2019年CTF3月比赛记录（二）：UTCTF AeroCTF 部分题目writeup

原创

極品━═☆宏 于 2019-03-12 16:02:37 发布 1281 收藏

分类专栏： 2019年CTF比赛—3月赛 CTF_web

本文链接：https://blog.csdn.net/qq_43214809/article/details/88383320

版权

2019年CTF比赛—3月赛 同时被 2 个专栏收录

3 篇文章 1 订阅

订阅专栏

CTF_web

13 篇文章 0 订阅

订阅专栏

这次的比赛基本上是连着的几个，先是AeroCTF，然后是PragyanCTF，然后是UTCTF，Pragyan的比赛之前注册没注册成功，就放弃了，只参加了剩下的两个，先说AeroCTF，不会，一道Web题目都不会 ╭(╯^╰)╮，知识有限未能作出解答，只能看别人的writeup了；至于UTCTF，Web题目起码还是做出来一个简单的题目，然后做了一个基础密码，一个基础杂项，其他的就呵呵了。但是说实话，当我看到解题人数不多时我也就放心了(￣▽￣")

比赛时间：2019年3月9日至2019年3月10日

重解时间：2019年3月11日至2019年3月12日

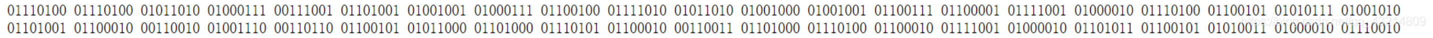
（比完赛后，UTCTF和AeroCTF的比赛网址关闭了，重解的话可能有些困难，我大部分是凭着记忆来的，没有图片，我从自己的思路出发去理解这个题到底该怎么做，事实上我觉得凭着记忆直接看大佬的writeup来得更快一些）

一、writeup：

1.[basics]crypto：（我是学web方向的，被迫做了道密码题，可能解题的方式比较蠢，还望密码学的大佬谅解）

首先是个二进制编码txt：

```
01010101 01101000 00101101 01101111 01101000 00101100 00100000 01101100 01101111 01101111 01101011 01110011 00100000 01101100 01101001 01101011 01100101 00100000 01110111 01100101 00100000
01101000 01100001 01110110 01100101 00100000 01100001 01101110 01101111 01110100 01101000 01100101 01110010 00100000 01100010 01101100 01101111 01100011 01101011 00100000 01101111 01100110
00100000 01110100 01100101 01111000 01110100 00101100 00100000 01110111 01101001 01110100 01101000 00100000 01110011 01101111 01101101 01100101 00100000 01110011 01101111 01110010 01110100
00100000 01101111 01100110 00100000 01110011 01110000 01100101 01100011 01101001 01100001 01101100 00100000 01100101 01101110 01100011 01101111 01100100 01101001 01101110 01100111 00101110
00100000 01000011 01100001 01101110 00100000 01111001 01101111 01110101 00100000 01100110 01101001 01100111 01110101 01110010 01100101 00100000 01101111 01110101 01110100 00100000 01110111
01101000 01100001 01110100 00100000 01110100 01101000 01101001 01110011 00100000 01100101 01101110 01100011 01101111 01100100 01101001 01101110 01100111 00100000 01101001 01110011 00111111
00100000 00101000 01101000 01101001 01101110 01110100 00111010 00100000 01101001 01100110 00100000 01111001 01101111 01110101 00100000 01101100 01101111 01101111 01101011 00100000 01100011
01100001 01110010 01100101 01100110 01110101 01101100 01101100 01111001 00101100 00100000 01111001 01101111 01110101 00100111 01101100 01101100 00100000 01101110 01101111 01110100 01101001
01100011 01100101 00100000 01110100 01101000 01100001 01110100 00100000 01110100 01101000 01100101 01110010 01100101 00100000 01101111 01101110 01101100 01111001 00100000 01100011 01101000
01100001 01110010 01100001 01100011 01110100 01100101 01110010 01110011 00100000 01110000 01110010 01100101 01110011 01100101 01101110 01110100 00100000 01100001 01110010 01100101 00100000
01000001 00101101 01011010 00101100 00100000 01100001 00101101 01111010 00101100 00100000 00110000 00101101 00111001 00101100 00100000 01100001 01101110 01100100 00100000 01110011 01101111
01101101 01100101 01110100 01101001 01101101 01100101 01110011 00100000 00101111 00100000 01100001 01101110 01100100 00100000 00101011 00101110 00100000 01010011 01100101 01100101 00100000
01101001 01100110 00100000 01111001 01101111 01110101 00100000 01100011 01100001 01101110 00100000 01100110 01101001 01101110 01100100 00100000 01100001 01101110 00100000 01100101 01101110
01100011 01101111 01100100 01101001 01101110 01100111 00100000 01110100 01101000 01100001 01110100 00100000 01101100 01101111 01101111 01101011 01110011 00100000 01101100 01101001 01101011
01100101 00100000 01110100 01101000 01101001 01110011 00100000 01101111 01101110 01100101 00101110 00101001 00001010 01010100 01101101 01010110 00110011 01001001 01000111 01001110 01101111
01011001 01010111 01111000 01110011 01011010 01010111 00110101 01101110 01011010 01010011 01000101 01100111 01010001 00110010 01000110 01110101 01001001 01001000 01101100 01110110 01100100
01010011 01000010 01101101 01100001 01010111 01100100 00110001 01100011 01101101 01010101 01100111 01100010 00110011 01010110 00110000 01001001 01001000 01100100 01101111 01011001 01011000
01010001 01101110 01100011 01111001 01000010 01101110 01100010 00110010 01101100 01110101 01011010 01111001 01000010 01110110 01100010 01101001 01000010 01101111 01011010 01011000 01001010
01101100 01010000 01111001 01000010 01001010 01100100 01000011 01000010 01110011 01100010 00110010 00111001 01110010 01100011 01111001 01000010 01110011 01100001 01010111 01110100 01101100
01001001 01001000 01010010 01101111 01011010 01010011 01000010 01110011 01011010 01011000 01010010 00110000 01011010 01011000 01001010 01111010 01001001 01000111 01000110 01111001 01011010
01010011 01000010 01111010 01100001 01000111 01101100 01101101 01100100 01000111 01010110 01101011 01001001 01000111 01001010 00110101 01001001 01001000 01001110 01110110 01100010 01010111
01010101 01100111 01011001 00110010 00111001 01110101 01100011 00110011 01010010 01101000 01100010 01101110 01010001 01110101 01001001 01000011 01101000 01101111 01100001 01010111 00110101
00110000 01001111 01101001 01000010 00110101 01100010 00110011 01010101 01100111 01100010 01010111 01101100 01101110 01100001 01001000 01010001 01100111 01100100 00110010 01000110 01110101
01100100 01000011 01000010 00110000 01100010 01111001 01000010 01111010 01100100 01000111 01000110 01111001 01100100 01000011 01000010 01110011 01100010 00110010 00111001 01110010 01100001
01010111 00110101 01101110 01001001 01001000 01010110 01110111 01001001 01000110 01001010 01110110 01100010 01010111 01000110 01110101 01001001 01001000 01000010 01101100 01100010 00110011
01000010 01110011 01011010 01010011 01101011 01110101 01000011 01101101 01110100 00110010 01011001 01101110 01001110 01111000 01100011 01101101 01010001 01110011 01001001 01000111 01101100
00110101 01011010 01010011 01100100 01101001 01100010 01111001 01000010 01110010 01100100 01101110 01100100 00110101 01011001 00110010 01010001 01100111 01011010 01001000 01001010 01110110
01011001 01101101 00111000 01101000 01001001 01000110 01101000 00110101 01011010 01111001 01000010 01110111 01100101 01010111 01001001 01100111 01011010 01001000 01001010 01110110 01001001
01001000 01000010 01111010 01100101 01000111 01110100 00110010 01001001 01000011 01101000 01110010 01100101 01000111 00110100 01100111 01100100 00110010 01110100 01110000 01100010 01000111
00111000 01100111 01011010 01001000 01001010 01110110 01001001 01001000 01001010 01110010 01011001 01101101 00110101 01110110 01011001 00110010 01010001 01110101 01001100 01101001 00110100
01110000 01001001 01001000 01110000 01110010 01011001 01101101 01010001 00110110 01001001 01000111 01110011 01100111 01011001 00110010 01010110 01110011 01011001 00110010 01010010 01111010
01011010 01000111 01010110 01101011 01100011 00110011 01101100 00110100 01001001 01000111 00110001 01111010 01100101 01101110 01001010 01110110 01011001 01101001 00110100 01100111 01010101
00110011 01100111 01100111 01011010 01001000 01001010 01110110 01001001 01001000 01000010 00110101 01100100 01101110 01011010 00110101 01011010 00110011 01001110 00110100 01100011 01010011
01000010 01101011 01100010 00110010 01101000 01101011 01001100 01000011 01000010 01010100 01001010 00110010 01011010 01110110 01001001 01000111 01010010 01110010 01100100 01010111 00111001
00110100 01001001 01001000 01100100 01110000 01001001 01001000 01100100 01110110 01011001 00110010 01001110 01110010 01100011 01010111 00111000 01100111 01100001 00110011 01101000 01110101
01001001 01000111 01001010 01110110 01100101 01101110 01011010 01110010 01100010 01010111 00111001 01110101 01001001 01000111 00111001 01101101 01100010 00110010 01001010 01110000 01001001
01000111 01110100 00110010 01100101 01101110 01001010 01110010 01100010 01000111 00111001 01101011 01100011 00110010 00110000 01100111 01100010 01011000 01001010 01110010 01011001 01101101
```

