**Добавление собственного сертификата для всего приложения (Network Security Configuration)**

Начиная с версии *Android 24 (N)* в системе исключена возможность доверять сайтам по умолчанию. В связи с этим появилась возможность добавлять собственные сертификаты через конфигурационные файлы.

Более подробно:

1. <https://android-developers.googleblog.com/2016/07/changes-to-trusted-certificate.html>
2. https://developer.android.com/training/articles/security-config

**Пример загрузки страницы в WebView без и с добавлением сертификата:** Рассмотрим добавление сертификата на простом приложении с WebView. В качестве тестового домена был выбран URL: <https://3dsecmt.sberbank.ru/payment/se/keys.do>.

1) Если попытаться загрузить страницу через WebView *без* добавления сертификата, то в методе *onReceivedSslError* класса *WebViewClient* вернется ошибка *SSL\_UNTRUSTED* ***(1)***.

**Пример инициализации WebView и загрузки страницы:**

val webView = findViewById<WebView>(R.id.*webview*)

webView.*webViewClient* = object : WebViewClient() {

override fun onReceivedSslError(

view: WebView?,

handler: SslErrorHandler?,

error: SslError?

) {

when (error!!.*primaryError*) {

SslError.*SSL\_UNTRUSTED* -> Log.d("WEB\_VIEW\_EXAMPLE","The certificate authority is not trusted.”) // ***(1)***

SslError.*SSL\_EXPIRED* -> Log.d("WEB\_VIEW\_EXAMPLE","The certificate has expired.")

SslError.*SSL\_IDMISMATCH* -> Log.d("WEB\_VIEW\_EXAMPLE","The certificate Hostname mismatch.")

SslError.*SSL\_NOTYETVALID* -> Log.d("WEB\_VIEW\_EXAMPLE","The certificate is not yet valid.")

}

super.onReceivedSslError(view, handler, error)

}

}

webView.*settings*.*javaScriptEnabled* = true

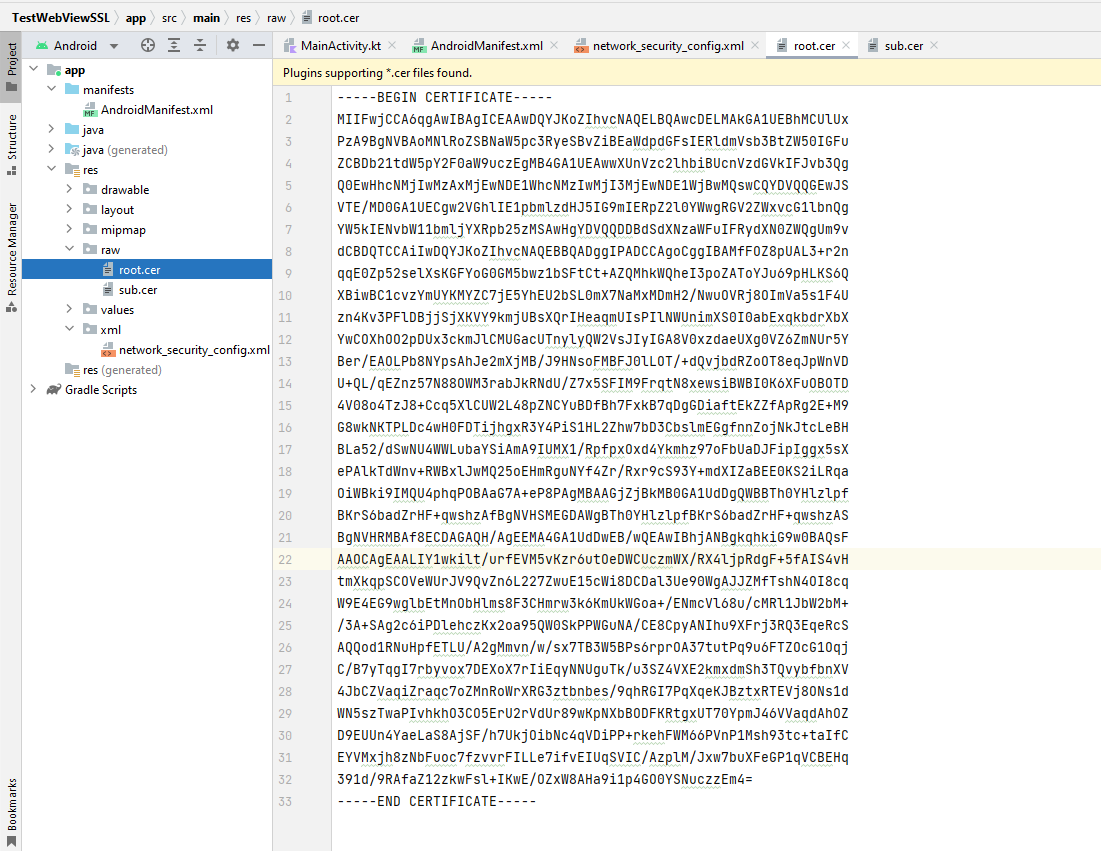
webView.loadUrl("https://3dsecmt.sberbank.ru/payment/se/keys.do")

