2019年CTF4月比赛记录（一）：ENCRYPT CTF 部分Web题目writeup与重解

原创

極品一═☆宏 于 2019-04-05 10:22:53 发布 2786 收藏 1

分类专栏： 2019年CTF比赛—4月赛 CTF_web

本文链接：https://blog.csdn.net/qq_43214809/article/details/89022757


2019年CTF比赛—4月赛 同时被 2 个专栏收录

3 篇文章 0 订阅

订阅专栏


CTF_web

13 篇文章 0 订阅

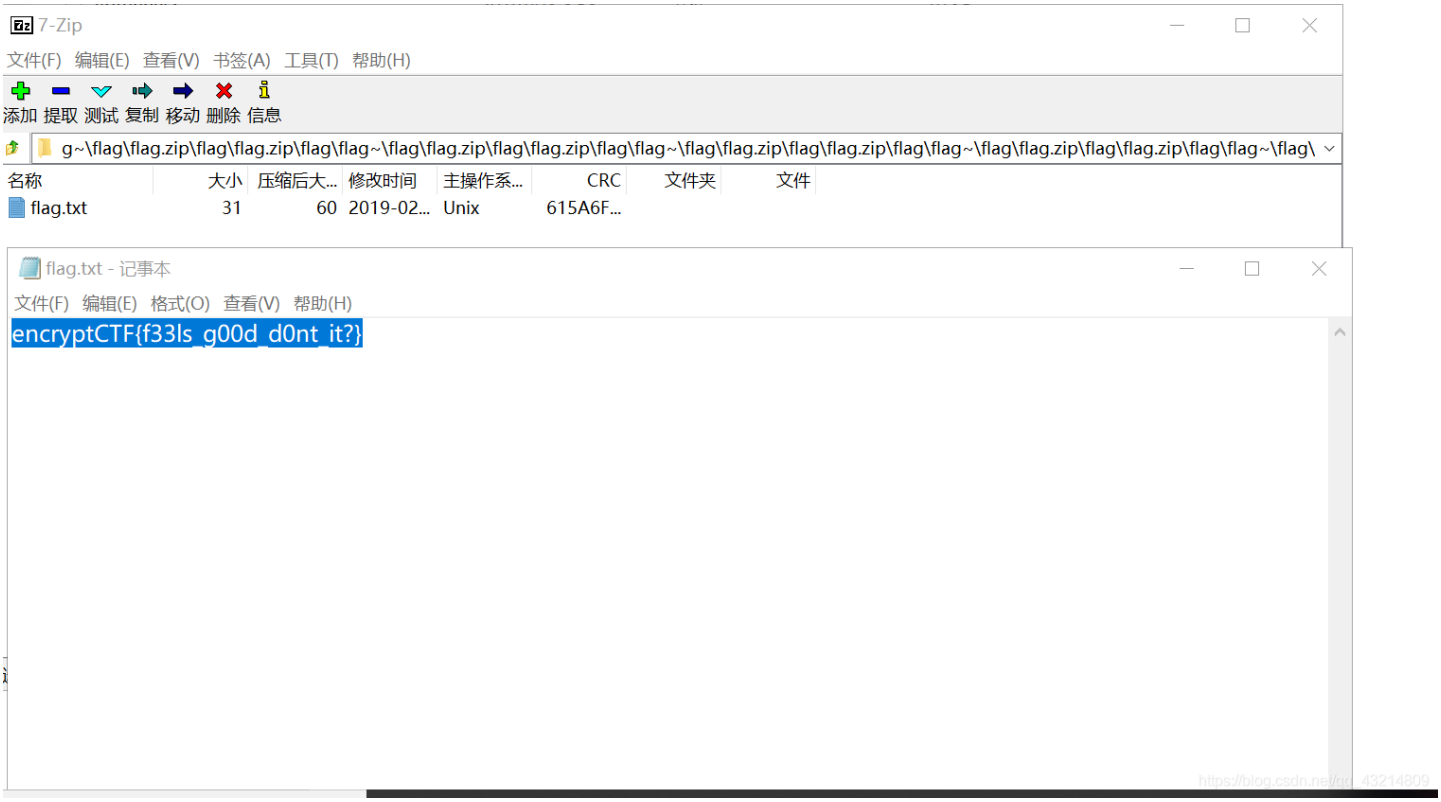
订阅专栏

简简单单做些题，踏踏实实做些事

时间：2019年4月2日至4月4日

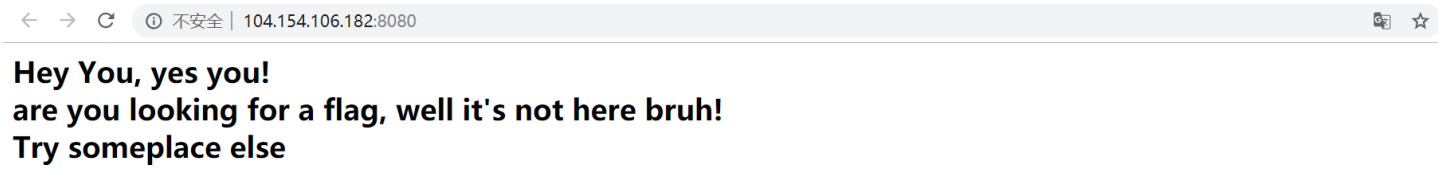
一、ENCRYPT CTF—Journey（writeup)：

两个水题，一个套路，直接打开文档，一直打开压缩包，直到出现flag.txt，没啥技术含量，就当签到题了：



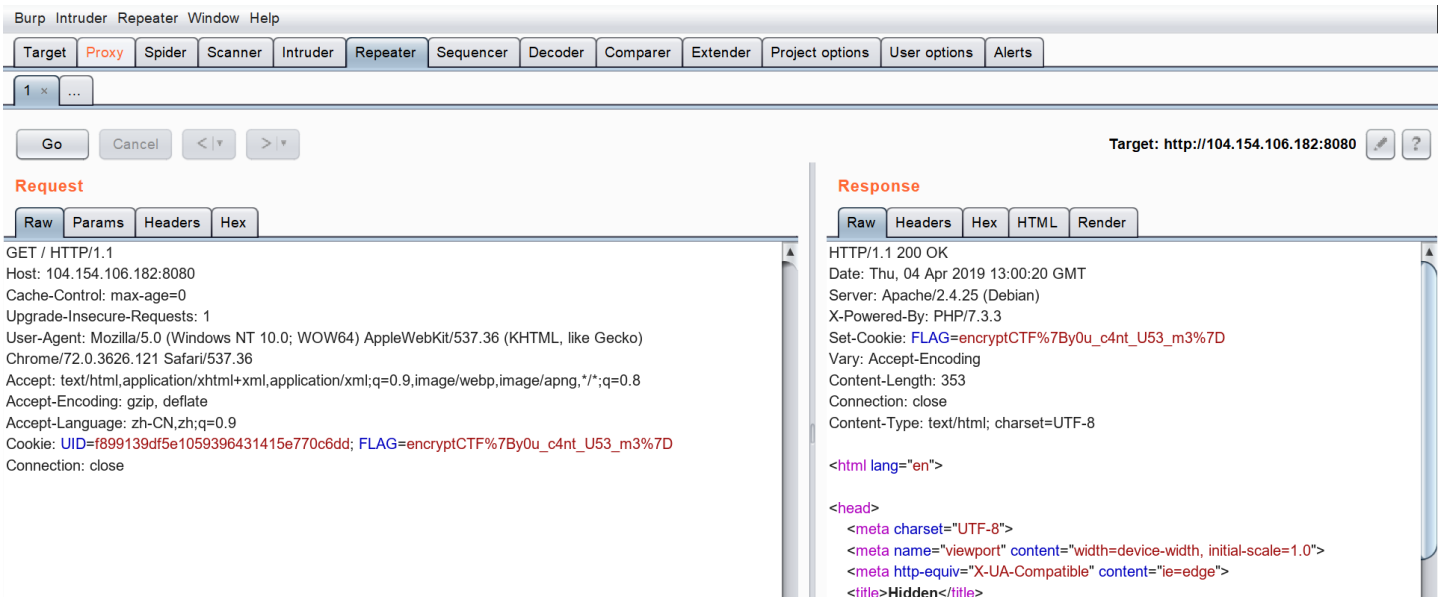
二、ENCRYPT CTF—Sweeeeeet（writeup）：

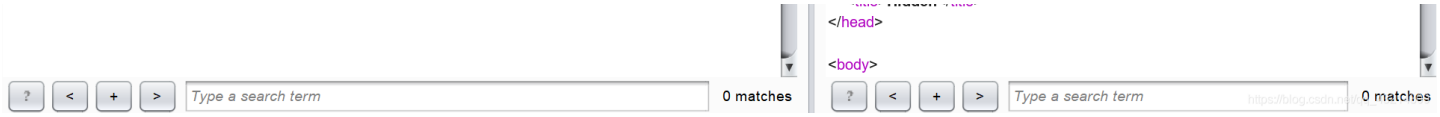
一开始打开网页是下面：



按照正常思路，先看以一下view-source，没看到；再试一下其他的后缀名，也没啥卵用，那就看一下Cookie：

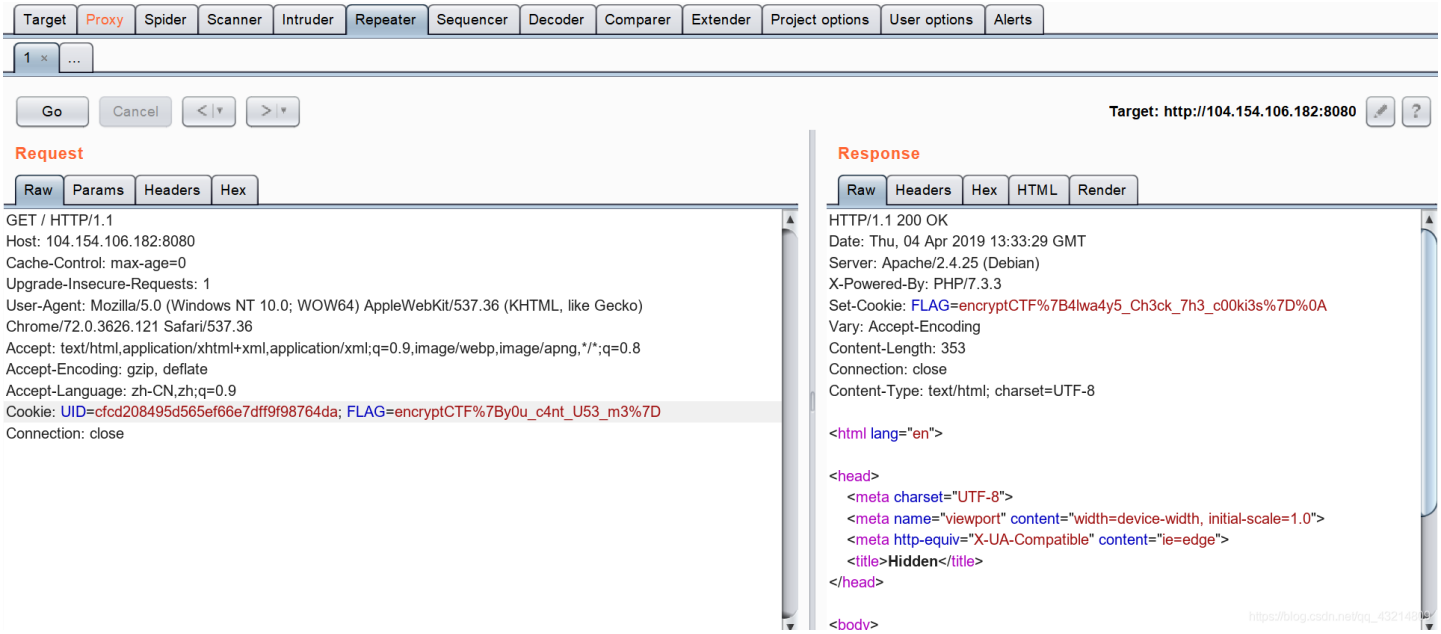
看见这个我还挺激动的，以为自己得到了flag，直接改一下提交答案了。结果显示Incorrect，我尼玛，还是太天真了。没办法，那就中规中矩的抓个包看一下：





既然这样的话，还剩下一个Cookie，看看能不能从里面得到点什么信息，试了一些编码，最后是MD5：

很明显，UID=100，那么接下来就应该跟数字有关了，尝试着更改不同的值再进行MD5转换，最后在md5(0)这个地方找到flag：

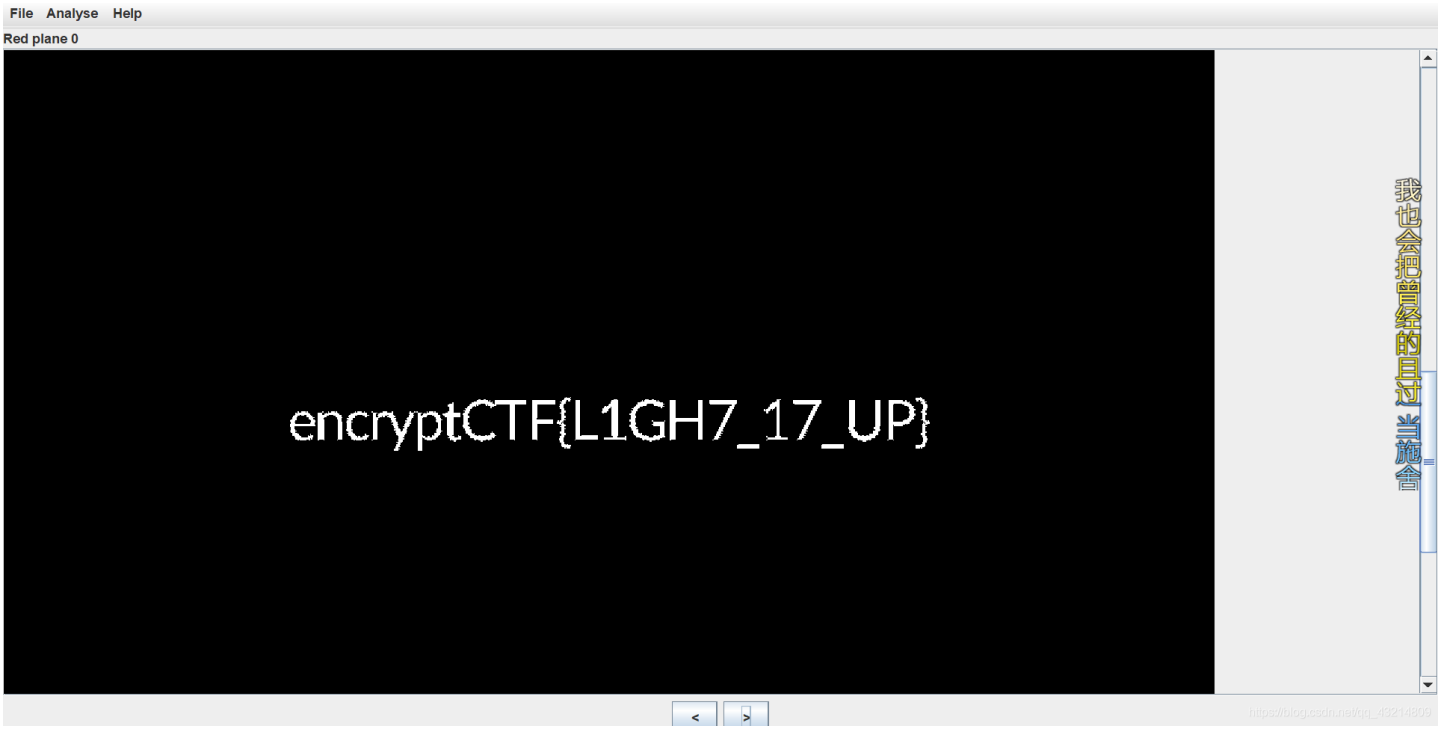


三、ENCRYPT CTF—Into the Black（writeup）：

这题也是比较水的，下载出来一张黑色的图片：

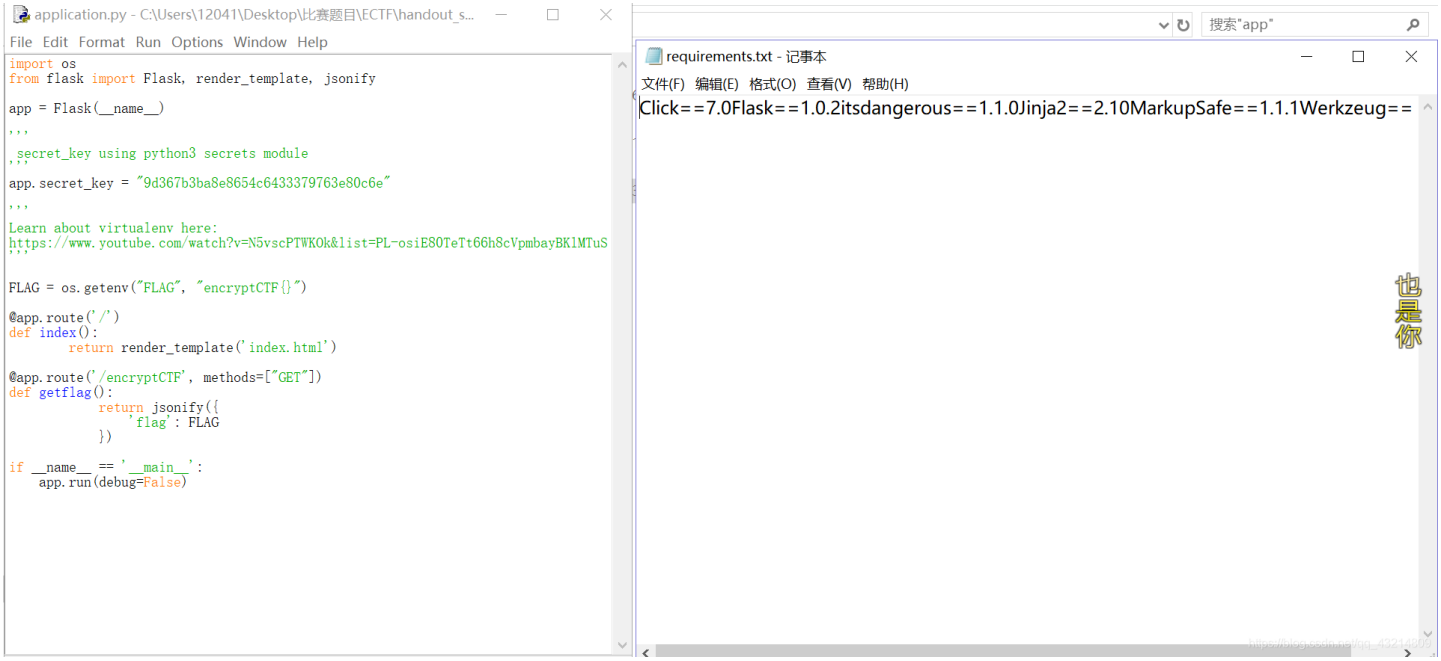
encryptCTF{L1GHT_17_UP}

直接放Stegsolve里看一下就行，在Red 0、Blue 0、Green 0、Random map 1、2、3都可以看到：



四、ENCRYPT CTF—Slash Slash（writeup）：

这道题怎么讲，误打误撞找到base64编码，直接解开得到flag (。￣ ￣。) 后来看了看别的大哥的writeup，这道题还是有点东西的。给了我们一个文件，打开后一个py文件，一个txt，三个文件夹，按照惯例先打开py和txt看一看。



在做出对我这种菜鸡没啥大的作用的判断后，果断打开别的文件夹，先看env/bin，顺手打开了第一个activate，一堆代码，往下翻，看到了一串base64编码：

```
45
46 _OLD_VIRTUAL_PATH="$PATH"
47 PATH="$VIRTUAL_ENV/bin:$PATH"
48 export PATH
49
```

```
# unset PYTHONHOME if set
if ! [ -z "${PYTHONHOME+_}" ] ; then
    _OLD_VIRTUAL_PYTHONHOME="$PYTHONHOME"
    unset PYTHONHOME
fi

if [ -z "${VIRTUAL_ENV_DISABLE_PROMPT-}" ] ; then
    _OLD_VIRTUAL_PS1="${PS1-}"
    if [ "x" != x ] ; then
        PS1="${PS1-}"
    else
        PS1="(`basename \"$VIRTUAL_ENV\"`) ${PS1-}"
    fi
    export PS1
fi

# Make sure to unalias pydoc if it's already there
alias pydoc 2>/dev/null >/dev/null && unalias pydoc || true

pydoc () {
    python -m pydoc "$@"
}

# This should detect bash and zsh, which have a hash command that must
# be called to get it to forget past commands.  Without forgetting
# past commands the $PATH changes we made may not be respected
if [ -n "${BASH-}" ] || [ -n "${ZSH_VERSION-}" ] ; then
    hash -r 2>/dev/null
fi

# export $(echo RkxBRwo= | base64 -d)="ZW5jcnlwdENURntjb21tZW50c18mX2luZGVudGF0aW9uc19tYWtlc19qb2hubnlfYV9nb29kX3Byb2dyYW1tZXJ9Cg=="
```

Normal text file | length : 2,302 | lines : 80 | Ln : 80 | Col : 132 | Sel : 92 | 1 | Unix (LF) | UTF-8 | INS

我开始决定回去

直接解个码，嘤嘤嘤~ㄟ(˙ ω˙ㄟ)：

Base编码

base16、base32、base64

```
ZW5jcnlwdENURntjb21tZW50c18mX2luZGVudGF0aW9uc19tYWtlc19qb2hubnlfYV9nb29kX3Byb2dyYW1tZXJ9Cg==
```

编码 base64　　字符集 utf8(unicode编码)

编 码　　解 码

```
encryptCTF{comments_&_indentations_makes_johnny_a_good_programmer}
```

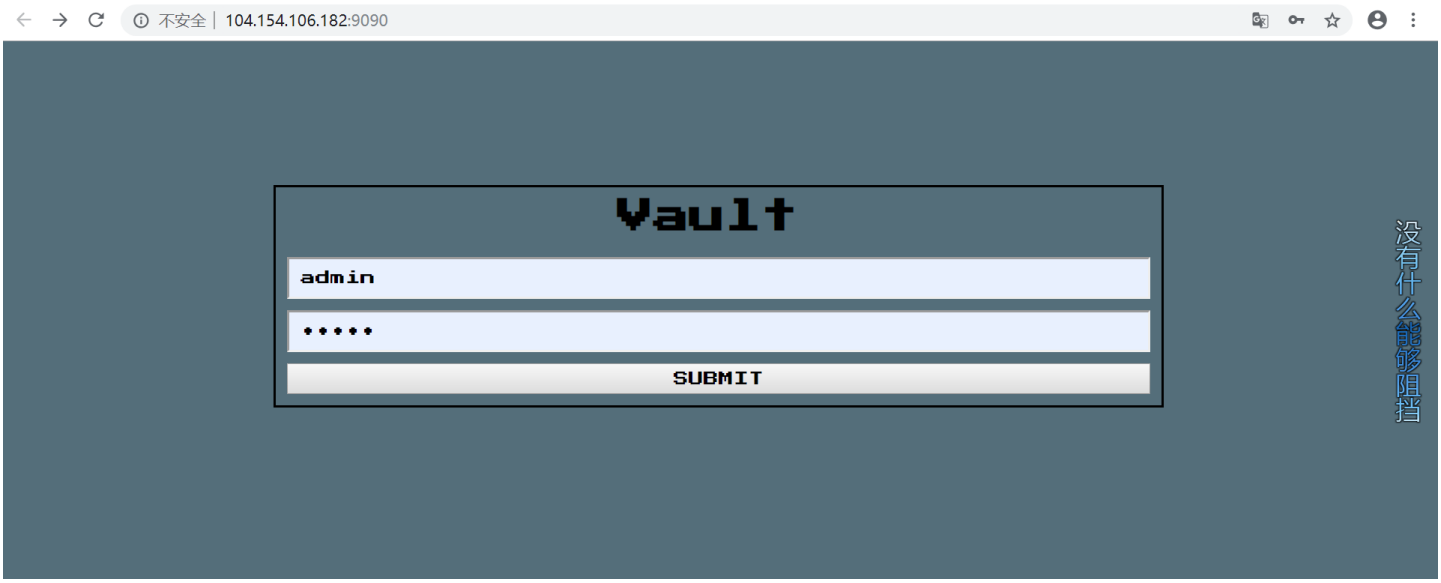

差你还愿意

当然好像还有别的方法，应该是虚拟环境的，我看不太懂，wp放下面了：

https://www.abs0lut3pwn4g3.cf/writeups/2019/04/04/encryptctf-slashslash.html

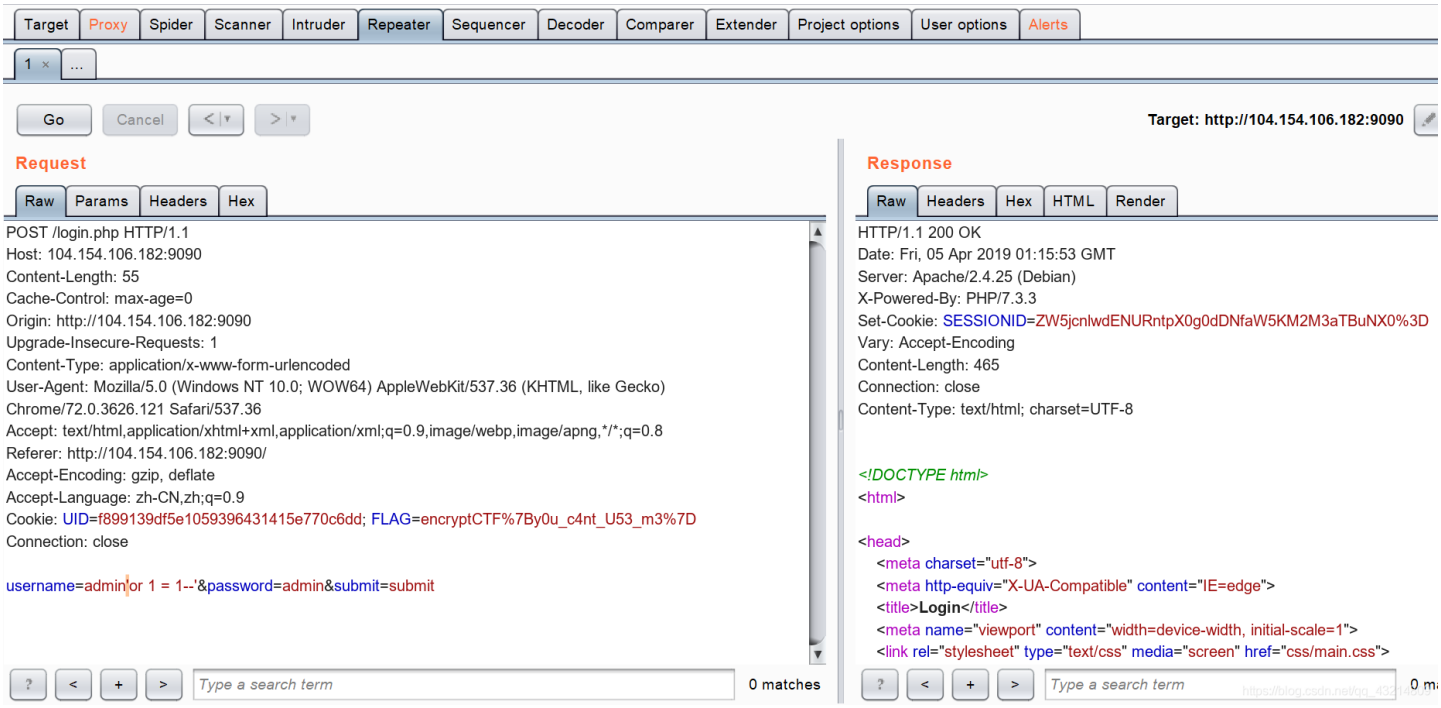
五、ENCRYPT CTF—Vault（writeup）：

登陆题，按照之前做的一些套路，应该会有注入：

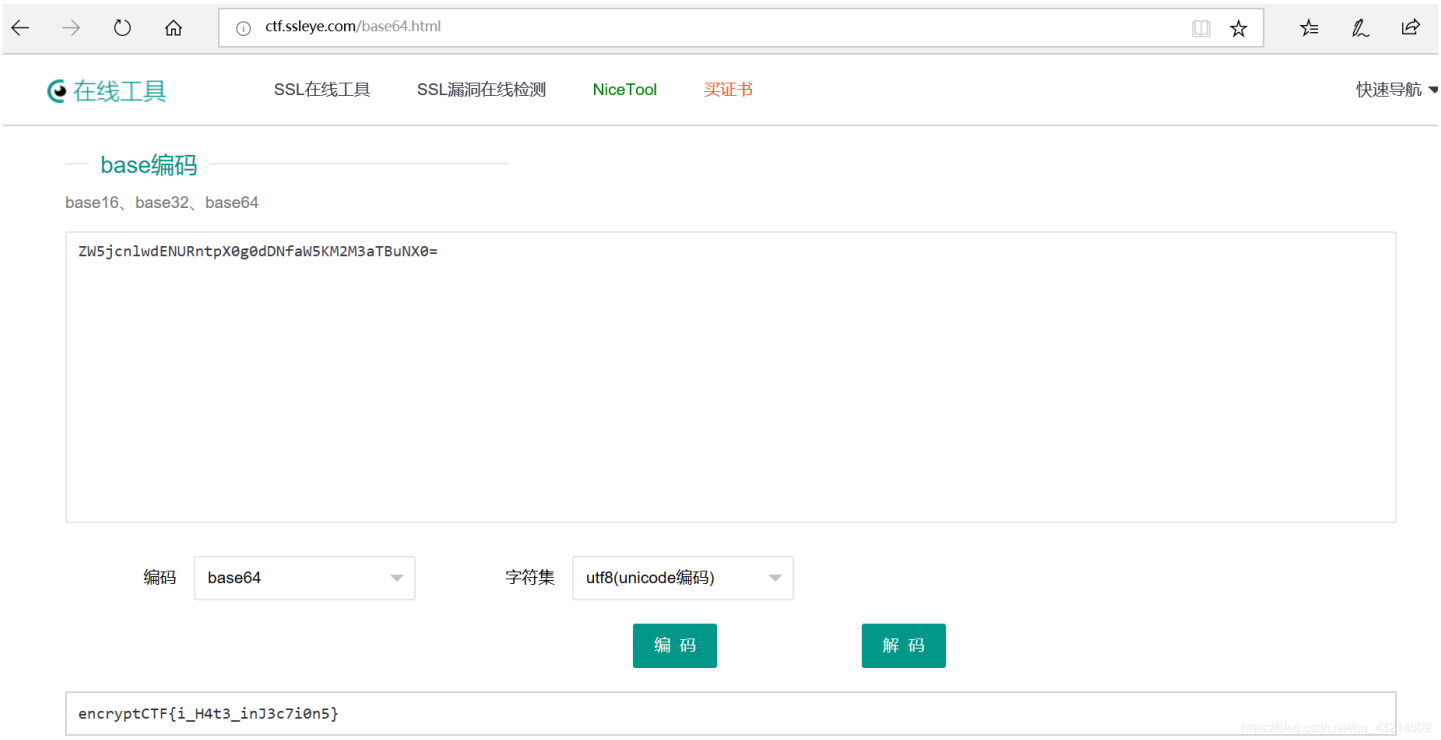


没有什么能够阻挡

直接抓包，修改username的值为：admin'or 1 = 1–'：

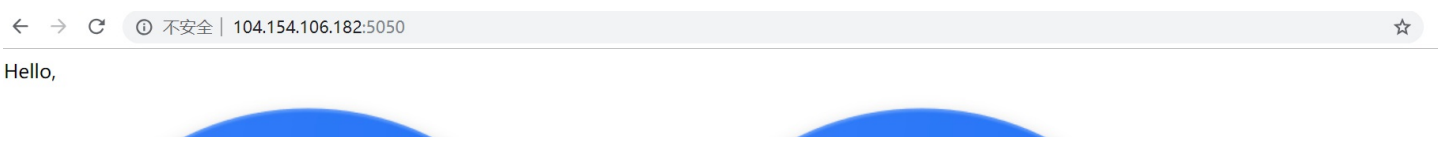


注意到右侧Set-Cookie，先放到URL里解码，再放到base64，直接得到flag：



六、ENCRYPT CTF—repeaaaaaat（重解）：

打开网页如下：



始终重复图片，没办法，先看一下view-source：

不安全 | view-source:104.154.106.182:5050

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>repeaaaaaat</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <script>
        function repeat() {
            for(var i=0; i<10; i++) {
            lol = document.createElement("img")
            lol.src = "/static/lol.png"
            var shit = document.getElementById('shit')
            shit.appendChild(lol)
            }
        }
    </script>
    </head>
    <body onscroll=repeat()>
        Hello,<div id="shit">
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
            <img src='/static/lol.png'>
        </div>
        <!-- L2xvbF9ub19vbmVfd2lsbF9zZWVfd2hhdHNfaGVyZQ== -->
    </body>
</html>
```

底部一个base64编码，解码看一下：

ctf.ssleye.com/base64.html

在线工具　SSL在线工具　SSL漏洞在线检测　NiceTool　买证书　快速

base编码

base16、base32、base64

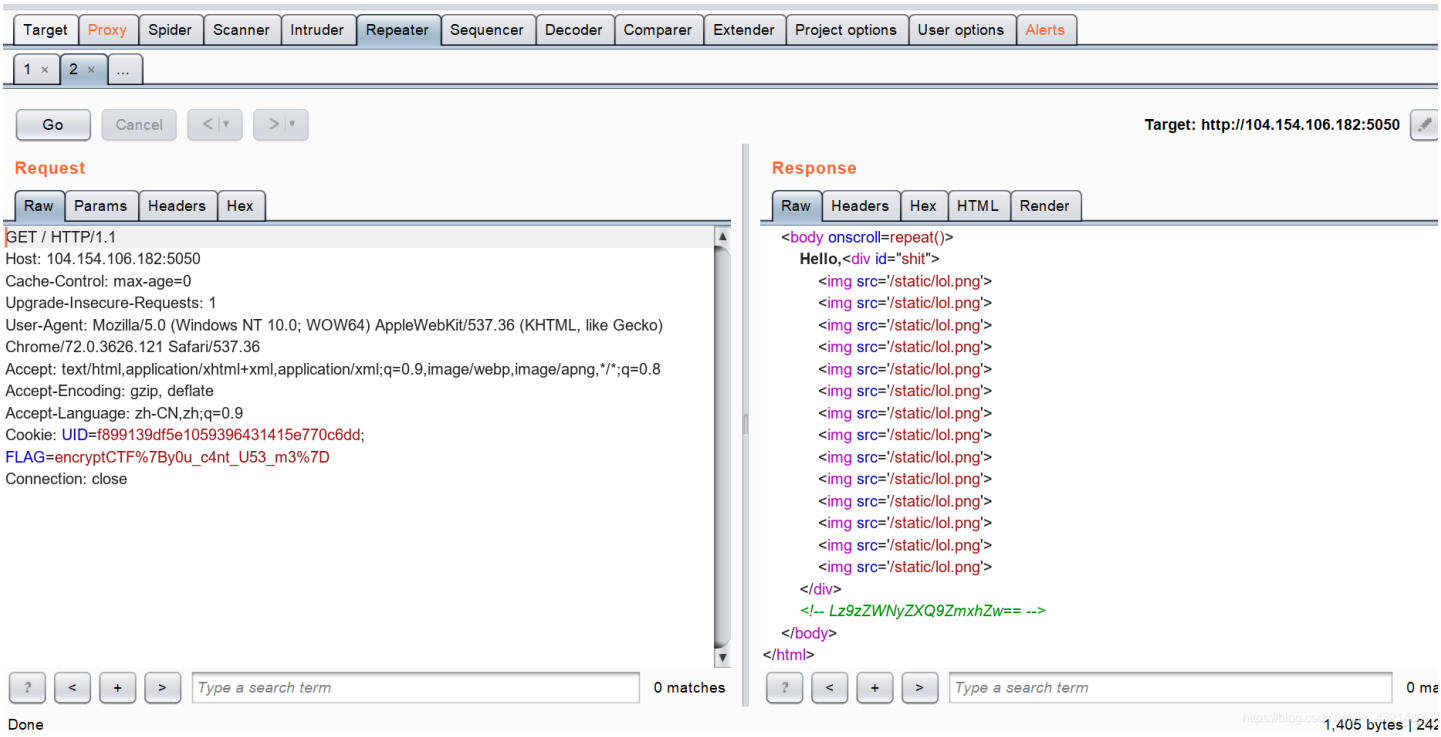
```
L2xvbF9ub19vbmVfd2lsbF9zZWVfd2hhdHNfaGVyZQ==
```

编码 base64 字符集 utf8(unicode编码)

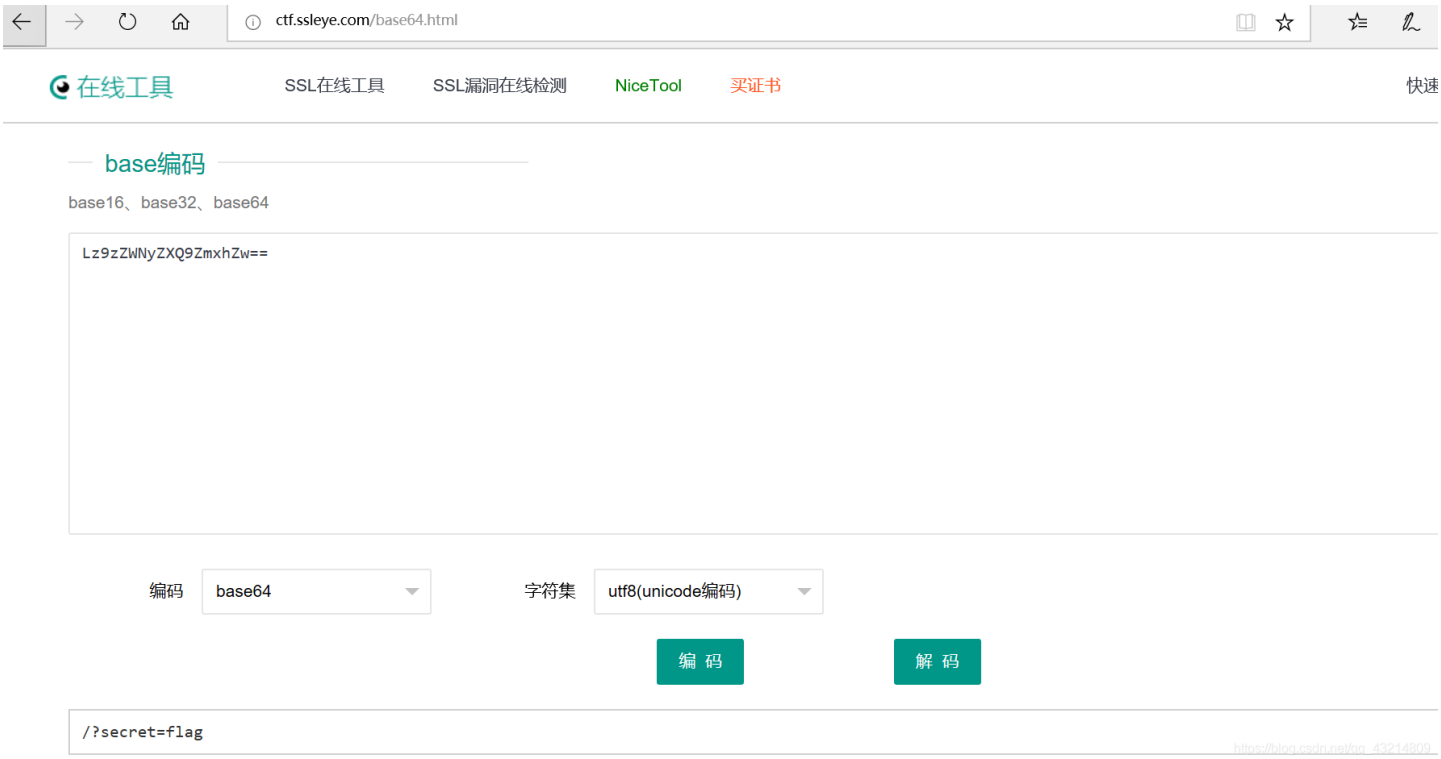
编 码 解 码

```
/lol_no_one_will_see_whats_here
```

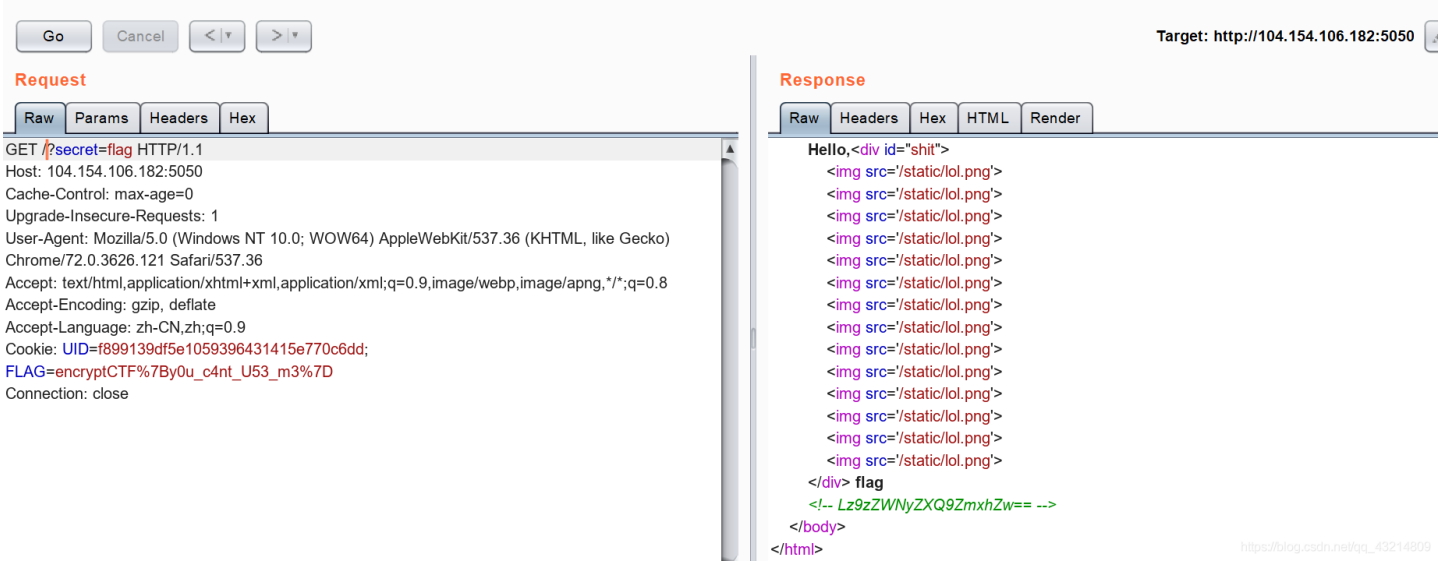
唔，那就只能老样子抓个包看看：



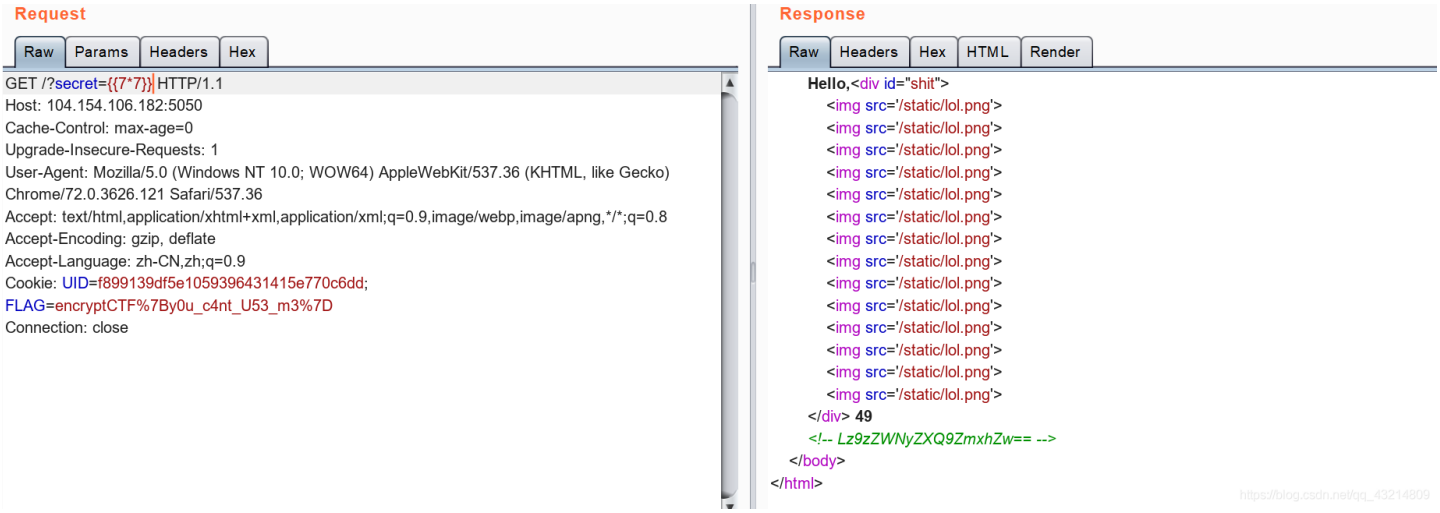
又出现了一段base64的编码，解码：



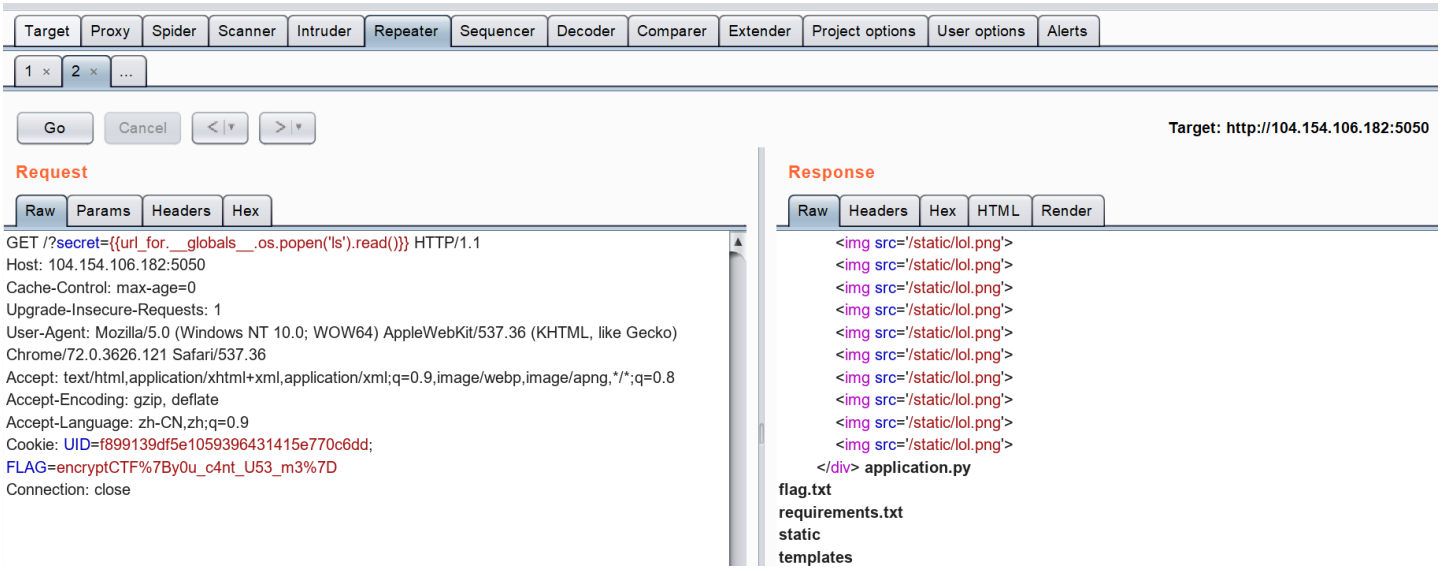
带进去：



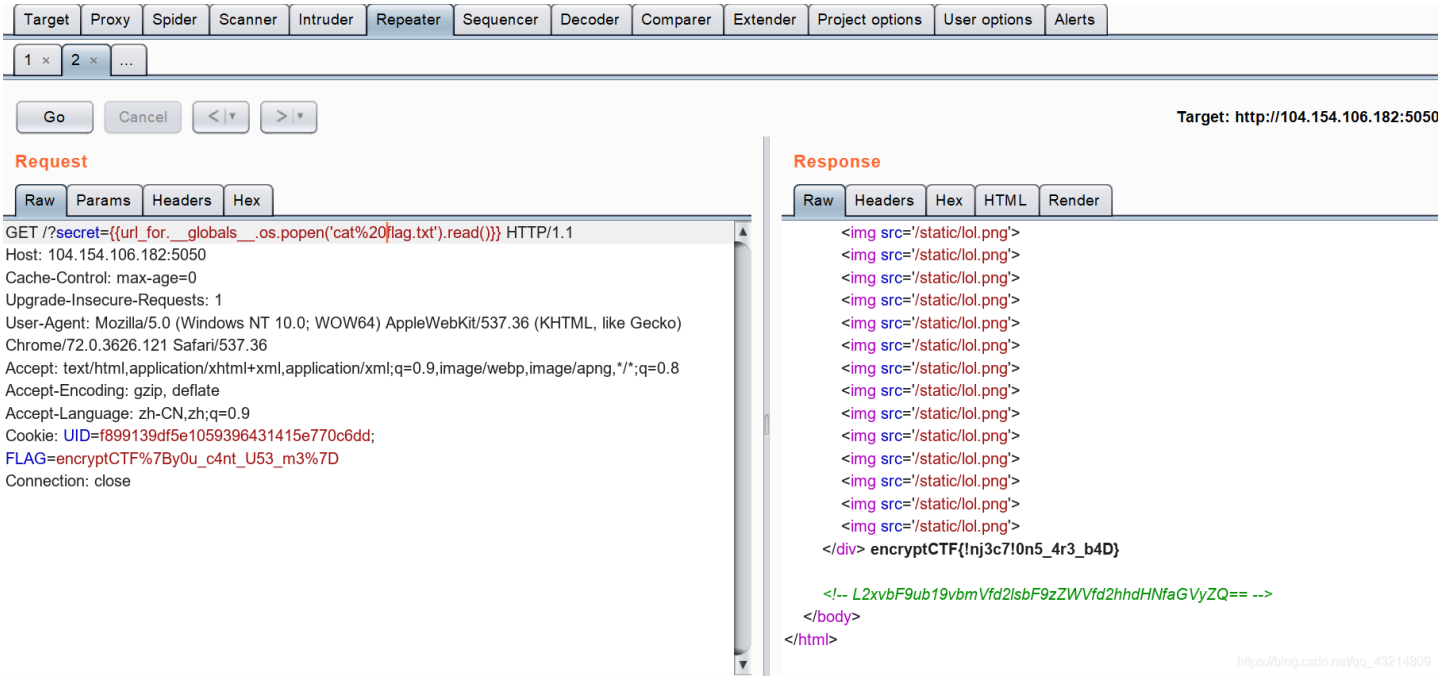
我们可以发现右侧div出现flag，通过改变左侧secret的值，可以发现右侧的div值也在改变，然后我就不知道下一步是干啥了。
后来看了大佬的writeup，说后面就是SSTI了，我尼玛...真想给自己一嘴巴子。之前在攻防世界上见到过一个这种题，沃日 ε(┬┬﹏┬┬)3
通过测试{{7*7}},发现返回49。行了，和攻防世界那个题一个样Jinjia2：



那么就是构造payload的问题了：之前那道题的payload是不能用在这里的，这道题的payload我也没搞清楚：



打开flag.txt就好：



当然也可以直接用tplmap，可能还更快一点。

大佬的writeup：

https://rawsec.ml/en/ENCRYPT-CTF-2019-write-ups/

小结：

1.这次比赛，Web题目还是比较简单的，各题目的解答人数都是蛮多的，在别的比赛里都是很少见的。对于我个人而言，解题有运气的成分，但是还好，可能也就是最后一题的payload之前没见到过，自己了解的知识内容也还没到那个程度，所以后面的话，还是要看一看相关文献资料。

2.日常感谢大佬们在ctftime上的writeup，能给我提供一些新的其他方面的思路，感谢。谢啦!!☆ヘ(*＾-゜)v