使用说明

程序由两个部分组成，主体组件与配置文件。配置文件的名字为 console.ini，重命名会导致无法读取。
主体组件中的可执行文件 HttpConsole.exe 是整个程序的核心引擎，其它则为依赖项。
配置文件 console.ini 包含了程序参数和预设。可以通过修改配置文件来实现对程序参数的调整。

配置文件

基本结构
预设，就是一系列提前设定好的参数值。由于针对不同网站的攻击需要不同参数，每次都重新设置非常繁琐，因此引入了配置文件。我们事先在配置文件中定义不同的预设，然后在程序中直接加载想要的预设名，就可以一次性调整所有的参数为合适的值。配置文件的结构非常简单，一共只有两级：第一级[name]代表了预设名为 name，第二级 a=xxx 代表了该预设的参数 a 的值为 xxx。在下面这个类似的例子中：

```
47  [猪猡]
48  性别=雄
49  物种=猡
50
51  [猡马]
52  性别=雌
53  物种=马
```

其中紫色字体是第一级。方括号中的字符串即为这个预设的名字。方括号下面的项则为第二级。
在图中，显示了“猪猡”和“猡马”的参数。例如可以看到第 49 行，代表着猪猡的“物种”参数值为“猡”。
要新建预设，只需直接用文本编辑器打开配置文件。假设我们想添加一个名为“梁慈智”的预设，则我们只要先输入[梁慈智]，然后在下面键入“性别=男”：
[梁慈智]
性别=男
物种=霰弹枪

程序实际使用的配置文件非常类似于上面这个例子。如图：

```
31  [Default]
32  encode=gb2312
33  url=http://fccaa.cn/zf.asp
34  user=q
35  password=w
36  accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
37  ct=application/x-www-form-urlencoded
38  refer=http://fccaa.cn/
39  addition=
40  repeat=222365
41  sleep=false
42  proxy=true
43  threads=false
44  reply=true
```

方括号内是名字，方括号下是各个参数。在程序中输入名字，就可以读取它下面的内容，并给对应的参数设置相应的值。这就是配置文件的作用。

参数的意义(概述)

程序运行需要依靠参数，因此必须对参数进行正确设置。参数决定了我们要制裁哪个网站以及以何种方式制裁、如何绕开反制措施等。设置参数的第一步是理解它们。这个部分仅仅描述了每个参数的作用，至于如何设置才能正确有效，会在之后的部分解释。

- **encode**：这个参数是必要参数，也就是每一个预设下都必须包含这个参数。它决定了发送信息时使用的字符集。通常情况下使用 gb2312 可以正常运行。只有少数情况下需要设置成 utf-8。一般不需要改，设置好就行。
- **url**：网站的地址。同样是一个必要参数。它决定了数据发送的去向，通常也就是钓鱼网站的域名/钓鱼页面文件。
- **user** 和 **password**：必要参数。这两个参数分别决定了发送的用户名与密码的变量名。因为钓鱼网站接收数据时会用两个变量接收用户名/密码这两个值。我们要精确地把数据喂给网站，必须需要知道这两个名字，这样网站才能正确接收我们发送的假消息。
- **accept**：必要参数。只需要知道它是 http 请求头的一部分。
- **ct**：必要参数。同上。
- **refer**：必要参数。同上。
- **addition**：必要参数。由于某些网站除了接收之前所说的用户名与密码，还会接收一些其他的东西。某些情况下，如果缺少了这些东西，服务器将不会把我们的假消息记入数据库。这时我们就需要用 addition 参数加入这些不重要的额外参数，从而让消息顺利进入。

  注意：必要参数必须要出现，即使它不一定有值。例如，某些网站不需要额外参数就能灌入假信息。但它们的预设参数里仍然必须出现“addition=”，而不是省略不写。这一行表示它的 addition 为空值。除了 addition，所有的必要参数同样也必须出现，即使值为空。否则，程序将会提醒你配置文件没有正确设置好。
- **repeat**：表示重复发送的次数。必要参数。

