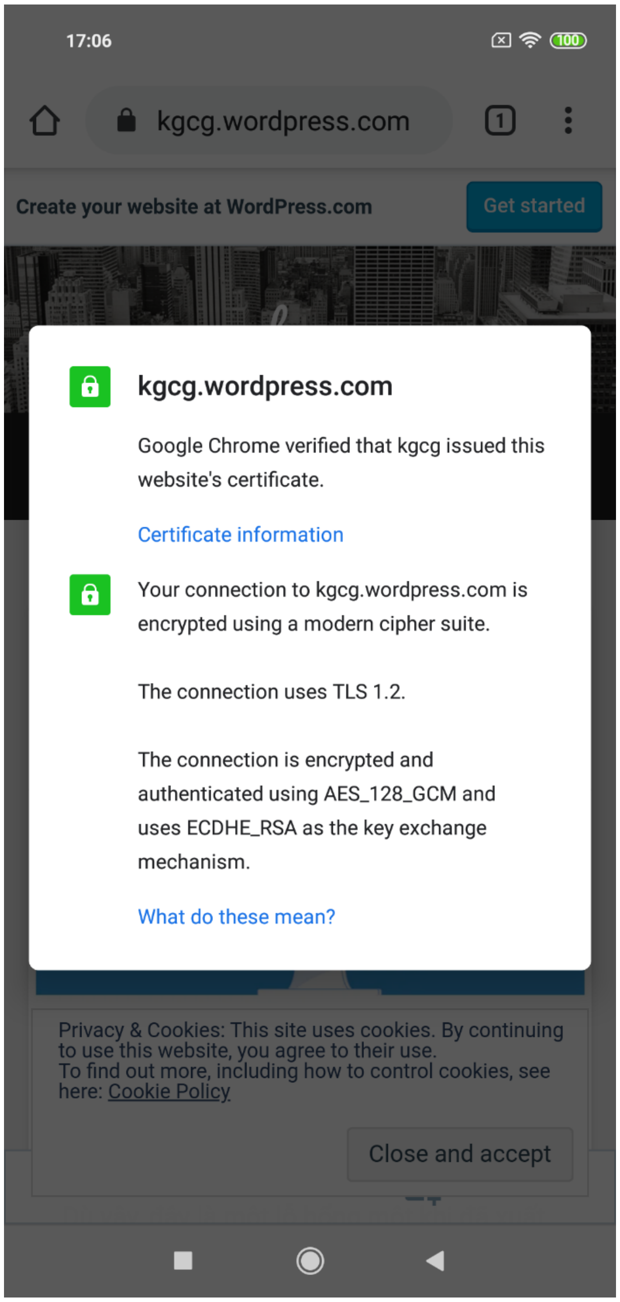
Select the file containing the CA certificate to import.

/home/kgcg/cert/ca.der

Select the file containing the private key to import.

/home/kgcg/cert/server.key.pkcs8.der

Now all done and you can intercept all the traffic:



kgcg.wordpress.com

Google Chrome verified that kgcg issued this website's certificate.

Certificate information

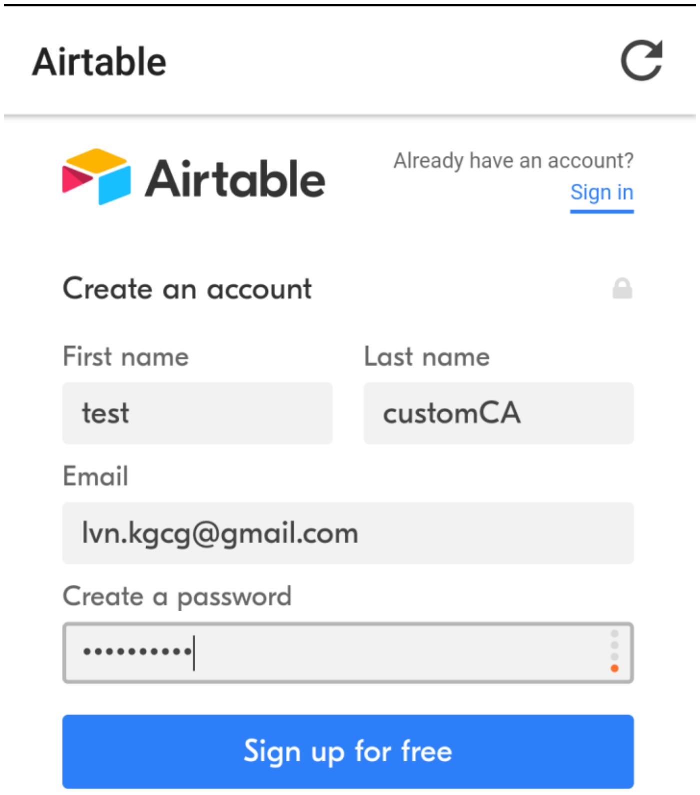
Your connection to kgcg.wordpress.com is encrypted using a modern cipher suite.