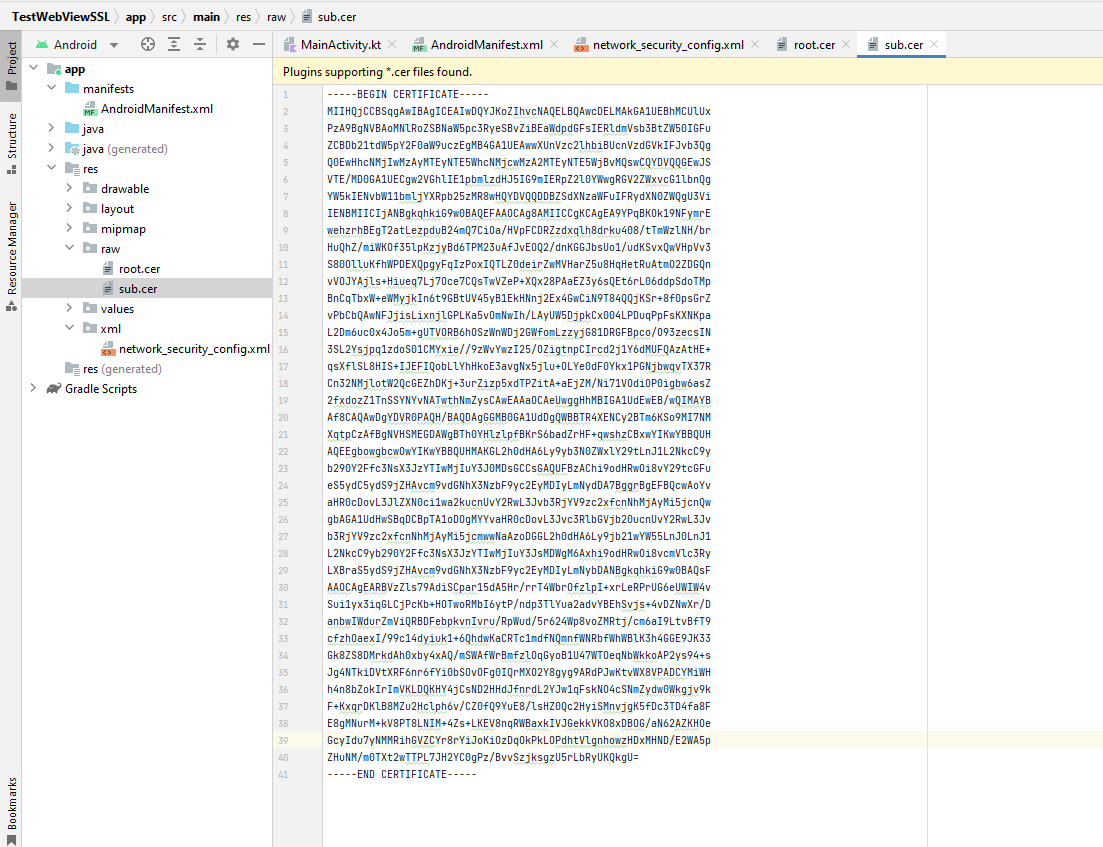
2) Для успешной загрузки страницы, необходимо выполнить следующие шаги для добавления сертификата.