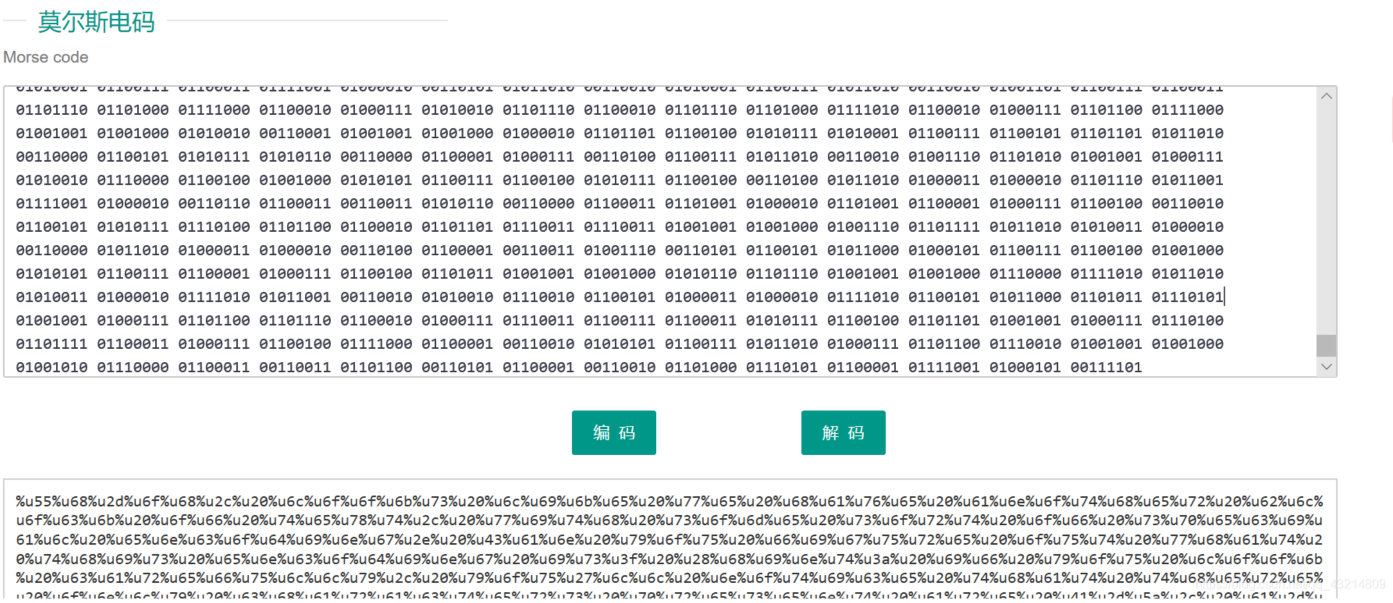


```
01110100 01110100 01011010 01000111 00111001 01101001 01001001 01000111 01100100 01111010 01011010 01001000 01001001 01100111 01100001 01111001 01000010 01110100 01100101 01010111 01001010
01101001 01100010 00110010 01001110 00110110 01100101 01011000 01101000 01110101 01100010 00110011 01101000 01110100 01100010 01111001 01000010 01101011 01100101 01010011 01000010 01110010
```

编码很长，自行体会...

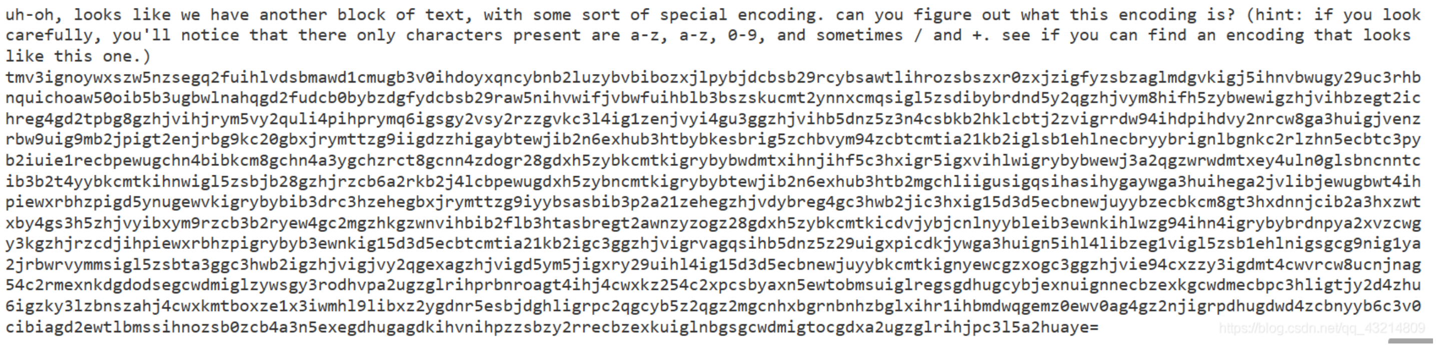
因为我不是密码这个方向的，所以没有一个大概的方向，或是说一个解题的框架。我首先的想法是通过进制之间的转换或是尝试其他的编码来解，毕竟有神器：http://ctf.ssleye.com/。

后来误打误撞的在摩尔斯电码里碰出来点东西，例如当我输入01010101时，解码会得到%u55,很明显55是转换后的16进制，后来又通过自己查询资料，觉得跟Escape有点关系，于是全部在%u后面加上两个00，进行解码，得到的是以下内容：



