



链滴

# openssl 获取服务端证书并导入到本地jdk证书库 + 忽略证书检测, 解决 '""'unable to find valid certification path to requested target

作者: [ymxf1](#)

原文链接: <https://ld246.com/article/1602393323951>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



在调用服务端接口时出现如下异常 **unable to find valid certification path to request target**，从异常信息中可以看到错误是因为找不到证书导致的。解决方案有两个：

- 正常的方案：当然是找到证书并安装证书。
- 暴力的方案：取消证书认证，信任所有请求链接。

第一种方案呢，要先获取服务端证书，然后安装到本地jdk的证书库。我们使用openssl来获取证书信，这里我写成了一个shell脚本，方便使用。

### get\_certs\_by\_domain\_port.sh

```
#!/bin/bash
#服务端地址
host=$1
#端口
port=$2

#证书名称
certName=$3

#获取证书
echo "Q"|openssl s_client -connect ${host}:${port} -servername ${host} -showcerts | openssl x
09 -outform pem > ${certName}.cert
```

如果需要循环获取多个证书可以用下面的脚本

```
#设置域名列表
hosts=(www.baidu.com www.csdn.net www.cnblogs.com)
#设置端口,HTTPS默认端口为443
port=443
#循环获取证书
```

```
for host in ${hosts[@]};do
  echo "Q"|openssl s_client -connect ${host}:${port} -servername ${host} -showcerts|openssl x
09 -outform pem > ${host}.crt
done
```

执行 `sh get_certs_by_domain_port.sh www.baidu.com 443 baidu` 即可在当前目录下生成名称为 `baidu.crt` 的证书。

```
[root@VM_0_13_centos ~]# sh get_certs_by_domain_port.sh www.baidu.com 443 baidu
depth=2 C = BE, O = GlobalSign nv-sa, OU = Root CA, CN = GlobalSign Root CA
verify return:1
depth=1 C = BE, O = GlobalSign nv-sa, CN = GlobalSign Organization Validation CA - SHA256 - G2
verify return:1
depth=0 C = CN, ST = beijing, L = beijing, OU = service operation department, O = "Beijing Baidu Netcom Science Technology Co., Ltd", CN = baidu.com
verify return:1
DONE
[root@VM_0_13_centos ~]# ll
total 8
-rw-r--r-- 1 root root 3589 Oct 11 12:46 baidu.crt
-rwxr-xr-x 1 root root 273 Oct 11 12:45 get_certs_by_domain_port.sh
```

使用下面的脚本，将获取到的证书导入到jdk证书库。

### import\_jdk\_certs.sh

```
#!/bin/bash
#将证书导入到jdk证书库
#证书别名
alias=$1
#证书文件名
fileName=$2
```

```
keytool -import -v -trustcacerts -alias ${alias} -file ${fileName} -storepass changeit -keystore $
JAVA_HOME}/jre/lib/security/cacerts
```

