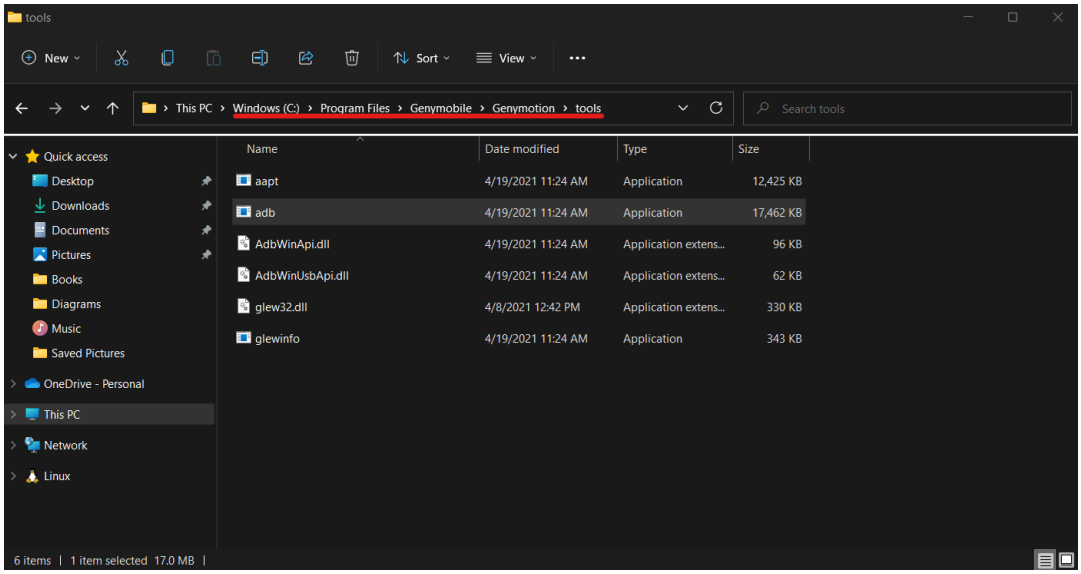
CA Installation for Root on Android Devices

- This issue occurs with the latest versions of Android (API ≥ 24).
- The change was made by Android developers on **07 July 2016**. This can be found at the following blog: **https://android-developers.googleblog.com/2016/07/changes-to-trusted-certificate.html**
- The change was made with Android Nougat, which changes **how applications interact with user-supplied CAs and admin-supplied CAs** (also referred to as **system-supplied CAs**).
- By default, apps that are designed such that they use the **API level 24** will not trust **user-supplied CAs**.

> ℹ️ **Note**: The app developer can **explicitly** specify if the app should trust **user-supplied** certificates. This is discussed in the blog post mentioned above.