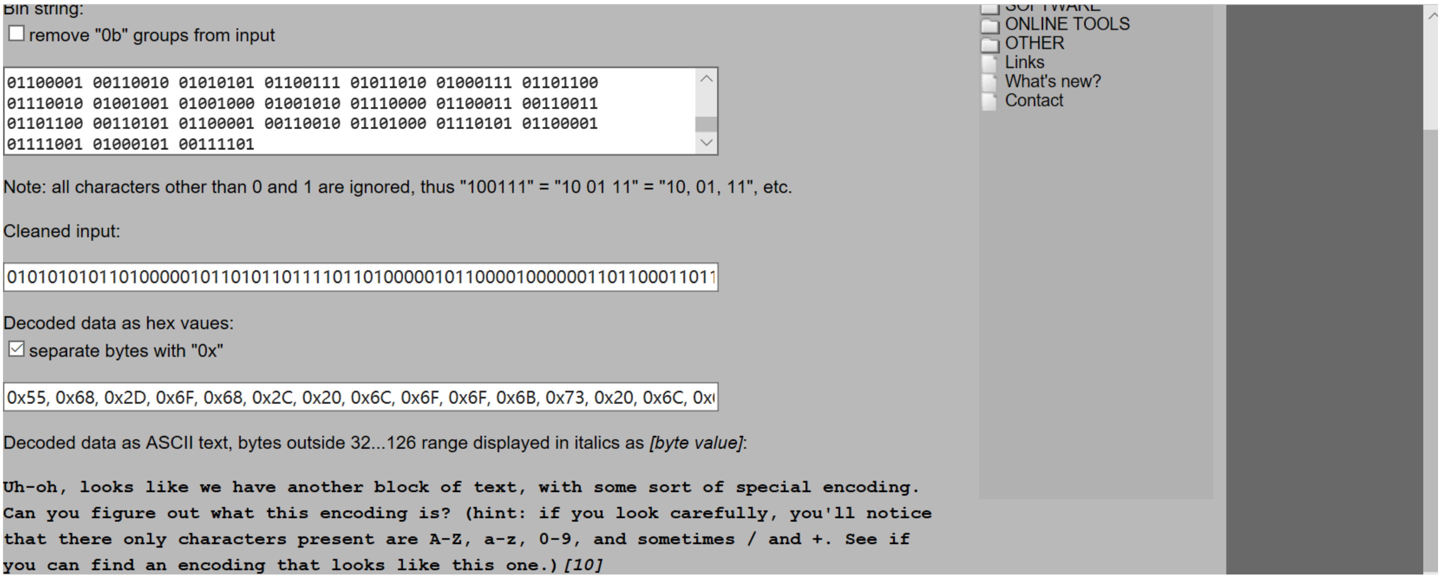
```
01101110 01101000 01111000 01100010 01000111 01010010 01101110 01100010 01101110 01101000 01111010 01100010 01000111 01101100 01111000
01001001 01001000 01010010 00110001 01001001 01001000 01000010 01101101 01100100 01010111 01010001 01100111 01100101 01101101 01011010
00110000 01100101 01010111 01010110 00110000 01100001 01000111 00110100 01100111 01011010 00110010 01001110 01101010 01001001 01000111
01010010 01110000 01100100 01001000 01010101 01100111 01100100 01010111 01100100 00110100 01011010 01000011 01000010 01101110 01011001
01111001 01000010 00110110 01100011 00110011 01010110 00110000 01100011 01101001 01000010 01101001 01100001 01000111 01100100 00110010
01100101 01010111 01110100 01101100 01100010 01101101 01110011 01110011 01001001 01001000 01001110 01101111 01011010 01010011 01000010
00110000 01011010 01000011 01000010 00110100 01100001 00110011 01001110 00110101 01100101 01011000 01000101 01100111 01100100 01001000
01010101 01100111 01100001 01000111 01100100 01101011 01001001 01001000 01010110 01101110 01001001 01001000 01110000 01111010 01011010
01010011 01000010 01111010 01011001 00110010 01010010 01110010 01100101 01000011 01000010 01111010 01100101 01011000 01101011 01110101
01001001 01000111 01101100 01101110 01100010 01000111 01110011 01100111 01100011 01010111 01100100 01101101 01001001 01000111 01110100
01101111 01100011 01000111 01100100 01111000 01100001 00110010 01010101 01100111 01011010 01000111 01101100 01110010 01001001 01001000
01001010 01110000 01100011 00110011 01101100 00110101 01100001 00110010 01101000 01110101 01100001 01111001 01000101 00111101
```

```
%u55%u68%u2d%u6f%u68%u2c%u20%u6c%u6f%u6f%u6b%u73%u20%u6c%u69%u6b%u65%u20%u77%u65%u20%u68%u61%u76%u65%u20%u61%u6e%u6f%u74%u68%u65%u72%u20%u62%u6c%
u6f%u63%u6b%u20%u6f%u66%u20%u74%u65%u78%u74%u2c%u20%u77%u69%u74%u68%u20%u73%u6f%u6d%u65%u20%u73%u6f%u72%u74%u20%u6f%u66%u20%u73%u70%u65%u63%u69%u
61%u6c%u20%u65%u6e%u63%u6f%u64%u69%u6e%u67%u2e%u20%u43%u61%u6e%u20%u79%u6f%u75%u20%u66%u69%u67%u75%u72%u65%u20%u6f%u75%u74%u20%u77%u68%u61%u74%u2
0%u74%u68%u69%u73%u20%u65%u6e%u63%u6f%u64%u69%u6e%u67%u20%u69%u73%u3f%u20%u28%u68%u69%u6e%u74%u3a%u20%u69%u66%u20%u79%u6f%u75%u20%u6c%u6f%u6f%u6b
%u20%u63%u61%u72%u65%u66%u75%u6c%u6c%u79%u2c%u20%u79%u6f%u75%u27%u6c%u6c%u20%u6e%u6f%u74%u69%u63%u65%u20%u74%u68%u61%u74%u20%u74%u68%u65%u72%u65%
```



```
uh-oh, looks like we have another block of text, with some sort of special encoding. can you figure out what this encoding is? (hint: if you look
carefully, you'll notice that there only characters present are a-z, a-z, 0-9, and sometimes / and +. see if you can find an encoding that looks
like this one.)
tmv3ignoywxszw5nzsegq2fuihlvdsbmawd1cmugb3v0ihdoyxqncybnb2luzybvbibozxjlpybjdcbsb29rcybsawtlihrozsbszxr0zxjzigfyzsbzaglmdgvkigj5ihnvbwugy29uc3rhb
nquichoaw50oib5b3ugbwlnahqgd2fudcb0ybzdgfydcbsb29raw5nihvwifjvbwfuihblb3bszskucmt2ynnxcmqsigl5zsdibybrdnd5y2qgzhjvym8hifh5zybwewigzhjvihbzegt2ic
hreg4gd2tpbg8gzhjvihjrym5vy2quli4pihprymq6igsgy2vsy2rzzgvkc3l4ig1zenjvyi4gu3ggzhjvihb5dnz5z3n4csbkb2hklcbtj2zvigrrdw94ihdpihdvy2nrcw8ga3huigjvenz
rbw9uig9mb2jpigt2enjrbg9kc20gbxjrymttzg9iigdzzhigaybtewjib2n6exhub3htbybkesbrig5zchbvym94zcbtcmtia21kb2iglsb1ehlnecbryybrignlbgnkc2rlzhn5ecbtc3py
b2iuie1recbpewugchn4bibkcm8gchn4a3ygchzrct8gcnn4zdogr28gdxh5zybkcmtkigrybybwdmtxihnjihf5c3hxigr5igxvihlwigrybybwewj3a2qgzwrwdmtxey4uln0glsbncnntc
ib3b2t4yybkcmtkihnwigl5zsbjb28gzhjrzcb6a2rkb2j4lcbpewugdxh5zybncmtkigrybybtewjib2n6exhub3htb2mgchliigusigqsihasihygaywga3huihega2jvlibjewugbwt4ih
piewxrbhzpigd5ynugewvkigrybybib3drc3hzehegbxjrymttzg9iyybsasbib3p2a21zehegzhjvdybreg4gc3hwb2jic3hxig15d3d5ecbnewjuyybzecbkcm8gt3hxdnnjcib2a3hxzwt
xby4gs3h5zhjvyibxym9rzcb3b2ryew4gc2mgzhkgzwnvihbib2flb3htasbregt2awnzyzogz28gdxh5zybkcmtkicdvjybjcnlnyybleib3ewnkihlwzg94ihn4igrybybrdnpya2xvzcwg
y3kgzhjrzcdjihpiewxrbhzpigrybyb3ewnkig15d3d5ecbtcmtia21kb2igc3ggzhjvigrvagqsihb5dnz5z29uigxpicdkjywga3huign5ihl4libzeg1vigl5zsb1ehlnigsgcg9nig1ya
2jrbwrvymmsigl5zsbta3ggc3hwb2igzhjvigjvy2qgexagzhjvigd5ym5jigxry29uihl4ig15d3d5ecbnewjuyybkcmtkignyewcgzxogc3ggzhjvie94cxzzy3igdmt4cwvrcw8ucnjnag
54c2rmexnkdgdodsegcwdmiglzywsgy3rodhvpa2ugzglrihprbnroagt4ihj4cwxkz254c2xpcsbyaxn5ewtobmsuiglregsgdhugcybjexnuignnecbzexkgcwdmecbpc3hligtjy2d4zhu
6igzky3lzbnszahj4cwxkmtboxze1x3iwmhl9libxz2ygdnr5esbjdghligrpc2qgcyb5z2qgz2mgcnhxbgrnbnhzbglxihr1ihbmdwqgemz0ewv0ag4gz2njigrpdhugdwd4zcbnyyb6c3v0
cibiagd2ewtlbmssihnozsb0zcb4a3n5exegdhugagdkihvnihpzzsbzy2rrecbzexkuiglnbgsgcwdmigtocgdxa2ugzglrihjpc3l5a2huaye=
```

很明显地从提示中发现下一个编码应该是base64，但是给我的段落内容中不含有大写字母，这就说明我转换的有问题，后来又找了好多资料，好多解码工具来找问题，最后终于找到了，找到了正确的段落：



☐ remove "0b" groups from input

```
01100001 00110010 01010101 01100111 01011010 01000111 01101100
01110010 01001001 01001000 01001010 01110000 01100011 00110011
01101100 00110101 01100001 00110010 01101000 01110101 01100001
01111001 01000101 00111101
```

Note: all characters other than 0 and 1 are ignored, thus "100111" = "10 01 11" = "10, 01, 11", etc.

Cleaned input:

```
010101010110100000101101011011110110100000101100001000000110110001101
```

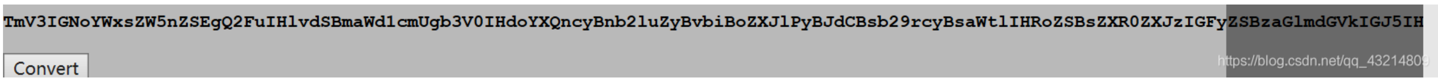
Decoded data as hex vaues:

☑ separate bytes with "0x"

```
0x55, 0x68, 0x2D, 0x6F, 0x68, 0x2C, 0x20, 0x6C, 0x6F, 0x6F, 0x6B, 0x73, 0x20, 0x6C, 0x
```

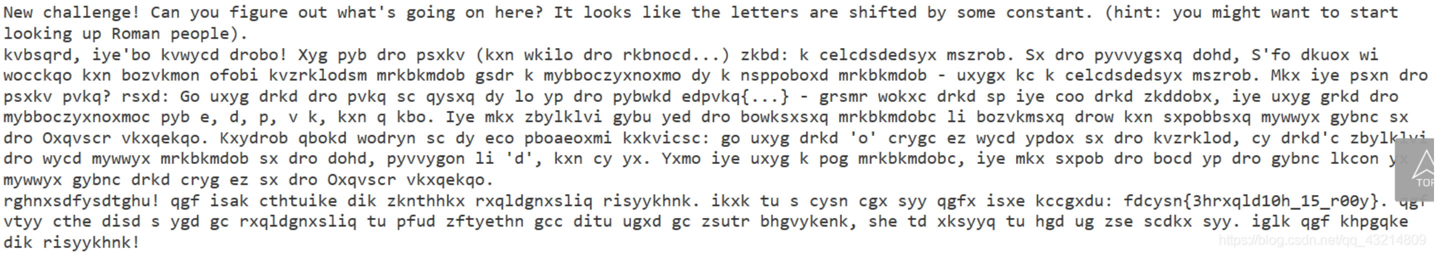
Decoded data as ASCII text, bytes outside 32...126 range displayed in italics as *[byte value]*:

```
Uh-oh, looks like we have another block of text, with some sort of special encoding.
Can you figure out what this encoding is? (hint: if you look carefully, you'll notice
that there only characters present are A-Z, a-z, 0-9, and sometimes / and +. See if
you can find an encoding that looks like this one.)[10]
```