执行 `sh import_jdk_certs.sh baidu baidu.crt` 即可将证书导入到jdk证书库（需要使用root用户，否则会没有权限）。

```
[root@JD certs]# sh import_jdk_certs.sh baidu baidu.crt
Owner: CN=baidu.com, O="Beijing Baidu Netcom Science Technology Co., Ltd", OU=service operation department, L=beijing, ST=beijing, C=CN
Issuer: CN=GlobalSign Organization Validation CA - SHA256 - G2, O=GlobalSign nv-sa, C=BE
Serial number: 725878366e9f56e81d418848
Valid from: Thu Apr 02 15:04:58 CST 2020 until: Mon Jul 26 13:31:02 CST 2021
Certificate fingerprints:
MD5: 53:9B:07:1C:F7:D2:F0:2B:1C:85:CC:09:DA:11:86:E5
SHA1: FC:B4:0A:45:F2:7E:B3:91:AD:B1:3F:34:A6:25:96:87:35:CE:DD:CB
SHA256: 2E:D1:89:34:9F:81:8F:34:14:13:2E:BE:A3:09:E3:6F:62:0D:78:A0:50:7A:2F:A5:23:30:5F:27:50:62:D7:3C
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:
#1: ObjectId: 1.3.6.1.4.1.11129.2.4.2 Criticality=false
0000: 04 82 01 6A 01 68 00 76 00 5C DC 43 92 FE E6 AB ...j.h.v.\.C...
0010: 45 44 B1 5E 9A D4 56 E6 10 37 FB D5 FA 47 DC A1 ED.^..V..7...G..
0020: 73 94 B2 5E E6 F6 C7 0E CA 00 00 01 71 39 B5 3C s...q9.<
0030: BA 00 00 04 03 00 47 30 45 02 20 69 86 68 31 21 .....G0E. i.h1!
0040: B3 59 50 A3 D6 89 4A 97 45 66 08 69 14 93 DF C2 .Y)...J.Ef.i....
0050: 9E 72 87 A4 28 8A 1F E5 F5 E6 B2 02 21 00 C9 19 .r.(.....!...
0060: B4 5F 17 C8 A0 AE FE 76 65 97 57 BD BC 89 4A 6D .....ve.W...Jm
0070: 6F 11 AD 4A FF EF 81 E9 99 27 15 45 1B 01 00 76 o..J.....'E...v
0080: 00 6F 53 76 AC 31 F0 31 19 D8 99 00 A4 51 15 FF oSv.1.1....0..
0090: 77 15 1C 11 D9 02 C1 00 29 06 80 B2 08 9A 37 D9 w.....).....7.
00A0: 13 00 00 01 71 39 B5 3C E4 00 00 04 03 00 47 30 ...q9.<.....G0
00B0: 45 02 20 54 45 6C 30 4E 05 D9 FB D2 15 DE 68 66 E. TEL0N.....hf
00C0: 13 41 0E CD B8 3D D3 50 CB 46 CC DB 2C 8A 5A 7C .A...=.P.F...Z.
00D0: 36 3F 5E 02 21 00 BB 2F 10 6A 98 05 5A E9 1B 18 6?^!..!./..j..Z...
00E0: 4A A1 35 65 CF BF 86 5A 89 BC 4A 87 D4 78 77 27 J.Se...Z..J..xw'
00F0: E3 EF 43 9F BC 46 00 76 00 F6 5C 94 2F D1 77 30 ..C..F.v.\./..w0
0100: 22 14 54 18 08 30 94 56 8E E3 4D 13 19 33 BF DF *.T..0.V..M..3..
0110: 0C 2F 20 0B CC 4E F1 64 E3 00 00 01 71 39 B5 3C ./ ..N.d....q9.<
0120: AB 00 00 04 03 00 47 30 45 02 20 45 05 83 F1 28 .....G0E. E...{
0130: 47 61 95 FE 98 A9 BC 31 41 8E ED B2 48 1D A0 5B Ga.....1A...H..[
0140: A5 3F 13 E6 76 BC D2 E6 D3 69 3F 02 21 00 A9 CA .?.v...i?!....
0150: AB 0C DC 87 DF 25 20 D4 18 CC B0 15 E7 6F 82 F0 ....% .....o..
0160: 65 D9 3D 71 59 35 F8 5F A4 CE 2D 43 AC 5B e.=qY5...-C.[

#2: ObjectId: 1.3.6.1.5.5.7.1.1 Criticality=false
AuthorityInfoAccess [
|
| accessMethod: caIssuers
| accessLocation: URIName: http://secure.globalsign.com/cacert/qsorganizationvalsha2g2r1.crt
```

```
DNSName: *.bce.baidu.com
DNSName: *.eyun.baidu.com
DNSName: *.map.baidu.com
DNSName: *.mbd.baidu.com
DNSName: *.fanyi.baidu.com
DNSName: *.baidubce.com
DNSName: *.mipcdn.com
DNSName: *.news.baidu.com
DNSName: *.baidupcs.com
DNSName: *.aipage.com
DNSName: *.aipage.cn
DNSName: *.bcehost.com
DNSName: *.safe.baidu.com
DNSName: *.im.baidu.com
DNSName: *.baiducontent.com
DNSName: *.dlnel.com
DNSName: *.dlnel.org
DNSName: *.dueros.baidu.com
DNSName: *.su.baidu.com
DNSName: *.91.com
DNSName: *.hao123.baidu.com
DNSName: *.apollo.auto
DNSName: *.xueshu.baidu.com
DNSName: *.bj.baidubce.com
DNSName: *.gz.baidubce.com
DNSName: *.smartapps.cn
DNSName: *.bdtjrcv.com
DNSName: *.hao222.com
DNSName: *.haokan.com
DNSName: *.pae.baidu.com
DNSName: *.vd.bdstatic.com
DNSName: click.hm.baidu.com
DNSName: log.hm.baidu.com
DNSName: cm.pos.baidu.com
DNSName: wn.pos.baidu.com
DNSName: update.pan.baidu.com
```

```
#10: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 9E C9 79 D7 E9 5B AB 8A 16 CC 32 8E C6 99 E6 9F ..y..[....2.....
0010: 20 42 35 87 B5.
```

```
Trust this certificate? [no]: yes
Certificate was added to keystore
[Storing /usr/local/java/jre/lib/security/cacerts]
[root@JD certs]#
```