**A)** В AndroidManifest добавить следующую строчку (доступна с версии API 24,N).

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **<uses-permission** android:name="android.permission.INTERNET" **/>**    **<application**  android:networkSecurityConfig="@xml/network\_security\_config" **/>**  ... |

**B)** Создать каталог raw с файлами сертификатов (сертификаты должен быть в РЕМ формате). **Необходимо наличие корневого и выпускающего сертификата. Сертификаты из примеры были скачены с сайта:** [**https://www.gosuslugi.ru/crt**](https://www.gosuslugi.ru/crt) **(Сертификаты для Android).** В примере: root – корневой, sub – выпускаюший.





**C)** Далее необходимо создать файл network\_security\_config.xml в директории xml. Содержимое файла приведено ниже.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <?xml version="1.0" encoding="utf-8"?>  **<network-security-config>**  **<domain-config>**  **<domain** includeSubdomains="true"**>**3dsecmt.sberbank.ru**</domain>**  **<trust-anchors>**  **<certificates** src="@raw/root"**/>**  **<certificates** src="@raw/sub"  **/>**  **</trust-anchors>**  **</domain-config>**  **</network-security-config>** |

Вы можете убрать опцию domain, и тогда сертификат будет работать для любых доменов на уровне всего приложения. Если хотим оставить как в примере выше, то убедитесь, что для промышленной сборки, будет добавлен домен: [https://securepayments.sberbank.ru](https://securepayments.sberbank.ru/)

**D)** После запуска приложения страница загрузится в WebView без ошибки SSL\_UNTRUSTED.

**Ручная проверка сертификатов**

Также возможен вариант с ручной проверкой сертификатов при первой попытке загрузки страницы в WebView. Для этого необходимо:

1. Необходимо подготовить TrustManager. **Необходимо наличие корневого и выпускающего сертификата. Сертификаты из примеры были скачены с сайта:** [**https://www.gosuslugi.ru/crt**](https://www.gosuslugi.ru/crt) **(Сертификаты для Android).** В примере: root – корневой, sub – выпускаюший.

Kotlin:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **private** **fun** initTrustStore(): TrustManagerFactory {  **val** cf: CertificateFactory = CertificateFactory.getInstance("X.509")  **val** certsIns = certNames.map { name -> readPemCert(name) }  **val** keyStore: KeyStore = KeyStore.getInstance(KeyStore.getDefaultType()).apply {  load(**null**, **null**)  certsIns.forEachIndexed { index, certIns ->  **val** cert = certIns.use { cf.generateCertificate(it) }  setCertificateEntry(certNames[index], cert)  }  }  **val** tmfAlgorithm = TrustManagerFactory.getDefaultAlgorithm()  **val** tmf = TrustManagerFactory.getInstance(tmfAlgorithm)  tmf.init(keyStore)  **return** tmf  } |

Java:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | **private** TrustManagerFactory initTrustStore() throws Exception {  **try** {  CertificateFactory cf = CertificateFactory.getInstance("X.509");  InputStream subIns = readPemCert(SUB\_CERTIFICATE);  Certificate sub = cf.generateCertificate(subIns);  InputStream rootIns = readPemCert(ROOT\_CERTIFICATE);  Certificate root = cf.generateCertificate(rootIns);  KeyStore keyStore = KeyStore.getInstance(KeyStore.getDefaultType());  keyStore.load(**null**, **null**);  keyStore.setCertificateEntry(SUB\_CERTIFICATE, sub);  keyStore.setCertificateEntry(ROOT\_CERTIFICATE, root);  String tmfAlgorithm = TrustManagerFactory.getDefaultAlgorithm();  TrustManagerFactory tmf = TrustManagerFactory.getInstance(tmfAlgorithm);  tmf.init(keyStore);  **return** tmf;  } **catch** (Exception e) {  **throw** **new** Exception("Error during TrustManagerFactory initialization");  }  } |

1. Реализовать собственный WebViewClient с предопределенным методом onReceivedSslError.