Adding a CA as a System-Supplied CA

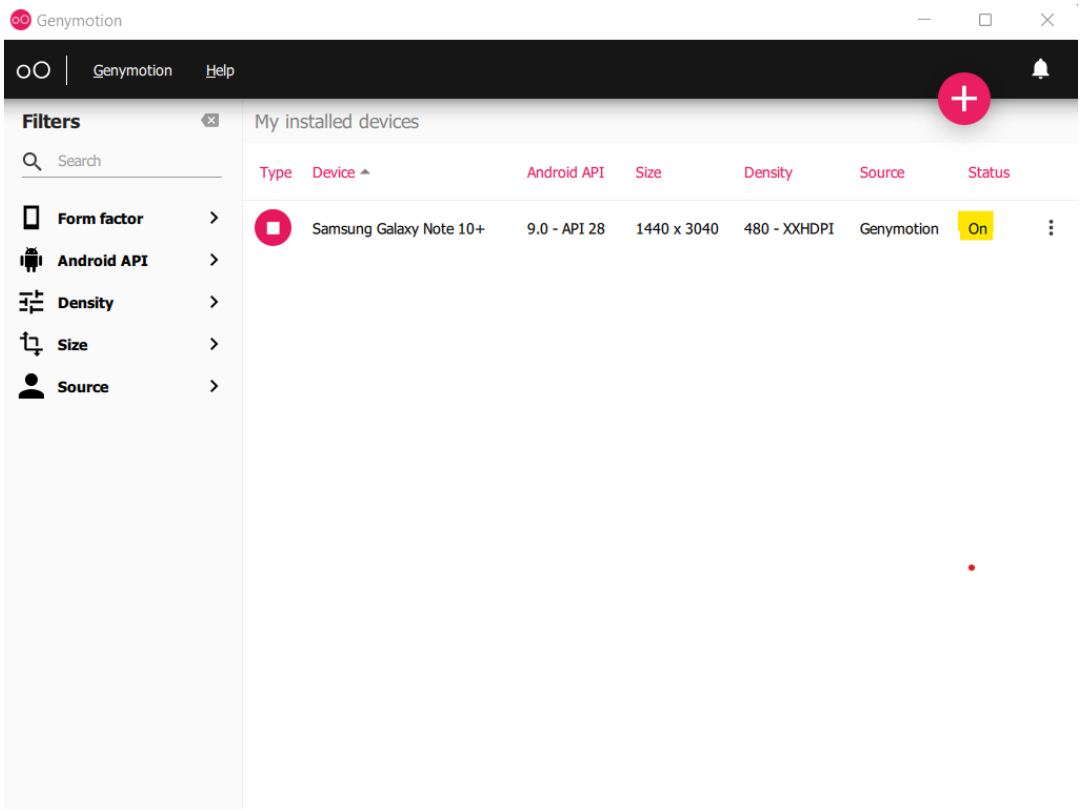
- The **system-added CAs** would be found on the following path: `/system/etc/security/cacerts`.
- In case you are installing the certificate on your own mobile, you must get root access on your device.
- In case you are running a VM on **Genymotion**, then you can get root access as follows:
    1. When you install Genymotion, there are a couple of tools that get installed with it. The tool that provides with a root access to the **running VM** is called `adb.exe`.



- You can run it directly from the path:
  - `"C:\Windows\Program Files\Genymotion\Genymotion\tools`

> **Note**: You can also add the path to environment variables to run `adb.exe` from anywhere.