使用下面的脚本可以查看已经添加到jdk证书库里的证书信息，如果输出为空则代表没有找到证书。

## cat\_jdk\_certs.sh

```
#!/bin/bash
certName=$1
keytool -list -keystore "${JAVA_HOME}/jre/lib/security/cacerts" -storepass changeit |grep ${certName}
```

执行sh cat\_jdk\_certs.sh baidu 查看。

```
[root@JD certs]# sh cat_jdk_certs.sh baidu
baidu, Oct 11, 2020, trustedCertEntry,
[root@JD certs]#
```

第二种方案：取消证书认证，信任所有请求链接。

```
/**
 * 获取 HttpClient
 *
 * @param host
 * @param path
 * @return
 * @author 李振华
 */
private static CloseableHttpClient wrapClient(String host, String path) {
    CloseableHttpClient httpClient = HttpClientBuilder.create().build();
    if (host != null && host.startsWith("https://")) {
        return sslClient();
    } else if (StringUtil.isBlank(host) && path != null && path.startsWith("https://")) {
        return sslClient();
    }
    return httpClient;
}

/**
 * 在调用SSL之前需要重写验证方法，取消检测SSL 创建ConnectionManager，添加Connection
置信息
 *
 * @return HttpClient 支持https
 * @author 李振华
 */
private static CloseableHttpClient sslClient() {
    try {
        // 在调用SSL之前需要重写验证方法，取消检测SSL
        AppLogger.debug("在调用SSL之前需要重写验证方法，取消检测SSL");
        X509TrustManager trustManager = new X509TrustManager() {

            @Override
            public void checkClientTrusted(X509Certificate[] arg0, String arg1) throws Certificat
Exception {
            }

            @Override
            public void checkServerTrusted(X509Certificate[] arg0, String arg1) throws Certificat
Exception {
            }

            @Override
            public X509Certificate[] getAcceptedIssuers() {
                return new X509Certificate[0];
            }
        };

        SSLContext ctx = SSLContext.getInstance(SSLConnectionSocketFactory.TLS);
        ctx.init(null, new TrustManager[] { trustManager }, null);
        SSLConnectionSocketFactory socketFactory = new SSLConnectionSocketFactory(ctx,
            NoopHostnameVerifier.INSTANCE);
        // 创建Registry
    }
}
```

```

        RequestConfig requestConfig = RequestConfig.custom().setCookieSpec(CookieSpecs.
TANDARD_STRICT)
            .setExpectContinueEnabled(Boolean.TRUE)
            .setTargetPreferredAuthSchemes(Arrays.asList(AuthSchemes.NTLM, AuthSchemes
DIGEST))
            .setProxyPreferredAuthSchemes(Arrays.asList(AuthSchemes.BASIC)).build();
        Registry<ConnectionSocketFactory> socketFactoryRegistry = RegistryBuilder.<Connec
tionSocketFactory>create()
            .register("http", PlainConnectionSocketFactory.INSTANCE).register("https", socket
actory).build();
        // 创建ConnectionManager, 添加Connection配置信息

        PoolingHttpClientConnectionManager connectionManager = new PoolingHttpClientC
nnectionManager(
            socketFactoryRegistry);
        CloseableHttpClient closeableHttpClient = HttpClients.custom().setConnectionManage
(connectionManager)
            .setDefaultRequestConfig(requestConfig).build();
        return closeableHttpClient;
    } catch (KeyManagementException ex) {
        throw new RuntimeException(ex);
    } catch (NoSuchAlgorithmException ex) {
        throw new RuntimeException(ex);
    }
}

```