Kotlin:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | **private** **class** CheckServerTrustedWebViewClient(**private** **val** tmf: TrustManagerFactory) :  WebViewClient() {  **override** **fun** onReceivedSslError(view: WebView, handler: SslErrorHandler, error: SslError) {  Log.d("WEB\_VIEW\_EXAMPLE", "onReceivedSslError")  **var** passVerify = **false**  **if** (error.primaryError == SslError.SSL\_UNTRUSTED) {  **val** cert = error.certificate  **val** subjectDN = cert.issuedTo.dName  Log.d("WEB\_VIEW\_EXAMPLE", "subjectDN: $subjectDN")  **try** {  **val** f: Field = cert.javaClass.getDeclaredField("mX509Certificate")  f.setAccessible(**true**)  **val** x509 = f.**get**(cert) **as** X509Certificate  **val** chain = arrayOf(x509)  **for** (trustManager **in** tmf.trustManagers) {  **if** (trustManager **is** X509TrustManager) {  **val** x509TrustManager: X509TrustManager = trustManager **as** X509TrustManager  **try** {  x509TrustManager.checkServerTrusted(chain, "generic")  passVerify = **true**  **break**  } **catch** (e: Exception) {  Log.e("WEB\_VIEW\_EXAMPLE", "verify trustManager failed", e)  passVerify = **false**  }  }  }  Log.d("WEB\_VIEW\_EXAMPLE", "passVerify: $passVerify")  } **catch** (e: Exception) {  Log.e("WEB\_VIEW\_EXAMPLE", "verify cert fail", e)  }  }  **if** (passVerify == **true**) handler.proceed() **else** handler.cancel()  }  } |

Java:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | **class** CheckServerTrustedWebViewClient extends WebViewClient {  **private** final TrustManagerFactory tmf;  CheckServerTrustedWebViewClient(TrustManagerFactory tmf) {  **this**.tmf = tmf;  }  @Override  **public** **void** onReceivedSslError(WebView view, SslErrorHandler handler, SslError error) {  Log.d("WEB\_VIEW\_EXAMPLE", "onReceivedSslError");  boolean passVerify = **false**;  **if** (error.getPrimaryError() == SslError.SSL\_UNTRUSTED) {  SslCertificate cert = error.getCertificate();  String subjectDn = cert.getIssuedTo().getDName();  Log.d("WEB\_VIEW\_EXAMPLE", "subjectDN: " + subjectDn);  **try** {  Field f = cert.getClass().getDeclaredField("mX509Certificate");  f.setAccessible(**true**);  X509Certificate x509 = (X509Certificate) f.**get**(cert);  X509Certificate[] chain = **new** X509Certificate[]{x509};  **for** (TrustManager trustManager : tmf.getTrustManagers()) {  **if** (trustManager instanceof X509TrustManager) {  X509TrustManager x509TrustManager = (X509TrustManager) trustManager;  **try** {  x509TrustManager.checkServerTrusted(chain, "generic");  passVerify = **true**;  **break**;  } **catch** (Exception e) {  Log.e("WEB\_VIEW\_EXAMPLE", "verify trustManager failed" + e);  passVerify = **false**;  }  }  }  Log.d("WEB\_VIEW\_EXAMPLE", "passVerify: " + passVerify);  } **catch** (Exception e) {  Log.e("WEB\_VIEW\_EXAMPLE", "verify cert fail" + e);  }  }  **if** (passVerify) {  handler.proceed();  } **else** {  handler.cancel();  }  }  } |

Исходный код примера на обоих языках можно посмотреть во вложении.