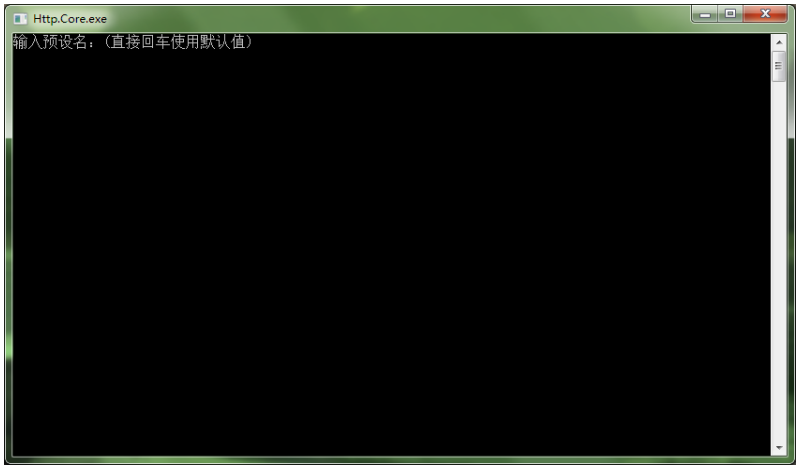
以下的参数都是非必要的。如果不设置，程序会使用默认值。

- **sleep**：表示睡眠模式，true 为开启，false 为关闭。开启睡眠模式时，程序每发送一次假消息就会挂起 90~7000 秒。默认情况下关闭。
- **proxy**：表示代理模式，true 为开启，false 为关闭。默认是关闭的。开启代理模式时，程序将通过本机的 8118 端口发起请求。正确使用这个模式的前提是配置好 TOR 和 privoxy。
- **threads**：表示并发模式。默认情况下关闭。开启并发模式时，会使用多个线程同步发送假消息。
- **strong**：表示强密码模式。默认情况下关闭。打开时，程序的密码生成器只会输出同时包含数字、大写、小写字母的强密码。因为某些钓鱼网站(尤其与苹果相关的)会验证密码强度并阻止不符合要求的密码。遇到这样的网站时，只需要把这个参数设置成 true 就可以通过。
- **email**：表示邮箱模式。默认情况下关闭。打开时，程序的用户名生成器会输出假邮箱（默认情况下是假 QQ 号）。
- **reply**：表示回复模式。默认情况下打开。默认情况下，程序每次发送完假消息，会等待

服务器的响应直到连接超时(30 秒)。如果关闭这个参数，程序将进入渣男模式——仅发送大量请求，发送完立刻关闭连接，不等待或处理任何回复。这样做的好处是，由于不用等待回复，假消息发送的速度大大提升，如果再打开并行模式，每秒钟能输出三千条以上的虚假请求。弊端则是看不见服务端的回应，有可能服务端报错了都不知道。