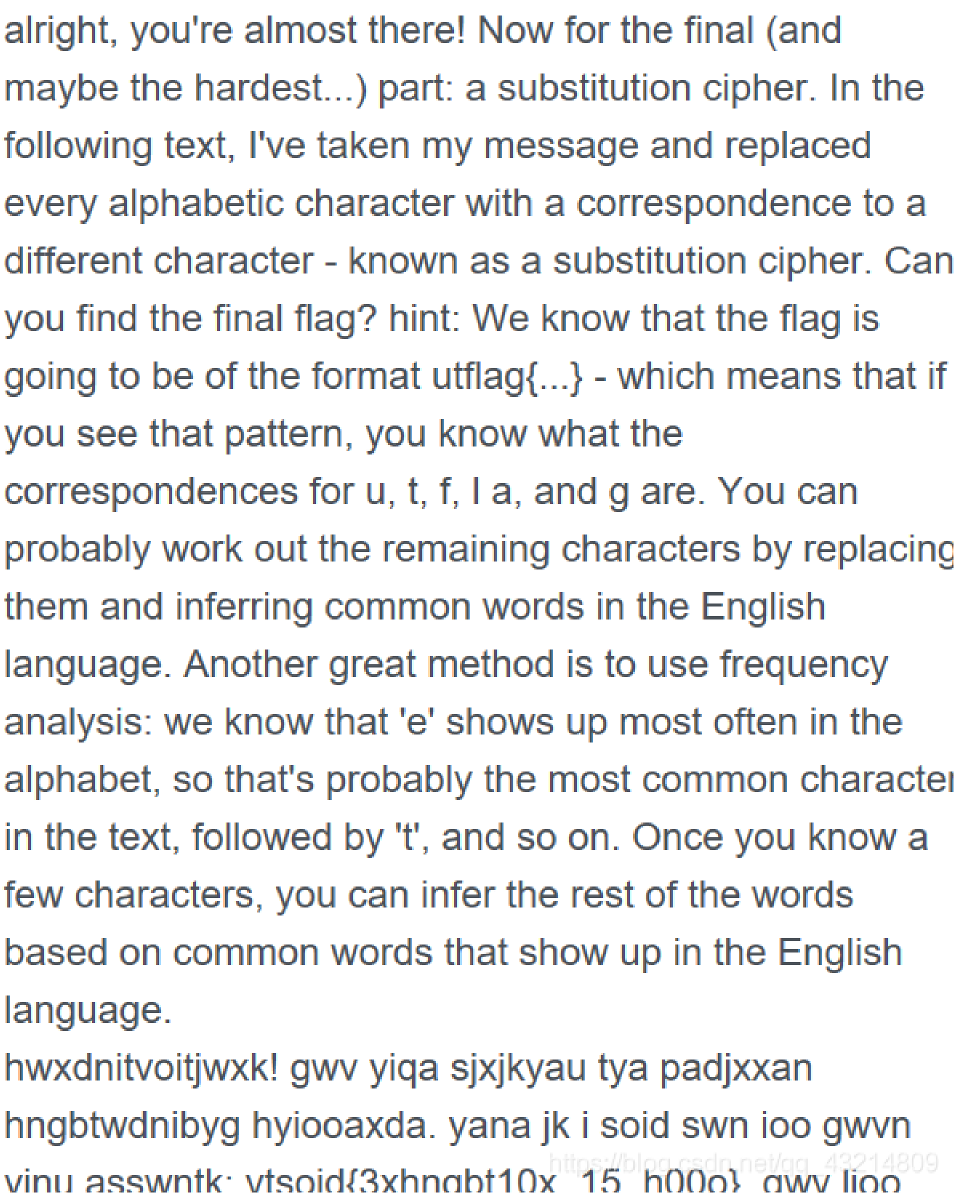
```
TmV3IGNoYWxsZW5nZSEgQ2FuIHlvdSBmaWd1cmUgb3V0IHdoYXQncyBnb2luZyBvbiBoZXJlPyBJdCBsb29rcyBsaWtlIHRoZSBsZXR0ZXJzIGFyZSBzaGlmdGVkIGJ5IH
```

这样的话，直接base64求解：



```
New challenge! Can you figure out what's going on here? It looks like the letters are shifted by some constant. (hint: you might want to start
looking up Roman people).
kvbsqrd, iye'bo kvwycd drobo! Xyg pyb dro psxkv (kxn wkilo dro rkbnocd...) zkbd: k celcdsdedsyx mszrob. Sx dro pyvvygsxq dohd, S'fo dkuox wi
wocckqo kxn bozvkmon ofobi kvzrklodsm mrkbkmdob gsdr k mybboczyxnoxmo dy k nsppoboxd mrkbkmdob - uxygx kc k celcdsdedsyx mszrob. Mkx iye psxn dro
psxkv pvkq? rsxd: Go uxyg drkd dro pvkq sc qysxq dy lo yp dro pybwkd edpvkq{...} - grsmr wokxc drkd sp iye coo drkd zkddobx, iye uxyg grkd dro
mybboczyxnoxmoc pyb e, d, p, v k, kxn q kbo. Iye mkx zbylklvi gybu yed dro bowksxsxq mrkbkmdobc li bozvkmsxq drow kxn sxpobbsxq mywwyx gybnc sx
dro Oxqvscr vkxqekqo. Kxydrob qbokd wodryn sc dy eco pboaeoxmi kxkvicsc: go uxyg drkd 'o' crygc ez wycd ypdox sx dro kvzrklod, cy drkd'c zbylklvi
dro wycd mywwyx mrkbkmdob sx dro dohd, pyvvygon li 'd', kxn cy yx. Yxmo iye uxyg k pog mrkbkmdobc, iye mkx sxpob dro bocd yp dro gybnc lkcon yx
mywwyx gybnc drkd cryg ez sx dro Oxqvscr vkxqekqo.
rghnxsdfysdtghu! qgf isak cthtuike dik zknthhkx rxqldgnxsliq risyykhnk. ikxk tu s cysn cgx syy qgfx isxe kccgxdu: fdcysn{3hrxqld10h_15_r00y}. qgf
vtyy cthe disd s ygd gc rxqldgnxsliq tu pfud zftyethn gcc ditu ugxd gc zsutr bhgvykenk, she td xksyyq tu hgd ug zse scdkx syy. iglk qgf khpgqke
dik risyykhnk!
```

很惊（扯）喜（淡）地发现还有密码，直接维吉尼亚无密钥解密走起：



alright, you're almost there! Now for the final (and maybe the hardest...) part: a substitution cipher. In the following text, I've taken my message and replaced every alphabetic character with a correspondence to a different character - known as a substitution cipher. Can you find the final flag? hint: We know that the flag is going to be of the format utflag{...} - which means that if you see that pattern, you know what the correspondences for u, t, f, l a, and g are. You can probably work out the remaining characters by replacing them and inferring common words in the English language. Another great method is to use frequency analysis: we know that 'e' shows up most often in the alphabet, so that's probably the most common character in the text, followed by 't', and so on. Once you know a few characters, you can infer the rest of the words based on common words that show up in the English language.

hwxdnitvoitjwxk! gwv yiqa sjxjkyau tya padjxxan hngbtwdnibyg hyiooaxda. yana jk i soid swn ioo gwvn yinu asswntk: vtsoid{3xhnqbt10x_15_h00o}. gwv jioo

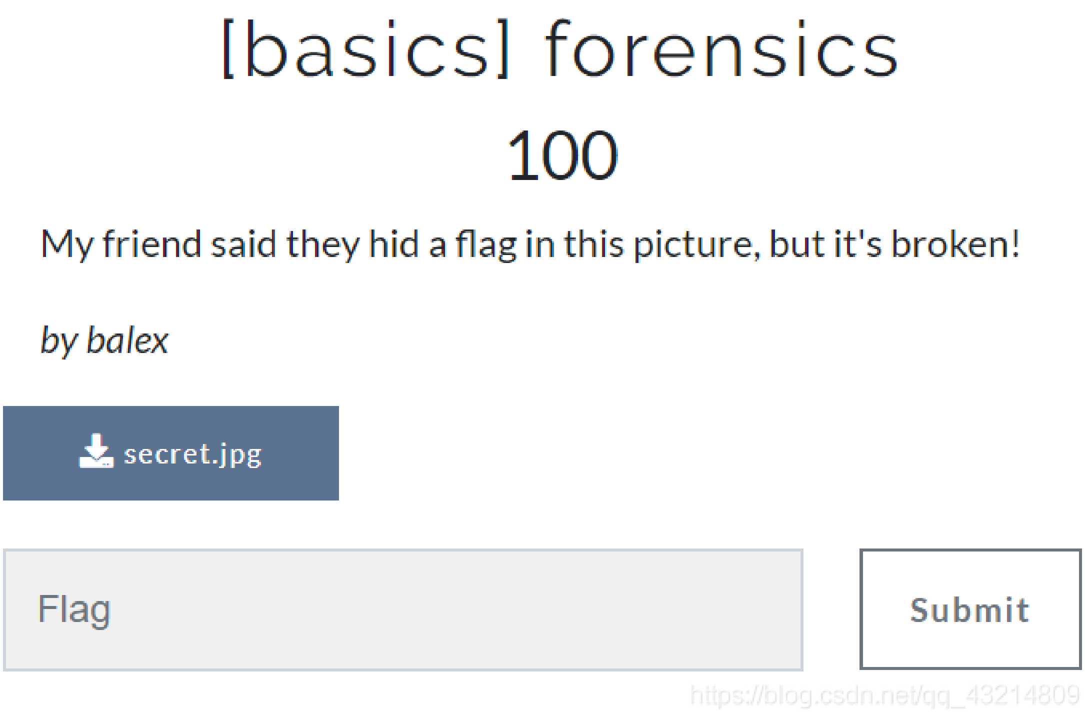
嗯，真会玩，还差最后一步，给出的提示很明确词频分析走一波：

```
congratulations! you have finished the beginner cryptography challenge. here is a flag for all your hard
efforts: utflag{3ncrypt10n_15_c00l}. you will find that a lot of cryptography is ?ust building off this
sort of basic knowledge, and it really is not so bad after all. hope you en?oyed the challenge!
```

