1. 安装charles
2. 运行charles
3. 安装charles证书：

help >> SSL Proxying >> Install Charles Root Certificate >> 安装证书

>> 下一步 >> 选中 将所有的证书都放入到下列存储 >> 浏览

>> 受信任的根证书颁发机构 >> 确定 >> 下一步 >> 选 是 >> 完成

1. 配置

HTTP的配置：

菜单项Proxy >> Proxy Settings

>> Port: 8888 默认8888，不修改 >> 勾选 Enable transparent HTTP proxying

>> Windows >> 不勾选 Enable Windows proxy 和 Enable Windows proxy at startup

设置完成，点击 ok

HTTPS配置：

菜单项Proxy >> SSL Proxy Settings

>> 选中 Enable SSL Proxying >> add >> Host设置 \* ， Port设置 443 >> 点击 确定

1. 抓包开启和关闭使用

 按钮 Start Recording

1. 清除所有的数据包使用  按钮
2. 保存数据使用  按钮
3. 打开一个已经保存好了的 charles数据包(chls后缀)，使用
4. 新建一个session，使用
5. 左边窗口有 Structure 和 Sequence

Structure：是结构化的数据显示，分 域名 显示， 这个窗口一般用于删除干扰数据

Sequence：是按时序的数据显示，主要的分析窗口就是这个

该窗口下，有多个列，分别是：

RC之前的没有名字的列：各种图标，不同类型的request，是不同的图标

RC：response的状态码

Method：request提交的方法

Host：目标服务器的域名或IP加端口

Path：url的path信息

start：request请求的起始时间

size：request的大小

duration：这次request所消耗的时间

status：request的状态，complete就是完成

info：该条记录的备注信息，一般不使用

在窗口下部有 overview、requests、response、summary、chart、notes 子窗口

overview：整个当条数据的全局信息，主要用于 复制 完整的 url

request：该窗口下有很多子窗口

headers：显示 请求行 和 所有的 请求头 的信息 非常重要的窗口

query string：url参数以表格的形式展示，即url中?后面的 key=value的数据

text：这个是form表单的文本显示，即 http 数据包 的 请求体 一般不查看

hex：请求体的 二进制显示 一般不查看

form：form表单的表格显示 很重要

json：json格式的 请求体

json text：json数据的字符串显示 很重要

multipart： 文件上传的 请求体

cookies：显示所有的 cookie 信息

raw：原始的通过socket发送的http数据包

response：等同于request，只不过信息是由服务器发送给浏览器的

Set Cookie：服务器在headers中进行的cookie的设置

summary：总结统计当前请求，几乎用不上

chart：图标展示，几乎用不上

notes：几乎用不上

1. 高级功能：

edit：在某条request中，右键，选择 edit，可以对该条request进行编辑，修改headers、cookies和参数，并且使用 execute 再次执行，查看response是否正常，以此确定修改的 值 是否对这次请求有作用

在自己程序提交的请求和浏览器抓包获取的请求之间，参数都一样，headers也一样，只是cookie不同时，这个功能就非常有用，可以选择 edit，逐一删除某个cookie，并且 execute执行，查看服务器的返回是否正常，通过不断重复这样的方法，可以得知哪几个cookie是关键的cookie，不提交则不能获取到正确的response

breakpoint settings： 在 proxy >> breakpoint settings 中设置，

通过  按钮进行开启和关闭， 这个功能类似 edit ，

只是edit是在正确提交的request上进行编辑，再次提交，查看结果

而 breakpoint 是在当前请求提交的基础上直接中断，进行编辑，提交，查看结果