链滴

有道翻译小爬虫

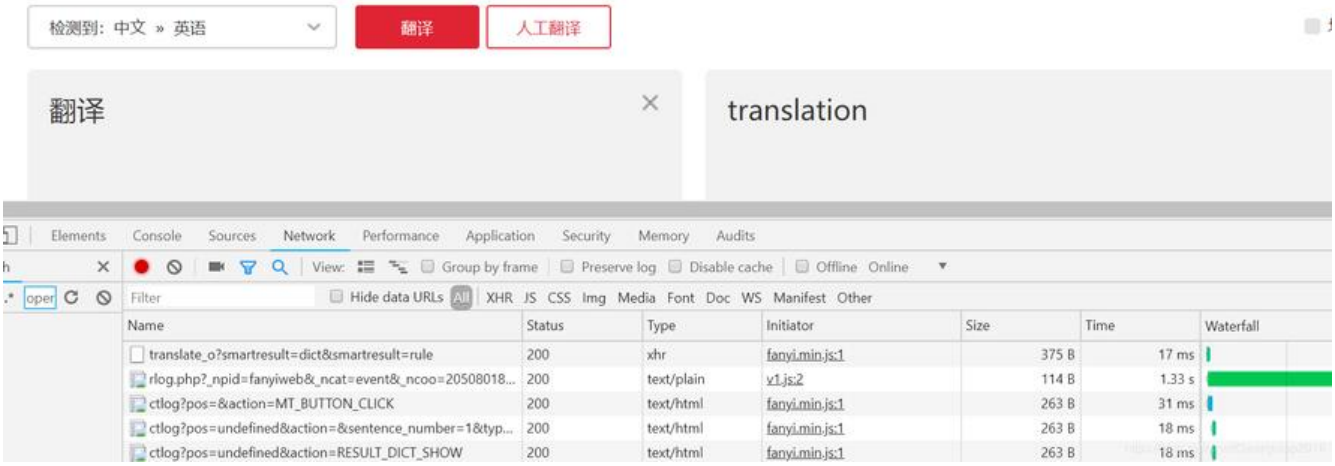
作者：kanadeblisst

原文链接：https://ld246.com/article/1565597605677

来源网站：链滴

最近想开始学习一下js逆向，发现有道翻译的js真的是非常适合没有玩过js逆向的小白练手。破解的话全没有任何技术含量，只需要你会浏览器断点和Ctrl + F就行。甚至连js代码都看不懂也没关系。

我们先输入一个词(翻译)，然后看Network里的请求，很明显第一条就是翻译的接口了。因为下面的是加载的图片链接。当然，我们可以点击请求，查看response或者preview里的内容确认一下有没有们的数据。



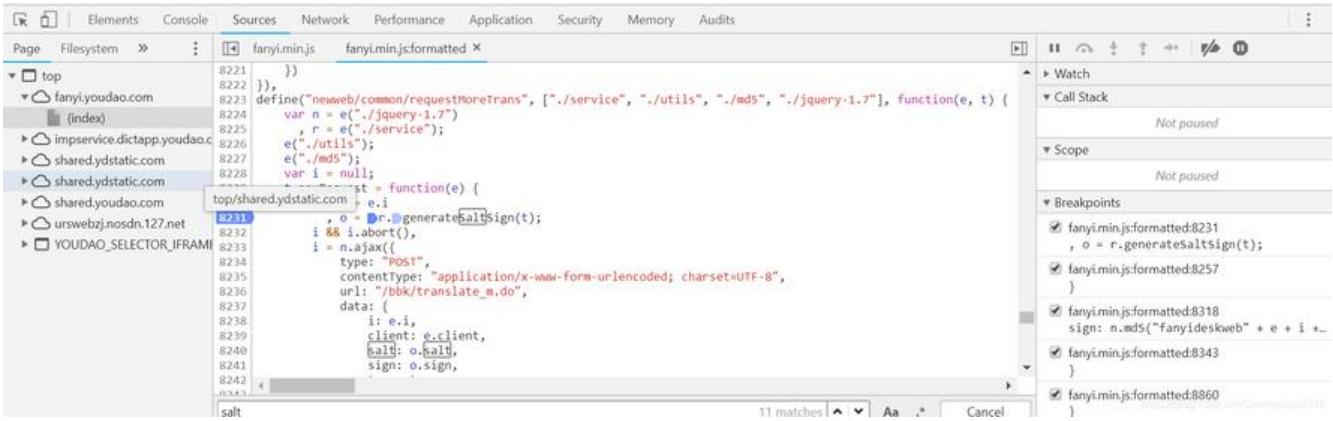
接着我们看第一条请求登headers里的内容，需要的参数有这些：

```
i: 翻译                                  #要翻译的词
from: AUTO                                 # 固定值，不管他
to: AUTO                                  # 固定值，不管他
smartresult: dict                          # 固定值，不管他
client: fanyideskweb                       # 固定值，不管他
salt: 15638634698882                       # 未知参数，看起来像时间戳
sign: c4e7935237470f6e9d022309ef09a2ce     # 未知参数，看起来像md5
ts: 1563863469888                          # 未知参数，看起来像时间戳
bv: 1e9538f95b23257ede9acdc941c8e1f8       # 未知参数，看起来像md5
doctype: json                              # 固定值，不管他
version: 2.1                               # 固定值，不管他
keyfrom: fanyi.web                         # 固定值，不管他
action: FY_BY_CLICKBUTTION                 # 固定值，不管他
```

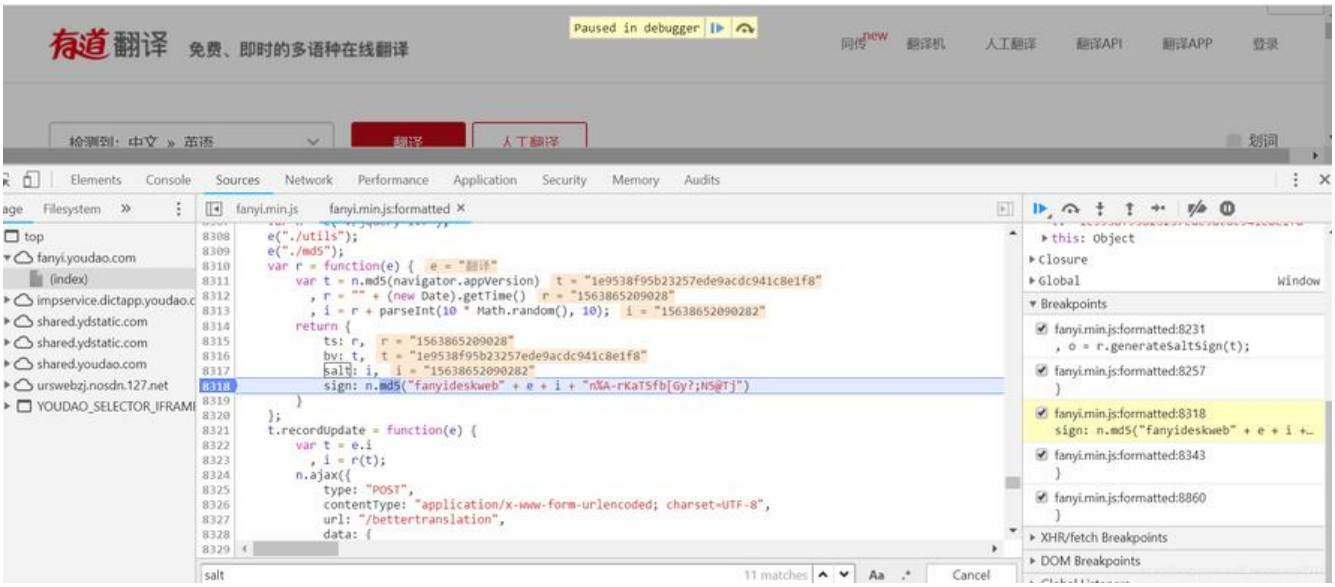
这样看起来一共有四个未知参数，只要要知道这四个未知参数是怎么得到的，就可以模拟请求这个UR实现翻译了。

所以我们刷新一下页面，在search栏全局搜索一下salt，只有fanyi.min.js含有这个关键字。我们格式代码(点一下js文件左下角的大括号)，再在js文件里Ctrl+F搜索一下salt。发现了一些代码很可疑，基

可以确认salt就是这个js文件生成的，为了找到它具体的生成代码，需要在sources里找到并打开(当然可以直接在Network里右键这个js然后选择open in sources panel)。我们将出现salt的行全部打上断(点击行号，会变蓝色)。有时候点击行号没有反应，实际上已经打上断点，只是在下面几行的某个位，你可以观察右边的Breakpoints里的内容，这就是你打了断点的位置。



打上断点之后应该干什么呢？当然是重新触发一次事件，执行这个js文件，我们打断点才有意义。只要在点一下翻译按钮就可以触发断点了。弹出的paused in debugger后面有两个按钮，第一个大概用是执行到下一个断点的位置。第二个是不进入函数一步一步的执行js。



我们当然是选择第一个了，于是就到了如上图的位置。这样所有的参数是不是一目了然。就算不知道j的一些内置函数，看显示的值就知道它是什么功能了。鼠标移到某个变量，就可以看它的当前值。比你不知道navigator.appVersion是什么，可以吧鼠标移到这个变量。

```
var r = function(e) {
        var t = n.md5(navigator.appVersion)
          , r = "" + (new Date).getTime()
          , i = r + parseInt(10 * Math.random(), 10);
        return {
            ts: r,
            bv: t,
            salt: i,
            sign: n.md5("fanyideskweb" + e + i + "n%A-rKaT5fb[Gy?;N5@Tj")
        }
    };
```

我们用Python得到上面的那些参数：

```
import hashlib

# bv也就是t是请求头的MD5值，注意这里登请求头没有Mozilla/
ua = '5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0
3626.121 Safari/537.36'
hashlib.md5(ua.encode()).hexdigest()

# ts也就是r是当前的时间戳，不过js的时间戳是13位
t = time.time()
ts = str(int(t * 1000))

# salt也就是i是当前时间戳加上一个随机0-9的数，为了不麻烦我们可以直接这样
salt = str(int(t * 10000))

# sign是"fanyideskweb" + e + i + "n%A-rKaT5fb[Gy?;N5@Tj"的md5值，其中e是要翻译的单词，
就是salt。
sign = hashlib.md5(("fanyideskweb" + word + salt + "n%A-rKaT5fb[Gy?;N5@Tj").encode()).he
digest()
```

有了这样的准备，我们只需要word，就可以请求那个接口得到数据了。

URL：http://fanyi.youdao.com/translate_o?smartresult=dict&smartresult=rule

headers：

Host: fanyi.youdao.com

Origin: http://fanyi.youdao.com

Referer: http://fanyi.youdao.com/

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko
Chrome/72.0.3626.121 Safari/537.36

但是当我使用requests发起请求的时候还是显示{errorcode:50},想了想应该是cookie没有加上。加上c
okie后数据就正常了（这里出现了一个小小的问题，当前浏览器登cookie没用，只有清空浏览器所有
据之后的cookie才有用。看了一眼这两个cookie的区别，开始的那个携带了某些信息，好像是我网易
号的信息，不清楚为什么这个cookie得不到数据，而清空之后的cookie则完全是一个游客cookie）。

如何实现更自动化一点？因为这里的cookie是我们从浏览器复制过去的，怎么让程序得到。模拟登录
下有道？大可不必，因为这里游客cookie就可以得到数据。所以没必要登录。而游客cookie只需要请
一次主页：http://fanyi.youdao.com/ 就可以得到cookie了，之后可以使用requests的session来
作cookie。

还有一点，sign的最后一个很长的字符串其实是会变化的。那么隔一段时间又要查看一次它登值然后
加到程序中吗？当然不需要，我们可以请求fanyi.min.js这个文件，然后正则匹配出这个字符串就可以
。

代码没有写，很简单就不打算写了。