顺利得到flag。

二、[basic]forensics：

这题目也蛮简单的，直接view-source看源码就行了：

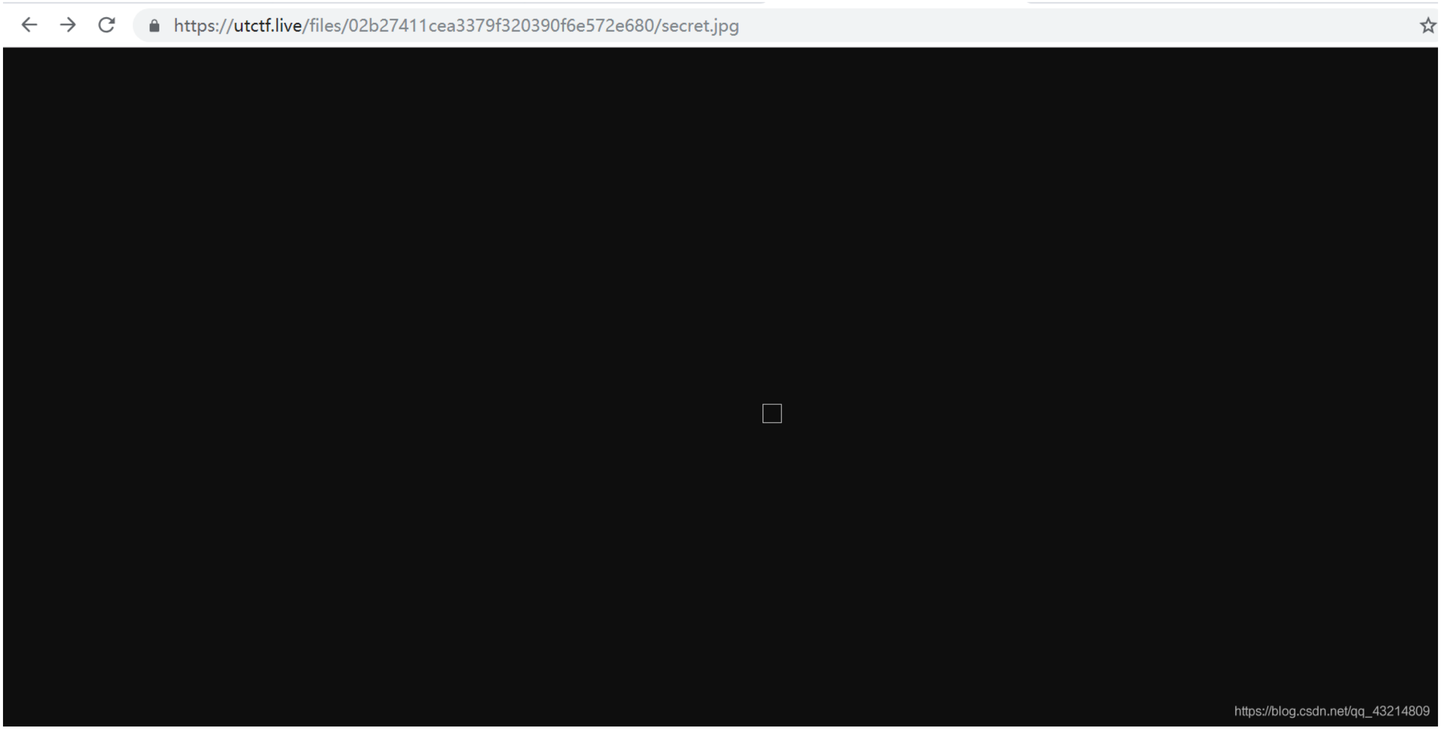


[basics] forensics

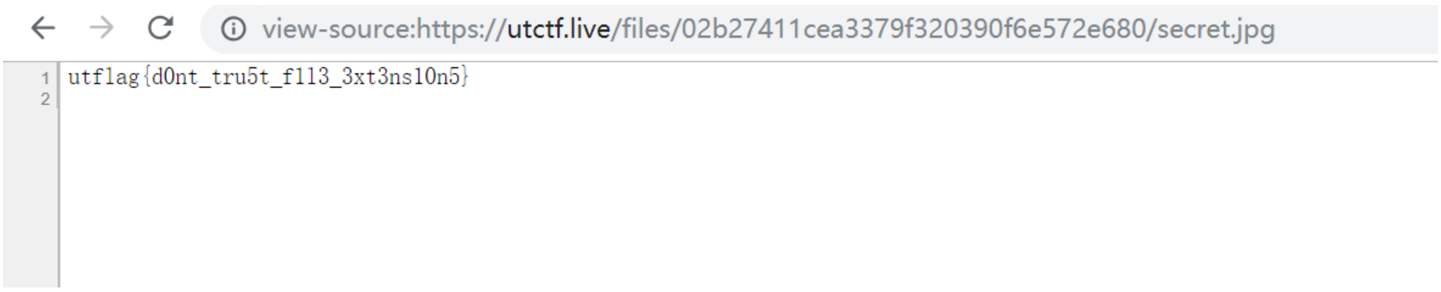
100

My friend said they hid a flag in this picture, but it's broken!

by balex



https://utctf.live/files/02b27411cea3379f320390f6e572e680/secret.jpg

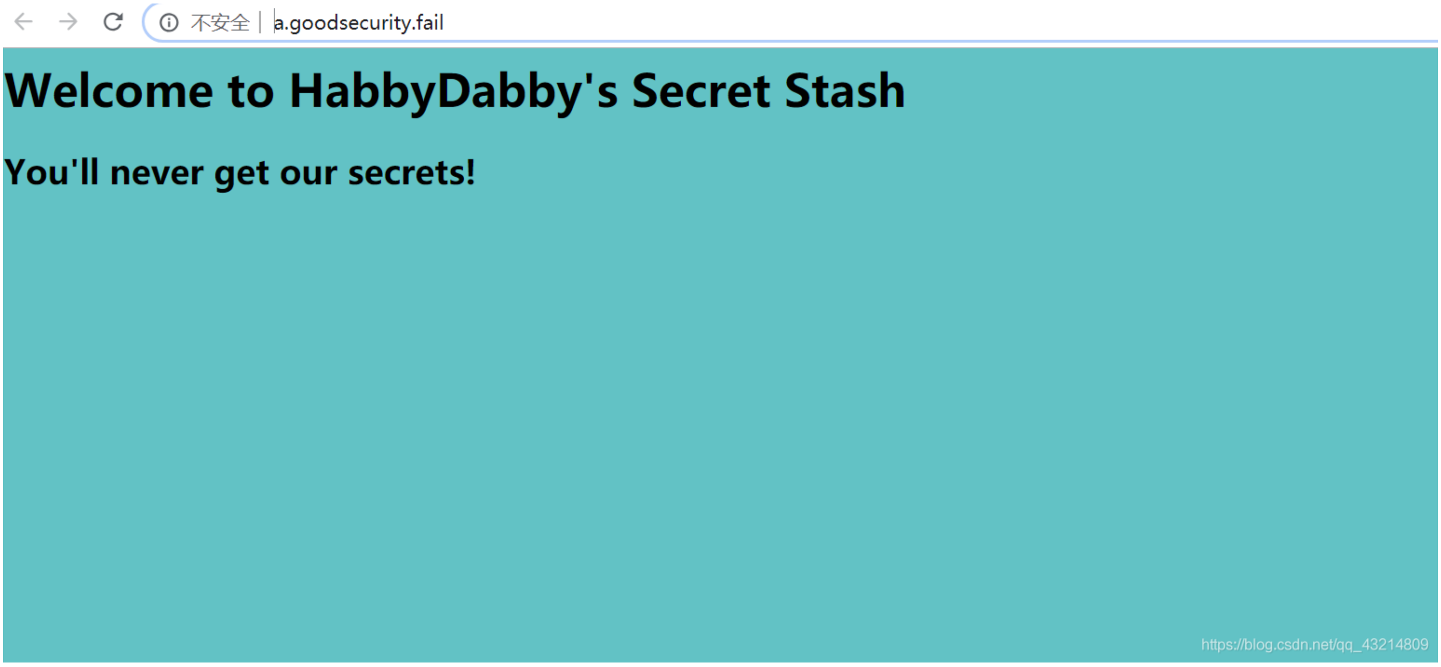


view-source:https://utctf.live/files/02b27411cea3379f320390f6e572e680/secret.jpg

```
utflag{d0nt_tru5t_f1l3_3xt3ns10n5}
```

三、**HabbyDabby’s Secret Stash**：

这道题目刚出来的时候是1000分，等到结束的时候贬到650分了，反倒是之前450的题目涨到了1000分，这道题其实也蛮简单的，直接试出来的：

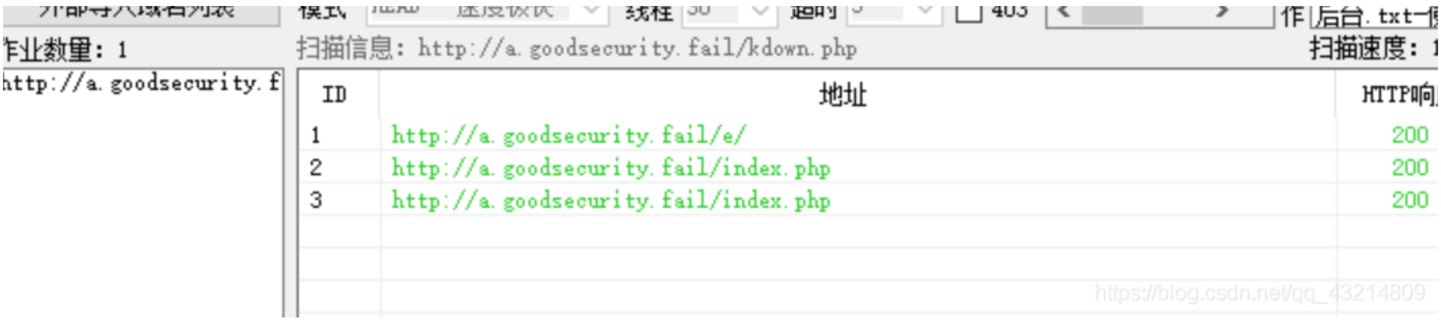


不安全 | a.goodsecurity.fail

Welcome to HabbyDabby's Secret Stash

You'll never get our secrets!

打开网页后，无论是view-source还是burpsuite抓包都找不到有用的信息，或是说根本没有信息，最后只能上御剑扫下后台，还挺幸运的找到了：



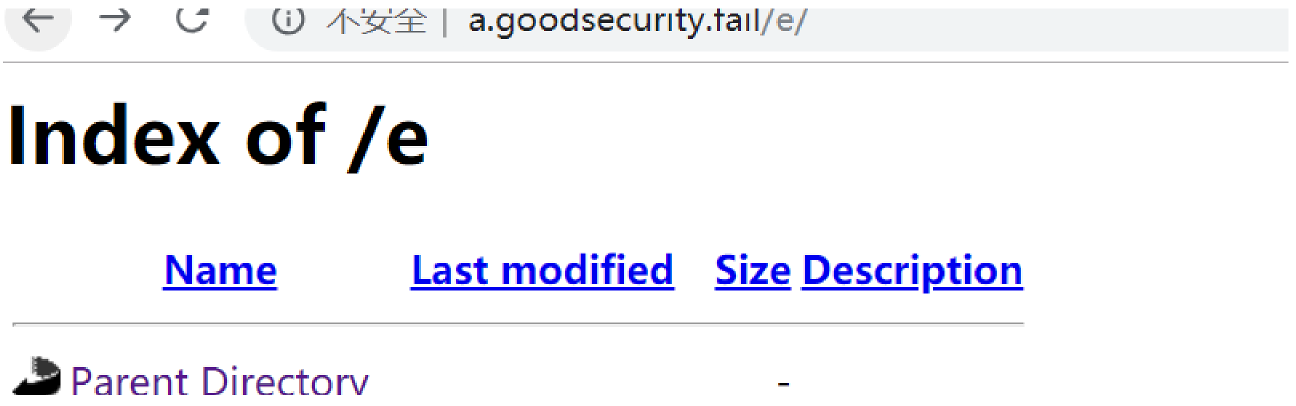
乍业数量：1

扫描信息：http://a.goodsecurity.fail/kdown.php

扫描速度：1

http://a.goodsecurity.f

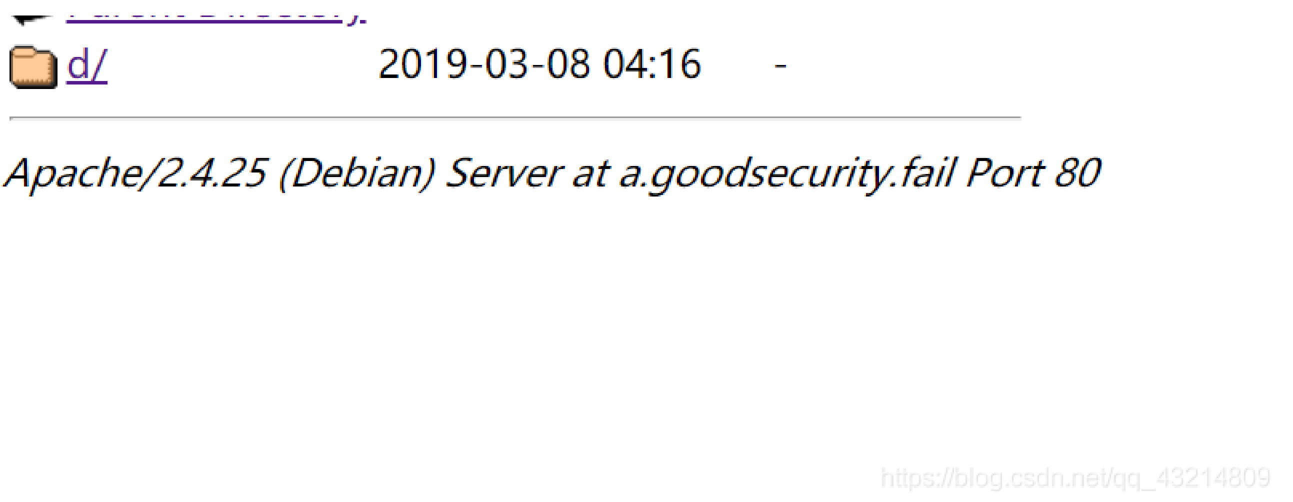
ID	地址	HTTP响
1	http://a.goodsecurity.fail/e/	200
2	http://a.goodsecurity.fail/index.php	200
3	http://a.goodsecurity.fail/index.php	200



不安全 | a.goodsecurity.fail/e/

Index of /e

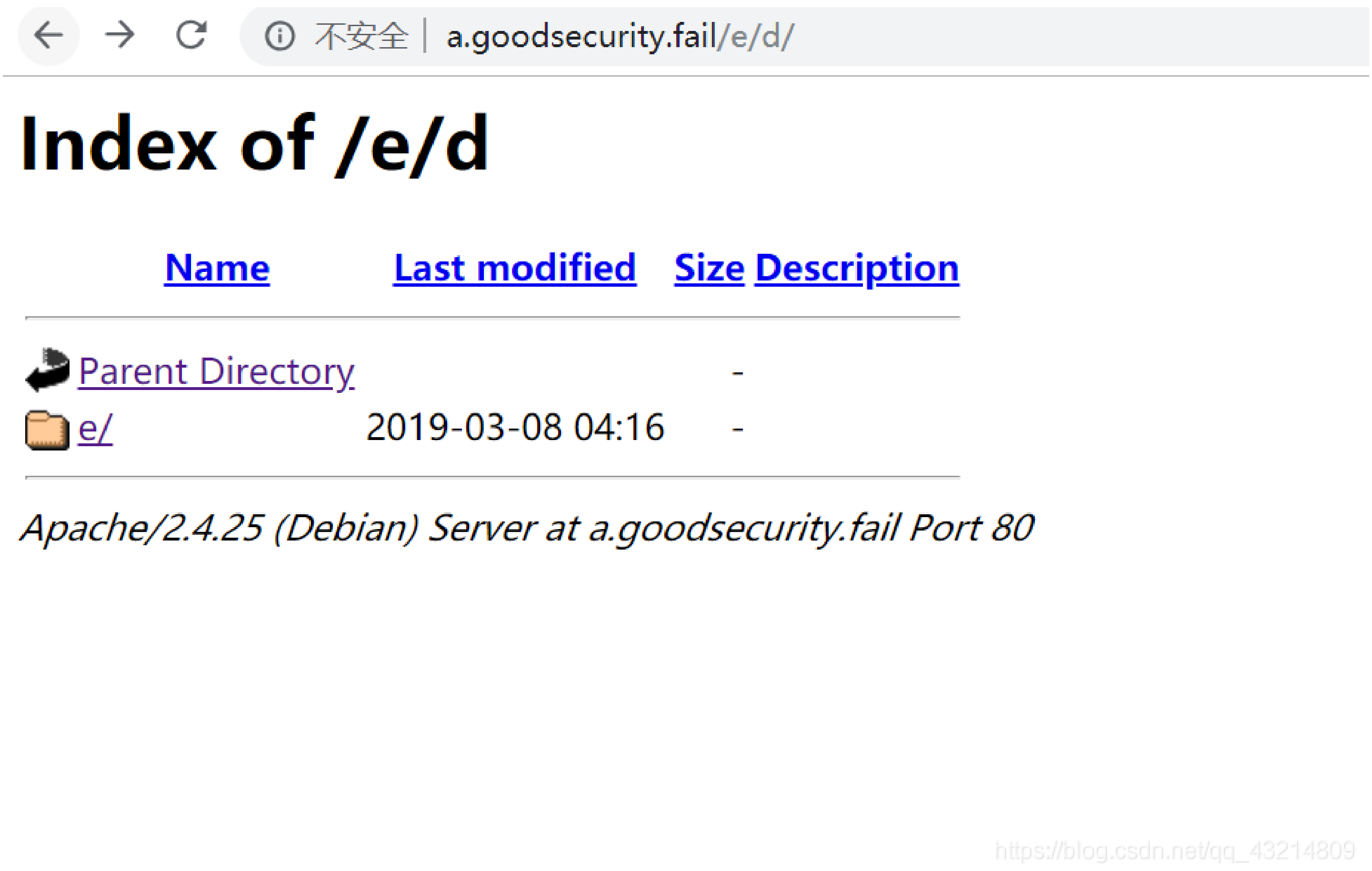
Name	Last modified	Size	Description
Parent Directory		-	



