

当 Android Studio 构建过程中出现“服务器不支持 TLSv1.2/TLSv1.3”的问题，通常是由于以下原因导致的：

1. 服务器端仅支持较旧的 TLS 版本（如 TLSv1.0/TLSv1.1）。
2. Gradle 或 JDK 默认不支持 TLSv1.0/TLSv1.1，导致连接被拒绝。
3. 网络库或 Gradle 依赖库不兼容。

要解决这个问题，需要强制客户端降级到支持服务器的旧版本 TLS 协议，同时更新工具链或修改 Gradle 配置来适应旧协议。

解决方法

1. 强制启用 TLSv1.0/TLSv1.1 支持

在 JDK 8 及更高版本中，TLSv1.0 和 TLSv1.1 默认被禁用。如果服务器仅支持这些旧版本协议，可以显式启用：

修改 Gradle 配置

在项目的 gradle.properties 文件中添加以下内容，强制使用 TLSv1.0 或 TLSv1.1：

```
systemProphttps.protocols=TLSv1,TLSv1.1,TLSv1.2
```

2. 使用兼容的 JDK

确保项目使用的 JDK 版本支持所需的 TLS 协议。如果服务器只支持 TLSv1.0/TLSv1.1，建议使用 JDK 8，或配置 JDK 11 兼容旧协议：

- 在 **Android Studio** 中：
 - 打开 File > Project Structure > SDK Location。
 - 将 JDK Location 设置为 JDK 8 的路径。

3. 修改 Gradle Wrapper 配置

如果问题出现在 Gradle 构建过程中，可以更新 gradle-wrapper.properties 文件以兼容旧协议：

```
distributionUrl=https\://services.gradle.org/  
distributions/gradle-8.1-bin.zip
```

- 如果使用的 Gradle 版本较新（如 7.x 或 8.x）导致问题，尝试切换到 Gradle 6.x：

```
distributionUrl=https\://services.gradle.org/  
distributions/gradle-6.9.2-bin.zip
```

4. 降级 OkHttp 或网络请求库

如果问题出现在运行时网络请求中，可以降级 OkHttp 或网络库，强制启用旧版本协议：

配置 OkHttp 支持旧版 TLS

```
val client = OkHttpClient.Builder()  
    .sslSocketFactory(  
        // ...  
    )
```

```
Tls12SocketFactory(SSLContext.getInstance("TLSv1").apply
{ init(null, null, null) }.socketFactory),
    TrustAllManager()
)
    .hostnameVerifier { _, _ -> true } // 忽略证书校验, 仅调试时使用
    .build()
```

5. 验证服务器支持

- 检查服务器的协议支持情况:

使用工具（如 [SSL Labs](#)）分析服务器支持的 TLS 协议和加密套件。

- 服务器侧更新:

如果可能，建议联系服务器管理员升级 TLS 支持到至少 TLSv1.2，因为 TLSv1.0 和 TLSv1.1 已被广泛弃用，不再安全。

6. 使用代理工具调试

使用代理工具（如 Charles 或 Fiddler）拦截网络流量，验证请求协议和服务器响应，进一步分析问题。

不推荐的临时解决方法

如果只是为了快速通过调试构建，你可以忽略 HTTPS 校验，但这仅限于调试阶段，生产环境严禁使用：

忽略所有 SSL/TLS 校验

```

val trustAllCerts = arrayOf<TrustManager>(
    object : X509TrustManager {
        override fun checkClientTrusted(chain: Array<out
X509Certificate>?, authType: String?) {}
        override fun checkServerTrusted(chain: Array<out
X509Certificate>?, authType: String?) {}
        override fun getAcceptedIssuers():
Array<X509Certificate> = arrayOf()
    }
)

val sslContext = SSLContext.getInstance("SSL")
sslContext.init(null, trustAllCerts, SecureRandom())
val sslSocketFactory = sslContext.socketFactory

val client = OkHttpClient.Builder()
    .sslSocketFactory(sslSocketFactory, trustAllCerts[0]
as X509TrustManager)
    .hostnameVerifier { _, _ -> true }
    .build()

```

总结

- 首选：升级服务器支持 TLSv1.2 或更高版本。
- 如果服务器升级不可行，尝试通过 Gradle 配置或网络库降级支持 TLSv1.0/TLSv1.1。
- 如果问题仍然存在，请提供更详细的错误日志，以便进一步分析。