使用预设进行快速攻击

完全理解上面的参数需要一些时间。但是如果只是单纯地利用写好的配置文件进行攻击，则非常简单。
打开程序

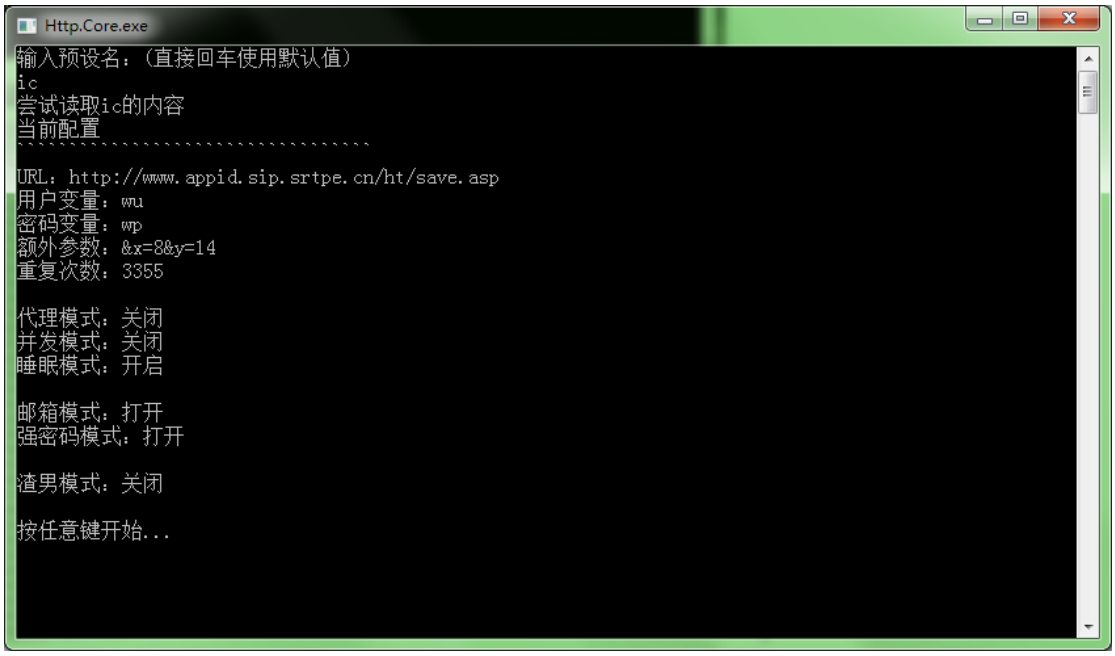


这里我们需要输入预设名。此时用文本编辑器打开配置文件，看一下里面有哪些预设：

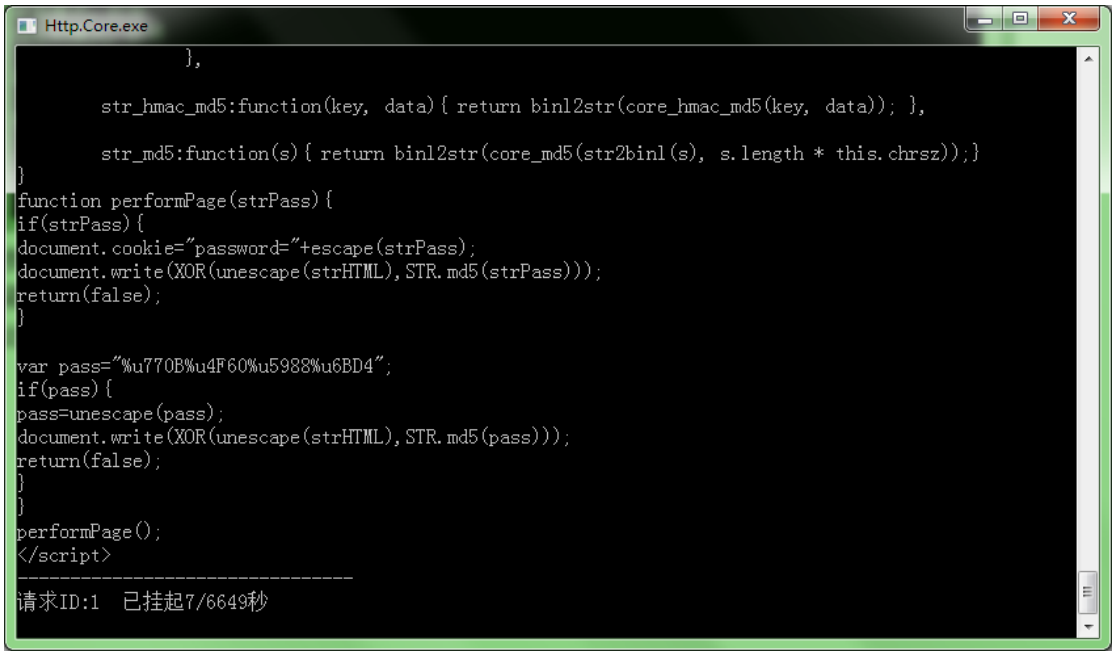
```
[ic]
#仿冒iCloud
encode=gb2312
url=http://www.appid.sip.srtpe.cn/ht/save.asp
user=wu
password=wp
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
ct=application/x-www-form-urlencoded
refer=http://www.appid.sip.srtpe.cn/q1y8vlctxx7298xc18af.asp?q1y8vlctxx7298xc18af
ck=
addition=&x=8&y=14
repeat=3355
sleep=true
proxy=false
threads=false
strong=true
email=true


[Default]
#仿冒王者荣耀充值
encode=gb2312
url=http://fccaa.cn/zf.asp
user=q
password=w
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
ct=application/x-www-form-urlencoded
refer=http://fccaa.cn/
addition=
repeat=222365
sleep=false
proxy=true
threads=false
reply=true
```

可以看到有两个预设。由于第一个预设看着让人不爽，我们把目标设定为这个仿冒 iCloud 的钓鱼网站。在程序中输入对应的预设名，此处为“ic”，然后回车：



参数被正确读取了。因为预设里已经写好了，所以这些参数都是适用于这个网站的。例如，注意到邮箱模式已经打开，因为这是个苹果钓鱼站，盗取的是 Apple ID 而不是 QQ 账号。此处每个模式的意思，在上一个部分都解释过。接下来按任意键就可以开始攻击。