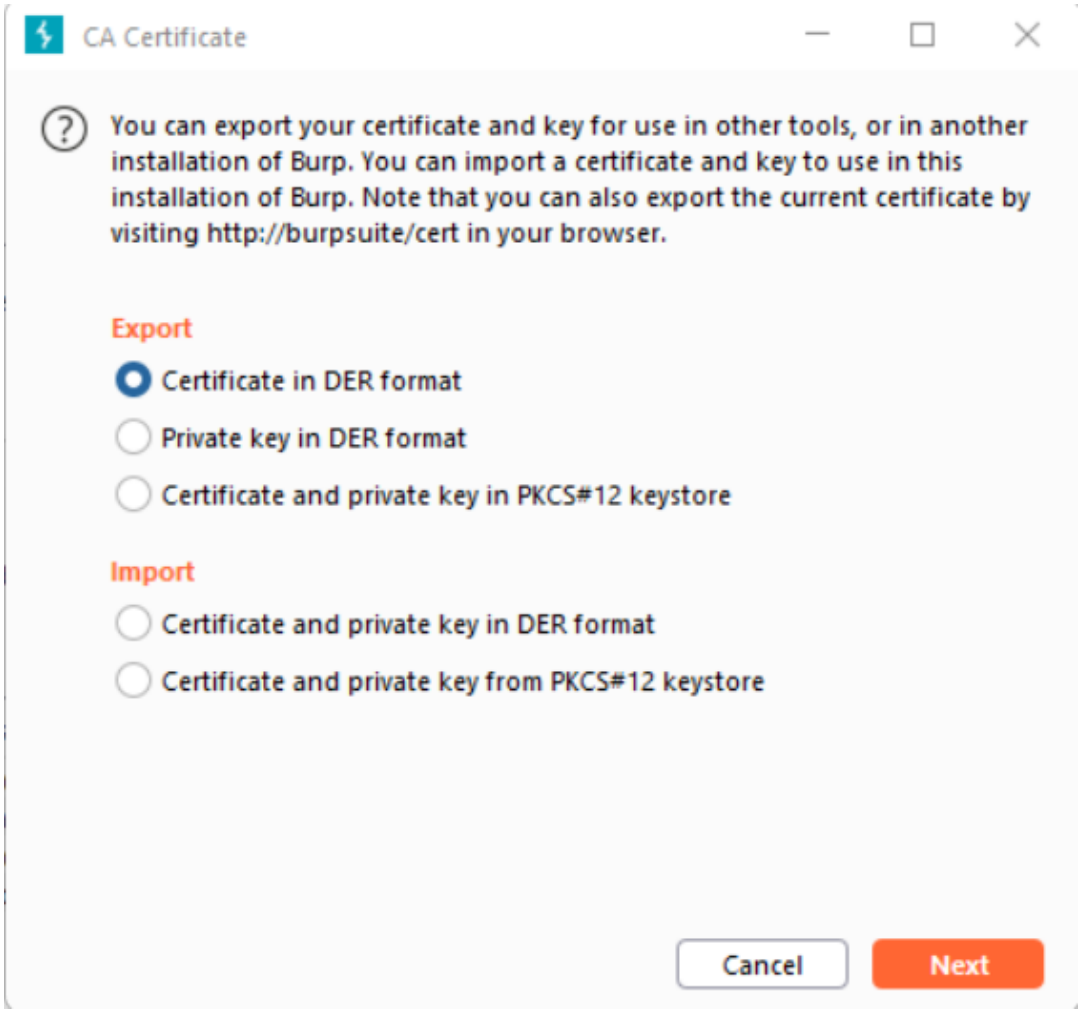
2. You can get root access to the machine while it is running by typing `adb shell`:




```
C:\>cd "C:\Program Files\Genymobile\Genymotion\tools"

C:\Program Files\Genymobile\Genymotion\tools>adb.exe shell
vbox86p:/ # ls
acct                 init.usb.rc             sdcard
bin                  init.vbox86.rc          sepolicy
bugreports           init.zygote32.rc        storage
cache                mnt                     sys
charger              odm                     system
config               oem                     tmp
d                    plat_file_contexts      ueventd.rc
data                 plat_hwservice_contexts ueventd.vbox86.rc
default.prop         plat_property_contexts  var
dev                  plat_seapp_contexts     vendor
etc                  plat_service_contexts   vendor_file_contexts
fstab.vbox86         proc                    vendor_hwservice_contexts
init                 product                 vendor_property_contexts
init.environ.rc      rom.trace               vendor_seapp_contexts
init.rc              root                    vendor_service_contexts
init.usb.configfs.rc sbin                    vndservice_contexts
vbox86p:/ #
```

3. Next, you would have to export Burpsuite's certificate and convert it from `.cer` **(DER encoded)** format to `.pem` format using **Openssl** tool:



4. After you import the certificate, you should run the following commands:
    - `openssl x509 -inform DER -in cacert.cer -out cacert.pem`
    - `openssl x509 -inform PEM -subject_hash_old -in cacert.pem` - you run this command to get the hash of the subject name of the certificate (the reason for this is that Android devices store certificates in `.pem` format with the filename being the hash value appended with `.0`)

```
C:\Program Files\OpenSSL-Win64\bin>openssl x509 -inform DER -in cacert.cer -out cacert.pem

C:\Program Files\OpenSSL-Win64\bin>openssl x509 -inform PEM -subject_hash_old -in cacert.pem
9a5ba575
-----BEGIN CERTIFICATE-----
MIIDqDCCApCgAwIBAgIFAO17iVUwDQYJKoZIhvcNAQELBQAwgYoxFDASBgNVBAYT
C1BvcnRTd2lnZ2VyMRQwEgYDVQQIEwtQb3J0U3dpZ2dlcjEUMBIGA1UEBxMLUG9y
dFN3aWdnZXIxFDASBgNVBAoTC1BvcnRTd2lnZ2VyMRcwFQYDVQQLEw5Qb3J0U3dp
Z2dlciBDQTEXMBUGA1UEAxMOUG9ydFN3aWdnZXIgQ0EwHhcNMTQxMTExMTQ0ODA5
WhcNMzExMTExMTQ0ODA5WjCBijEUMBIGA1UEBhMLUG9ydFN3aWdnZXIxFDASBgNV
BAgTC1BvcnRTd2lnZ2VyMRQwEgYDVQQHEwtQb3J0U3dpZ2dlcjEUMBIGA1UEChML
UG9ydFN3aWdnZXIxFzAVBgNVBAsTDlBvcnRTd2lnZ2VyIENBMRcwFQYDVQQDEw5Q
b3J0U3dpZ2dlciBDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMRJ
qPzl4DylJD1jfAThvPx+hLGxrGrPLjq0wscutNFArehCxA6HLDcfxe8WN7JMn68b
3Kq+6oG5mv1kXvDfEbGNc2Gl6oGgoMuNGeu8EL54h1R90yZU158aRJxT9Y83k8PM
YE1FU5P4BCDOWVbNFlfL+WvUro3Gz/9i6xmdHW6LpWbDpIN13ZIOV6MNoDECd45e
SfQK3XEnbz4Fa4HfAEdOnYAw0XGzUqDAUPJsw3G5GXDUaORKyxgDaAJq/SEbX7Rn
3K5XdS5pncFtMboPIQv/V6PkkiJP4evHso+fCbcef2c/0hEwLyhNF9EIp8pJD/cI
uy/7FkLNWRUVskm0GPMCAwEAAaMTMBEwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG
9w0BAQsFAAOCAQEAtJ7caXybzpNW6i51rUZ9IE7wNIrrZLpM8FwCU1nP49lSUCYy
a0iUOdsjOEQfFN7WHwUMvNv3toC0H2PvP2umLp9tDow0SwJLSPDR/gIQsu2AV9bO
hl1atp9HV4KNbnW/05yguwFXBlwGl5XVL4rzJV07q1z2ltOTksvionqiUM9JsvWB
y07x8/ouofOH64JiYoDdgsAT64hiYmGvvdyt29/A85dowsiEIkOB21Wm1WszsnhH
NbyvAMsrxyYcFRvYs7VZsByWbnTs/m6z1f5wGoSq9bVXnvZ3RCrIJ/PHcCuIKoJ5
yq0Gm7Hkt5HUtK5pZfsKRDXybTKqOxUJE1vF8A==
-----END CERTIFICATE-----
```

- Therefore, you should rename the output certificate to `9a5ba575.0`

5. Now, you can copy the certificate to the device using `adb` by executing the following commands in order:
   - `adb root`
   - `adb remount`
   - `adb push 9a5ba575.0 /sdcard/`

```
C:\Program Files\Genymobile\Genymotion\tools>adb root

C:\Program Files\Genymobile\Genymotion\tools>adb remount
remount succeeded

C:\Program Files\Genymobile\Genymotion\tools>adb push 9a5ba575.0 /sdcard/
9a5ba575.0: 1 file pushed. 0.1 MB/s (1352 bytes in 0.016s)

C:\Program Files\Genymobile\Genymotion\tools>
```

6. After that, run a root shell using `adb shell`, and run the following commands:
   - `mv /sdcard/9a5ba575.0 /system/etc/security/cacerts/`
   - `chmod 644 /system/etc/security/cacerts/9a5ba575.0`

```
C:\Program Files\Genymobile\Genymotion\tools>adb shell
vbox86p:/ # mv /sdcard/9a5ba575.0 /system/etc/security/cacerts/
vbox86p:/ # chmod 644 /system/etc/security/cacerts/9a5ba575.0
vbox86p:/ #
```

7. At last, you have to fully reboot the device by using `adb reboot` or by restarting the VM.

```
C:\Program Files\Genymobile\Genymotion\tools>adb reboot
```

- Now, we can see the certificate as a **system-supplied**: