将整个项目拉下来后，依次完成以下步骤：

**01 创建python虚拟环境并下载依赖库**

在当前目录.../project下打开终端：

Python -m venv venv

然后激活虚拟环境：

venv\Scripts\activate

在虚拟环境中安装所需的依赖库：

pip install mitmproxy

pip install pyahocorasick

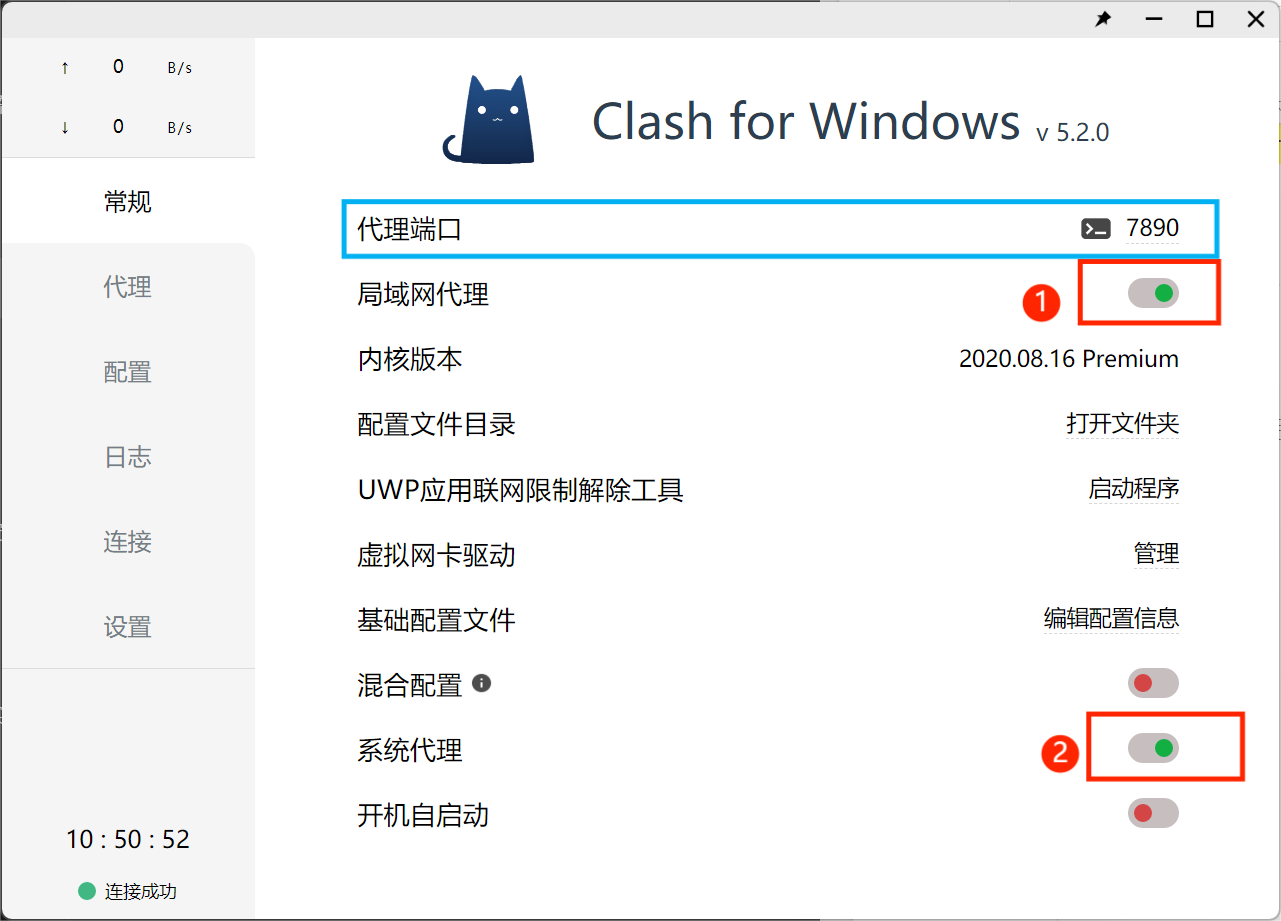
pip install flask

pip install requests

基本的代码部分已经完成，下面开始启动

**02 项目启动**

打开clash，开启VPN，并顺便查看clash的代理端口：



然后运行代码（在虚拟环境下）：  
mitmweb -s youtube\_proxy.py --set upstream\_cert=false --mode upstream:http://127.0.0.1:7890

这个命令是打开mitmproxy的图形化界面mitmweb，注意绿色部分和clash的端口号保持一致。

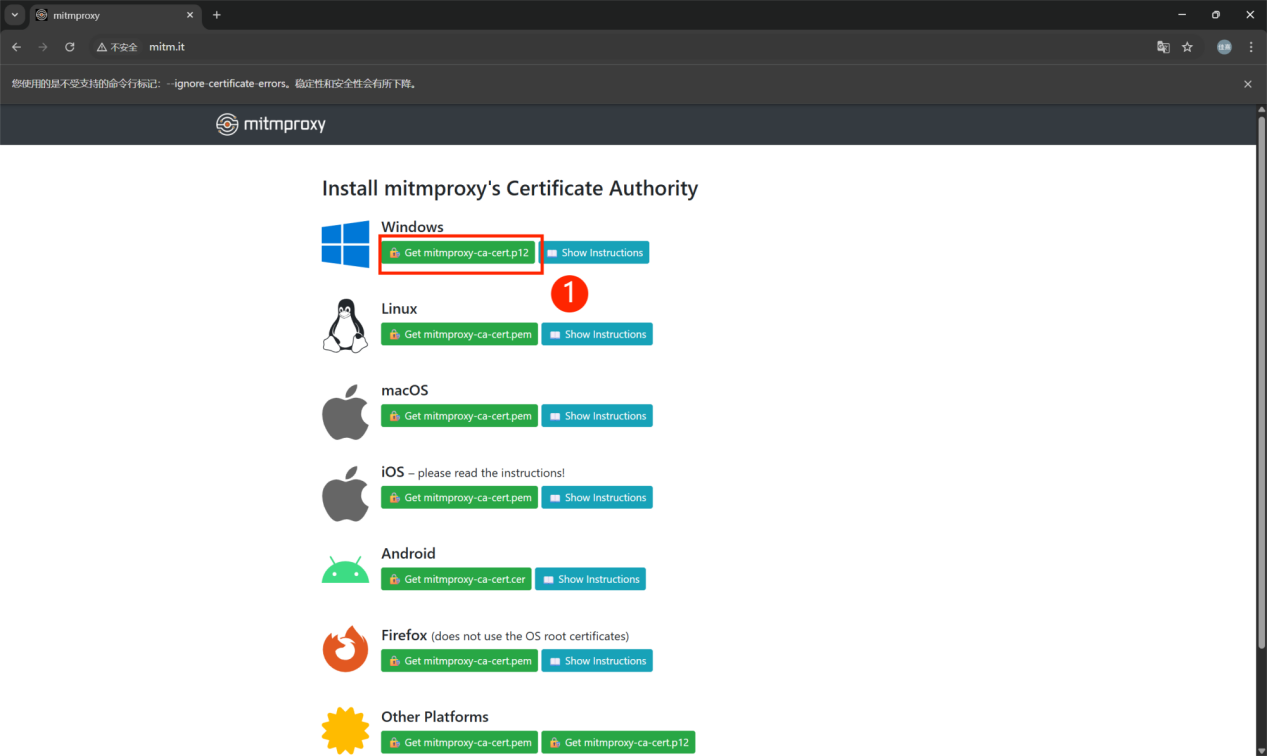
下面新开启一个cmd（注意一定是win+r打开的cmd，不能是“在终端中打开”，因为命令会格式错误），然后输入以下命令打开一个“特殊的”google chrome：  
"C:\ProgramFiles\Google\Chrome\Application\chrome.exe" --proxy-server=127.0.0.1:8080 --ignore-certificate-errors