Name	Last modified	Size	Description
Parent Directory		-	
d/	2019-03-08 04:16	-	

Apache/2.4.25 (Debian) Server at a.goodsecurity.fail Port 80

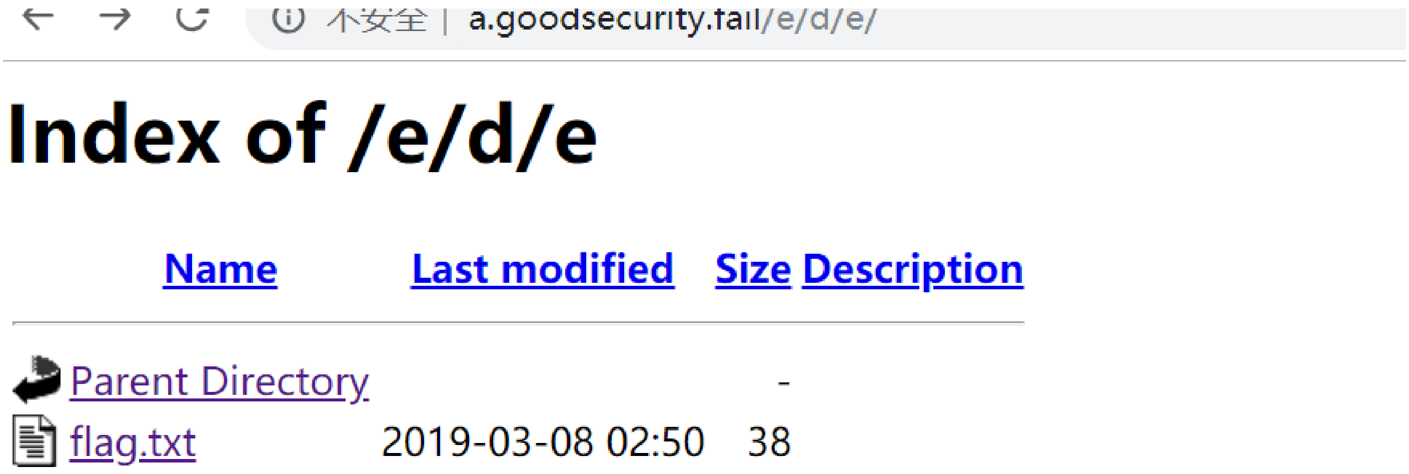
直接下面的文件夹一路点下去：



Index of /e/d

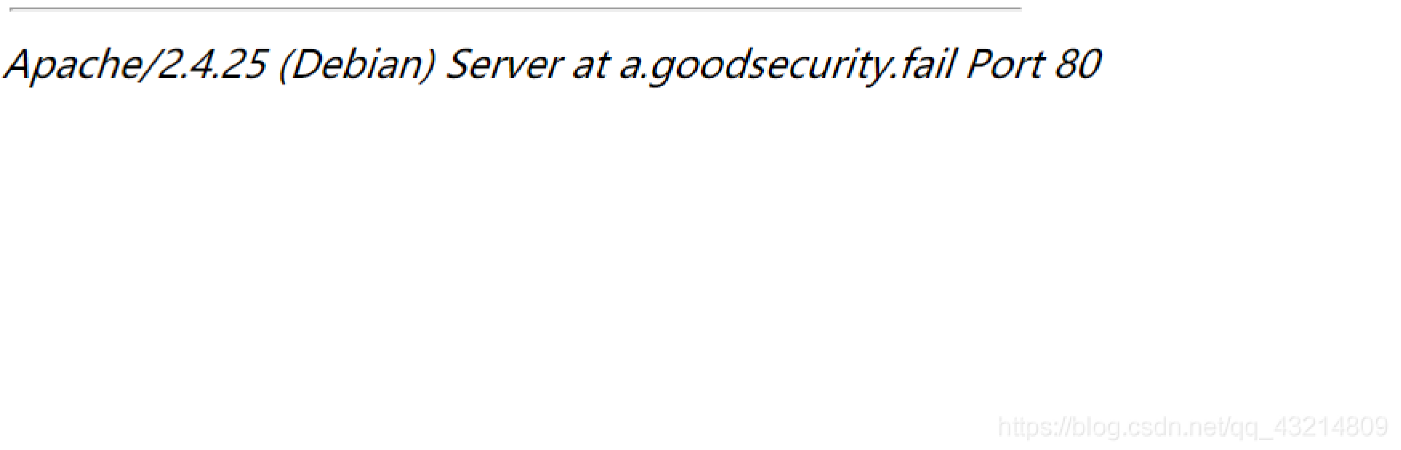
Name	Last modified	Size	Description
Parent Directory		-	
e/	2019-03-08 04:16	-	

Apache/2.4.25 (Debian) Server at a.goodsecurity.fail Port 80

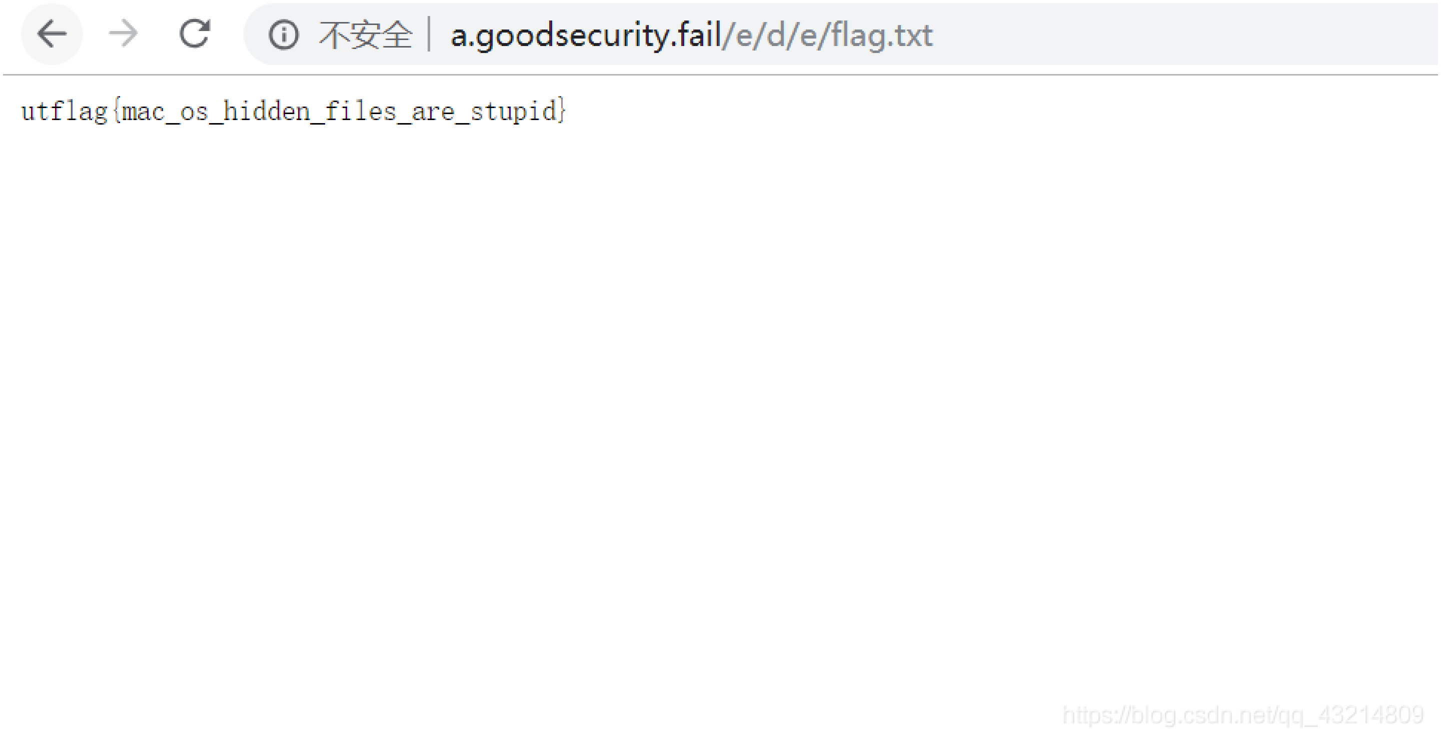


Index of /e/d/e

Name	Last modified	Size	Description
Parent Directory		-	
flag.txt	2019-03-08 02:50	38	



Apache/2.4.25 (Debian) Server at a.goodsecurity.fail Port 80



```
utflag{mac_os_hidden_files_are_stupid}
```