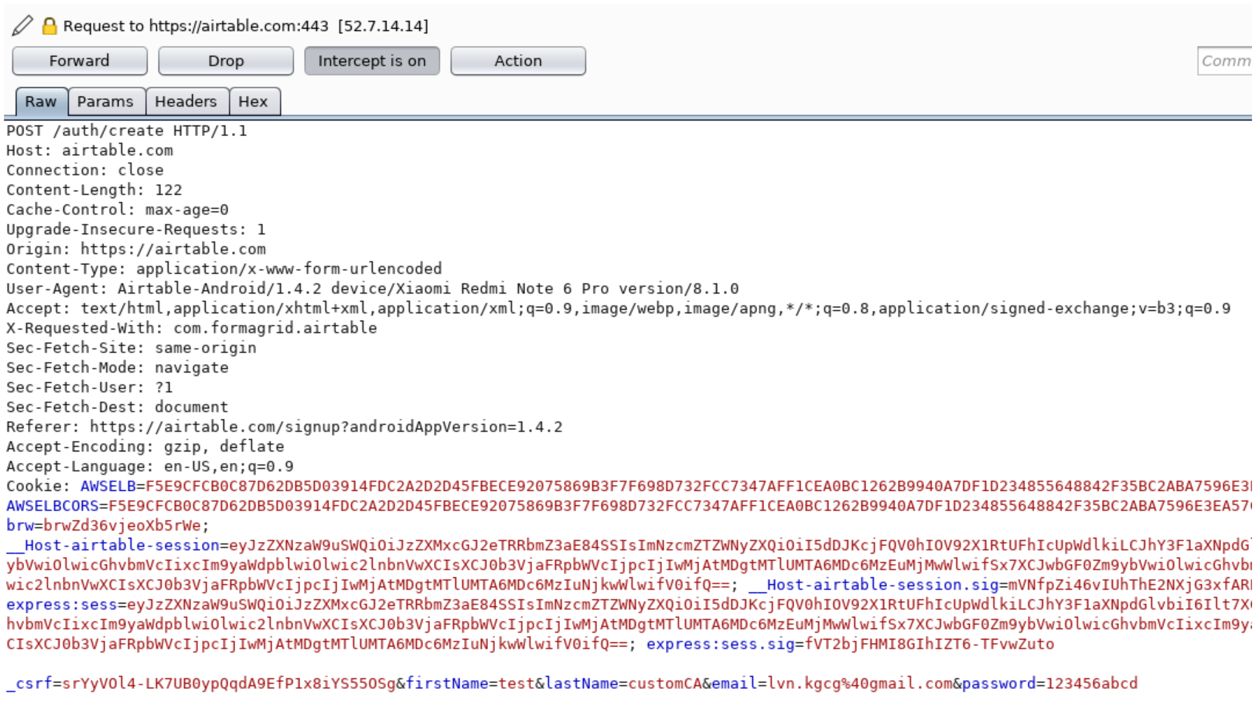
The connection uses TLS 1.2.

The connection is encrypted and authenticated using AES_128_GCM and uses ECDHE_RSA as the key exchange mechanism.

What do these mean?

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.
To find out more, including how to control cookies, see here: Cookie Policy





```
POST /auth/create HTTP/1.1
Host: airtable.com
Connection: close
Content-Length: 122
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: https://airtable.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Airtable-Android/1.4.2 device/Xiaomi Redmi Note 6 Pro version/8.1.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
X-Requested-With: com.formagrid.airtable
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://airtable.com/signup?androidAppVersion=1.4.2
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: AWSELB=F5E9CFCB0C87D62DB5D03914FDC2A2D2D45FBECE92075869B3F7F698D732FCC7347AFF1CEA0BC1262B9940A7DF1D234855648842F35BC2ABA7596E3
AWSELBCORS=F5E9CFCB0C87D62DB5D03914FDC2A2D2D45FBECE92075869B3F7F698D732FCC7347AFF1CEA0BC1262B9940A7DF1D234855648842F35BC2ABA7596E3EA57
brw=brwZd36vjeoXb5rWe;
__Host-airtable-session=eyJzZXNzaW9uSWQiOiJzZXMxcGJ2eTRRbmZ3aE84SSIsImNzcmZTZWNyZXQiOiI5dDJKcjFQV0hIOV92X1RtUFhIcUpWdlkiLCJhY3F1aXNpdG
ybVwiOlwicGhvbmVcIixcIm9yaWdpblwiOlwic2lnbnVwXCIsXCJ0b3VjaFRpbWVcIjpcIjIwMjAtMDgtMTlUMTA6MDc6MzEuMjMwWlwifSx7XCJwbGF0Zm9ybVwiOlwicGhvb
wic2lnbnVwXCIsXCJ0b3VjaFRpbWVcIjpcIjIwMjAtMDgtMTlUMTA6MDc6MzIuNjkwWlwifV0ifQ==; __Host-airtable-session.sig=mVNfpZi46vIUhThE2NXjG3xfAR
express:sess=eyJzZXNzaW9uSWQiOiJzZXMxcGJ2eTRRbmZ3aE84SSIsImNzcmZTZWNyZXQiOiI5dDJKcjFQV0hIOV92X1RtUFhIcUpWdlkiLCJhY3F1aXNpdGlvbiI6Ilt7X
hvbmVcIixcIm9yaWdpblwiOlwic2lnbnVwXCIsXCJ0b3VjaFRpbWVcIjpcIjIwMjAtMDgtMTlUMTA6MDc6MzEuMjMwWlwifSx7XCJwbGF0Zm9ybVwiOlwicGhvbmVcIixcIm9y
CIsXCJ0b3VjaFRpbWVcIjpcIjIwMjAtMDgtMTlUMTA6MDc6MzIuNjkwWlwifV0ifQ==; express:sess.sig=fVT2bjFHMI8GIhIZT6-TFvwZuto

_csrf=srYyVOl4-LK7UB0ypQqdA9EfP1x8iYS55OSg&firstName=test&lastName=customCA&email=lvn.kgcg%40gmail.com&password=123456abcd
```

V. References

[1]: https://android.googlesource.com/platform/system/ca-certificates/+/master/files/

[2]: https://en.wikipedia.org/wiki/DigiNotar#Issuance_of_fraudulent_certificates

[3]: https://en.wikipedia.org/wiki/Man-in-the-middle_attack

[4]: https://hackerone.com/airtable

[5]: https://android-developers.googleblog.com/2016/07/changes-to-trusted-certificate.html

[6]: https://portswigger.net/burp/documentation/desktop/tools/proxy/options