按任意键几秒后，窗口内出现了大量字符。这些是来自服务器的回应。因为预设里开启了睡眠模式，所以程序会挂起一段时间。时间到后会自动继续。这一次攻击会持续到关闭窗口/完成预设的次数为止。这就是一次简单的攻击。

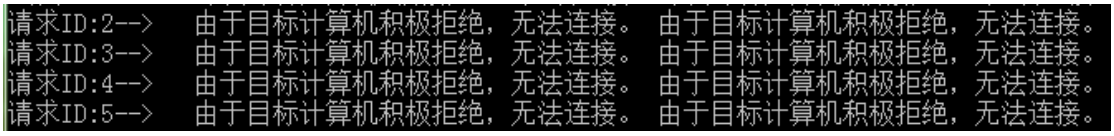
对结果的判断

如果这次假信息的灌入没有发生什么差错，服务器的回应会是网页的代码，通常情况下是 HTML 代码(包含大量<></>符号，很容易辨别)。如果你收到了一串类似这样的回复，那么

攻击是有效的，程序正在向钓鱼网站精准灌注假信息。

第一种失败：网络错误

这种情况下，程序或服务器会告诉你错误类型。例如：



可能造成这个问题的原因有很多，需要具体情况具体分析。

第二种失败：服务器识别并拒绝

简单来说就是，你的意图被服务器的过滤器发现了。对于这种失败，不同的服务器可能会返回不同的内容，这主要取决于该钓鱼网站站长的弱智程度。例如，有的服务器会返回“IP address forbidden”这样非常正常直接的句子，也有的服务器会返回

这样的弱智内容。总之，这时你需要考虑以下两点：[以下内容可以跳过]

1.你是如何被识别的

通常来说，被封禁 IP 很有可能是因为没有打开睡眠模式，甚至有可能打开了并行模式。这种情况下，由于你短时间发送多个请求，触发了服务器的保护机制，因而 IP 被封禁。这可以简单地通过打开睡眠模式来避免。但是如果你已经被封禁了 IP，程序已经无法再正常连接到该服务器，所以你仍要先考虑修改 IP。

2.如何修改 IP

VPN 是一个可行的手段，但是要保证效果，需要每隔一段时间不断地手动修改节点。
原因是深层次的。首先，程序本身已经内置了一个非常粗糙却意外有用的、隐藏真实 IP 的机制。不少网站，由于其获取 IP 的代码写得非常烂，会被这个机制骗过。在这种网站看来，这个程序的每一条假消息的来源 IP 都是随机的。

而你会被该网站的服务器封禁，就意味着这个网站不同于上述劣等网站，它必定拥有能获取真实 IP 的代码，否则你不可能触发它的保护机制，它也不可能对你做到成功封禁。

既然它获得你的真实 IP 轻而易举，那么从数据库中分辨你发送的假消息也不可能难到哪里去。如果使用 VPN，即使打开了睡眠模式，不会触发保护机制，但自始至终你的 IP 都没有改变，网站的运行者将很容易从数据库中过滤来自你的 VPN IP 的假消息。要保证你的假内容对他产生干扰，你只能手动更换节点。