注意绿色部分是Google chrome的实际地址，动态更改哈。

打开的一瞬间查看mitmweb界面其实就能发现已经捕到数据包了。

**03 下载证书**

Mitmproxy代理需要CA证书。在这个打开的google中访问http://mitm.it下载证书，步骤如下：



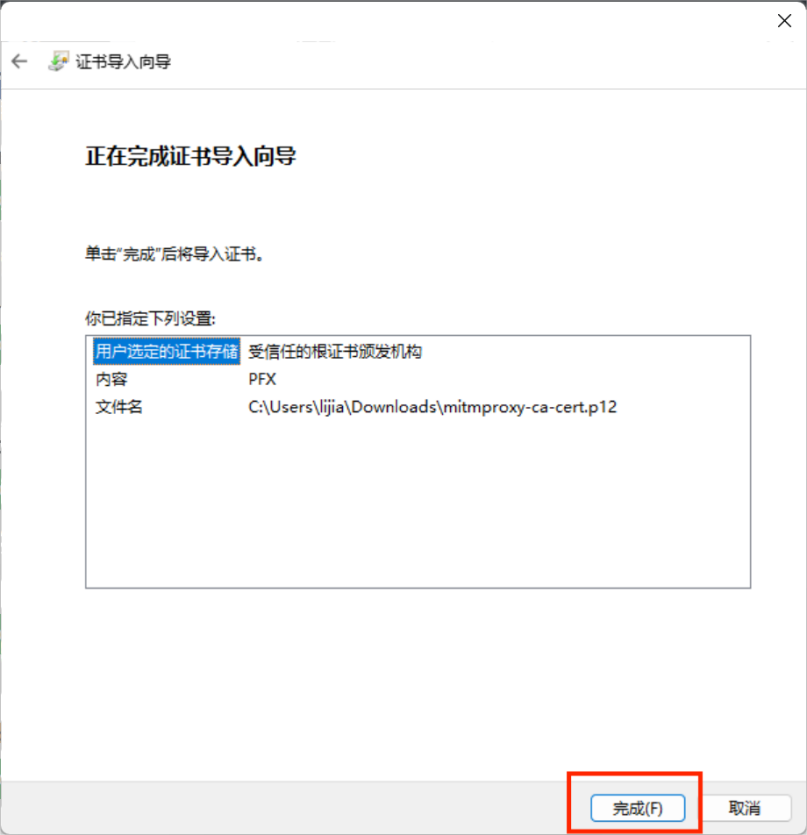
下载的文件可能会显示“不安全”，当它放屁。打开刚刚下载的文件：











到此，CA证书导入成功。

**04 运行测试**

在刚刚打开的google中访问YouTube，即可发现文件内生成的sensitive\_content\_analysis.txt文件中打印了解析出来的信息。

**附-简要原理及工作进度**

咱们的项目第一部分是捕包+解析。Mitmproxy是一个串行代理工具，它可以捕获、解析、修改、拦截、转发数据包，功能很牛逼。

正常的原理是在YouTube服务器和咱们的浏览器之间插入一个mitmproxy代理，进行双向的数据流处理。

**但是！**

YouTube是外网，所以在mitmproxy和YouTube服务器中间还要加一层clash代理以实现“科学上网”，因此你会发现打开mitmweb的命令行指令中有一个“upstream:http://127.0.0.1:7890”，代表将clash设置为上层代理。

并且，mitmproxy默认监听本地的8080端口，所以打开google chrome的命令行指令中有一个“--proxy-server=127.0.0.1:8080”。

目前咱们的项目已经能够捕包、解析，以及内容的初步解密。YouTube的数据加密方式是base64 + protobuf，其中protobuf负责核心内容加密，其解密需要逆向工程，非常麻烦。

暂时已经能够通过解密base64，挖掘出标题、描述等关键信息，已经能够支持下一步敏感词检测。