因此更换 IP 方面，有更加优秀的解决方案，即 TOR+privoxy。

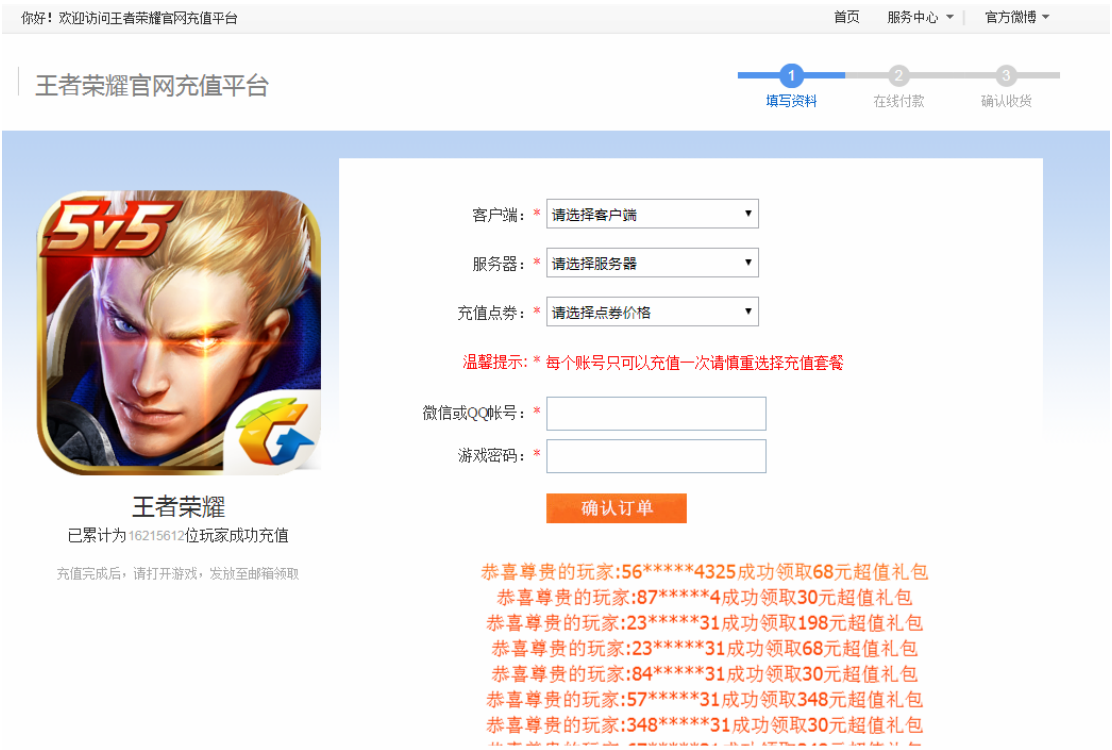
但是，由于 TOR 提供较高的匿名性，GFW 对 TOR 服务器进行了过滤，因此在中国大陆，要成功连接 TOR 网络必须要通过前置代理，此处只能按下不表。VPN 属于比较可行的方案，尽管制作精良到需要用 VPN 的钓鱼网站本身就已经非常少见。

如果只想用现成的预设文件进行攻击，那么教程到这里就结束了。下面的内容则与更进阶的使用有关。

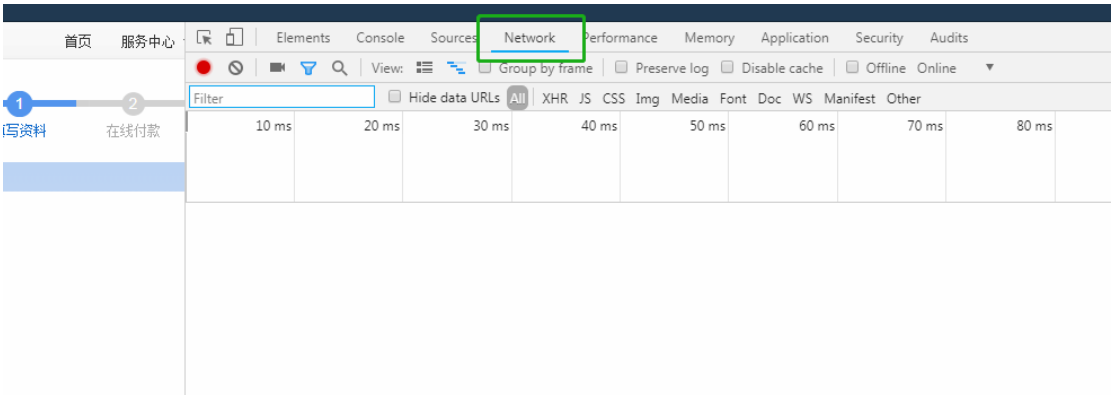
自己编写预设

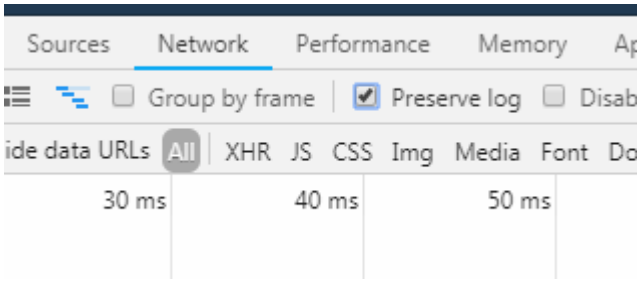
如果我们由于某种原因遇到了新的钓鱼网站并想制裁它，那么就需要自己编写一个全新的预设。知道预设的编写过程也能帮助我们彻底理解每个参数的意义。为此，我们需要 Chrome 内核的浏览器一枚。
这里以钓鱼网站 http://fccaa.cn 为例子。

首先我们需要找到登陆页面：



按 F12 打开开发者模式。浏览器的右边/下面会出现一个控制台。我们的目标是用这个功能来抓取数据包，想办法找出浏览器到底给服务器发了什么数据。点击控制台上的 Network/网络选项卡，如图

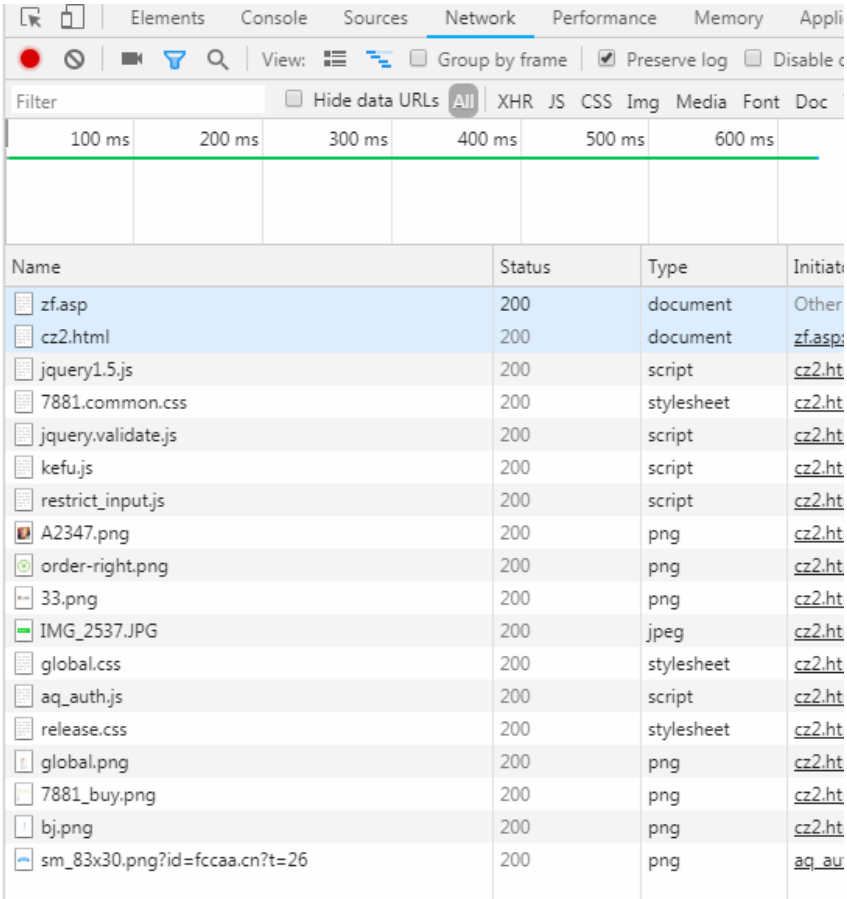




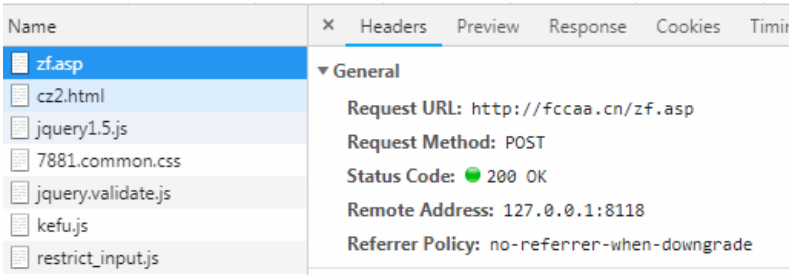
这个 Preserve log 选项需要勾上，因为有些网站会多次跳转，勾选这个复选框可以保证跳转时我们的有用数据不会被清空。

左边的红点表示浏览器已经开始监控流量。接下来，我们模拟一名普通的受害者进行登陆操作。

填好这些没用的数据，然后点击提交，右边的网络监视器会记录下所有的网络活动信息。



查看第三列，我们需要关注的只是 Type 为 document 的内容。在上图中只有前两个。这些就是登陆页面的页面文件。左键点击可以查看这个活动的详细信息。

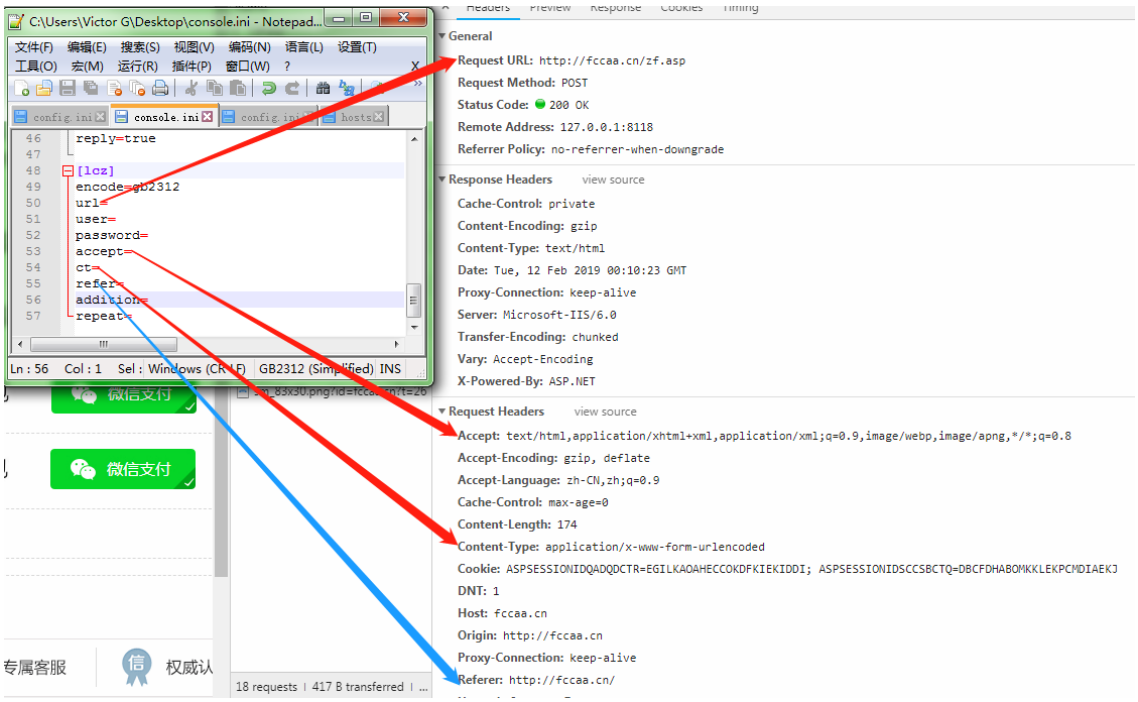


注意到右边 Request Method 的值为 POST。这意味着这个请求就包含着我们想要的数据。HTTP-POST 方法是一种向指定地址发送数据包的方法，绝大多数钓鱼网站都通过这个方法获得账号和密码。这个 POST 请求里包含了我们写预设需要的参数的信息

留着谷歌浏览器不动，打开配置文件。在标准配置文件的第一个区域，有一个叫[Template]的预设。这是一个模版，包含了所有的必要参数。这样在编写新预设时，只需要复制模版到最下方，修改名字，然后依次填入相应参数，最后再酌情添加修改其他非必要参数即可。

```
[Template]
encode=gb2312
url=
user=
password=
accept=
ct=
refer=
ck=
addition=
repeat=
```

在这里，为预设取一个简短易记的名字来替换“Template”，本例中取名 lcz。部分参数与浏览器中的数据对应关系如下：

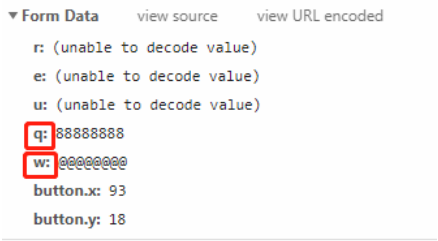


这个过程可以简单地通过复制粘贴完成。

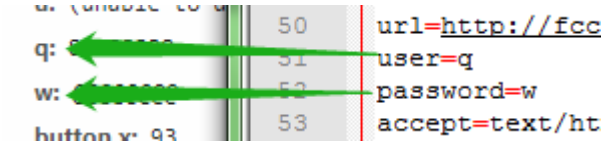
按照上述对应关系，填好参数，还有四个参数没有填。下面详细说明。

```
[lcz]
encode=gb2312
url=http://fccaa.cn/zf.asp
user=
password=
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
ct=application/x-www-form-urlencoded
refer=http://fccaa.cn/
addition=
repeat=
```

User 参数和 password 参数可以在浏览器网络监视器数据的最下方，即 From Data 部分找到。这里是浏览器本次提交给服务器的数据。其中应该能够找到你刚才填写的假用户名和假密码，如图



在本例中，假 QQ 号前面是 q，假密码前面是 w，因此我们分别在 user 和 password 参数后填写 q 和 w。不同的钓鱼网站，这两个变量的名称可能会有很大的区别。



此外还有其他参数(r, e, u, button.x, button.y)。此时我们还不知道服务器是否会因为这些参数的缺少而拒绝我们的消息，所以我们暂时不去理会。因此，addition 参数我们暂时留空。运气好的话，服务器不会回复任何错误。

在 repeat 参数项输入我们想要的重复次数，例如两万次。接着就可以设定非必要参数了。鉴于这个网站制作非常粗糙，有理由假定它没有什么精巧的保护机制。因此打算使用比较暴力的方式进行制裁：

打开并行模式：在下面输入 threads=true

```
threads=true
```

打开渣男模式(关闭等待回应)：在下面输入 reply=false

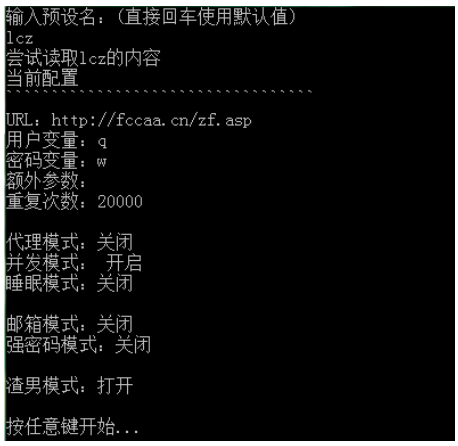
```
threads=true
reply=false
```

至于其他模式，如之前所说，默认情况下是关闭的，而且是非必要参数，因此就不用特殊设置。此时我们的预设 lcz 就成功完成了：

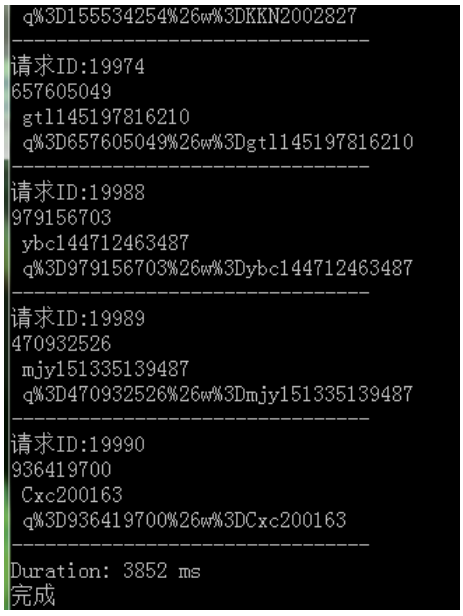
```
[lcz]
encode=gb2312
url=http://fccaa.cn/zf.asp
user=q
password=w
accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
ct=application/x-www-form-urlencoded
refer=http://fccaa.cn/
addition=
repeat=20000
threads=true
reply=false
```

当一个预设经常使用，可以把它的名字更改为 Default，这样每次需要输入预设名时直接回车，加载的将是 Default 区域内的参数，比较方便。

我们可以在程序中使用刚刚写好的预设。就像使用其他预设一样：



程序正确读取了预设，并且具体配置和我们的想法一样。下面是运行效果。



不愧是渣男+并发模式，两万个假消息仅用了 3.8 秒就发送完毕。平均每秒发出 5263 次请求。当然，由于关闭了回复，实际上有可能这两万个请求都是失败的，服务器全部没输入数据库。看不到回复，我们没法知道每一个请求的结果是什么。这是关闭 reply 的弊端。