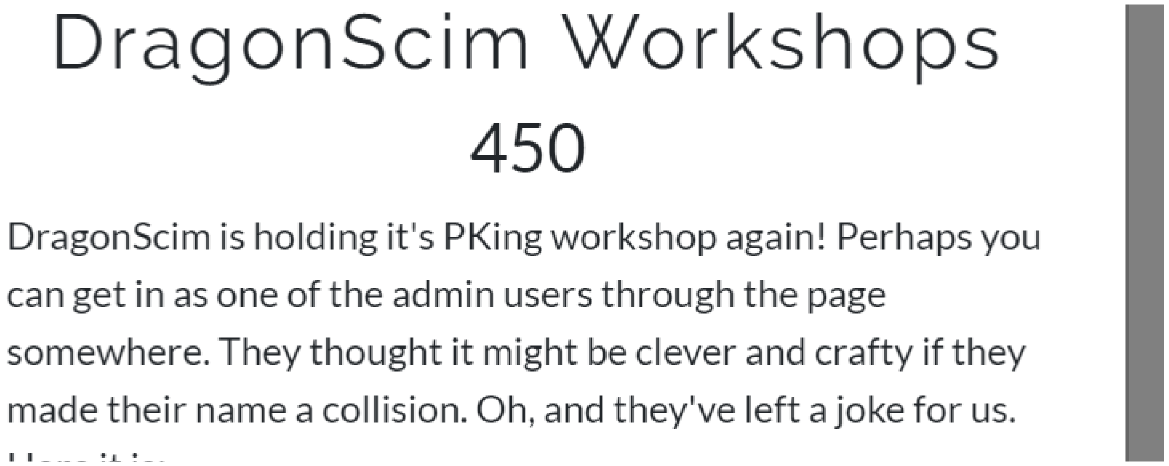
直接拿到flag。

二、重解：（有必要说明的是在我想重解的时候，UTCTF解题链接好像已经关掉了，AeroCTF的还是可以用的）

1.DragonScim Workshops（UTCTF)：

这道题怎么讲，一开始是Web题目里分值最少的，只有450分，但是遗憾的是一开始并没有队伍做出来，后来给出了许多提示，也把分数提高到100分，但比赛结束后也只有15支队伍解出。

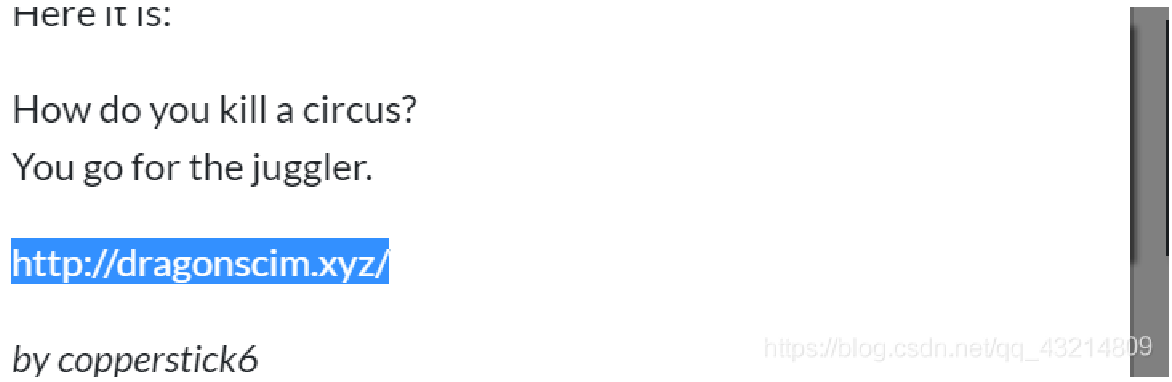
一开始是这样的：



DragonScim Workshops

450

DragonScim is holding it's PKing workshop again! Perhaps you can get in as one of the admin users through the page somewhere. They thought it might be clever and crafty if they made their name a collision. Oh, and they've left a joke for us.



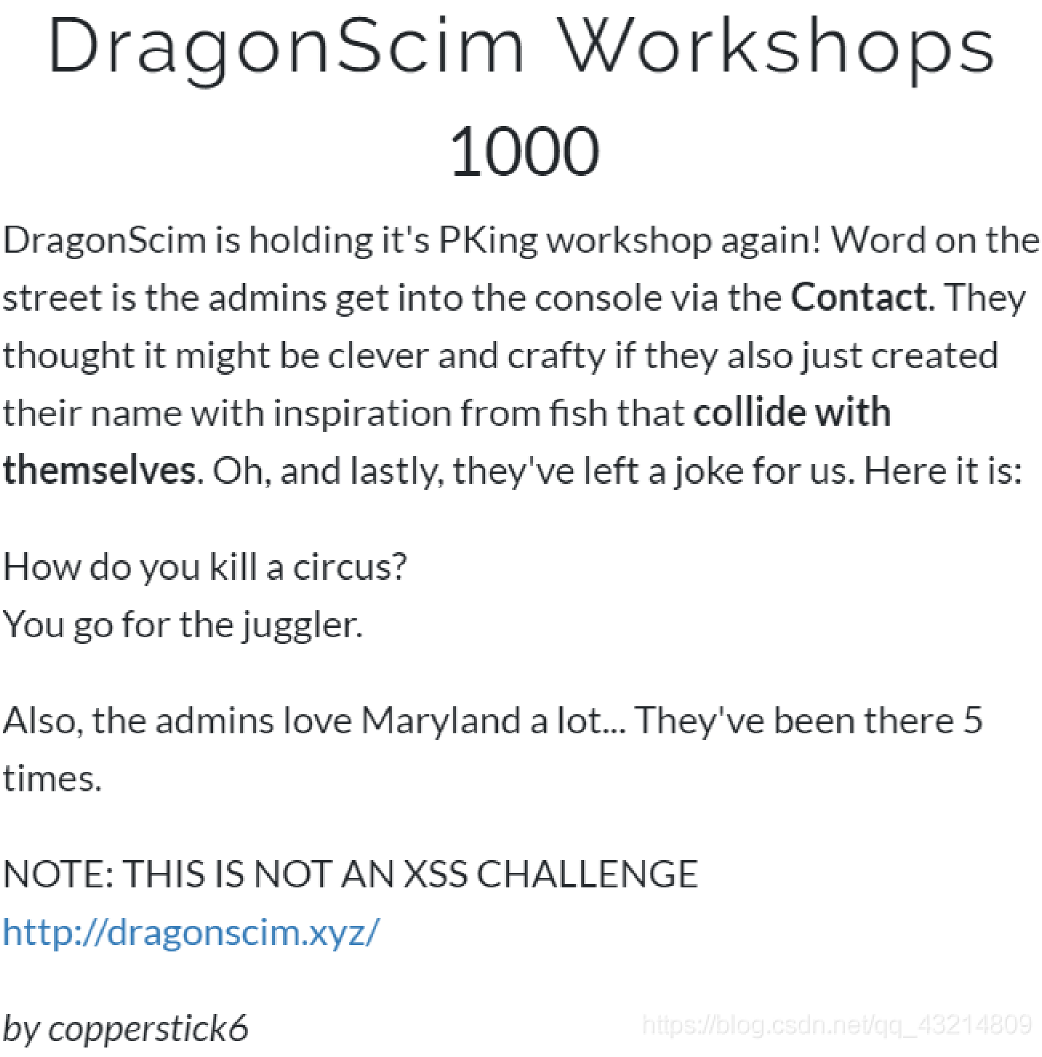
Here it is:

How do you kill a circus?
You go for the juggler.

http://dragonscim.xyz/

by copperstick6

后来就这样了：



DragonScim Workshops

1000

DragonScim is holding it's PKing workshop again! Word on the street is the admins get into the console via the **Contact**. They thought it might be clever and crafty if they also just created their name with inspiration from fish that **collide with themselves**. Oh, and lastly, they've left a joke for us. Here it is:

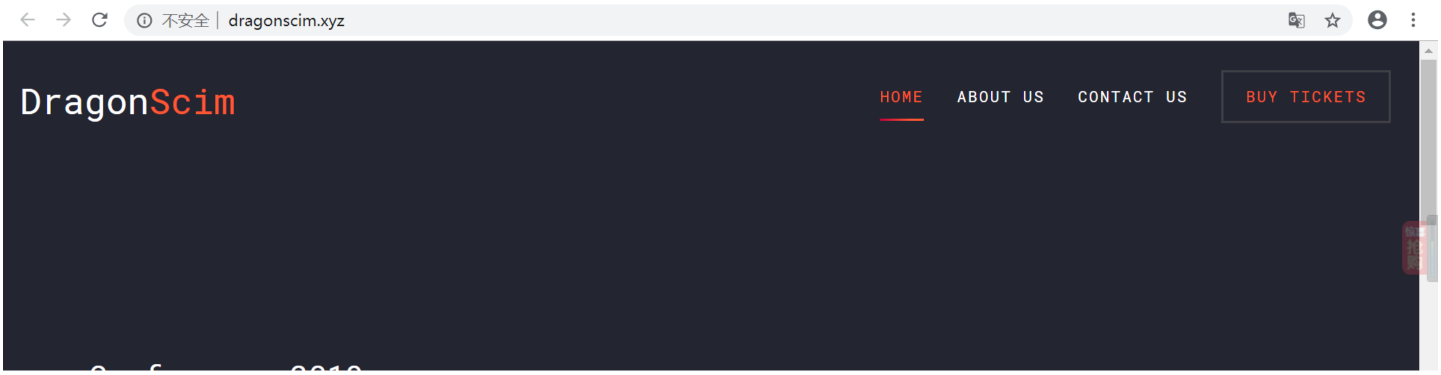
How do you kill a circus?
You go for the juggler.

Also, the admins love Maryland a lot... They've been there 5 times.

NOTE: THIS IS NOT AN XSS CHALLENGE
http://dragonscim.xyz/

by copperstick6

真是可以了，但是我还是不会，我已知的所有方法全部都试了一遍，但是真的没个卵用，绝望之情溢于言表，后来看了大佬的writeup，又把这道题重解了一遍，真是…，这解题思路我尼玛，学不来啊。一开始的界面是这样的：



Conference 2019

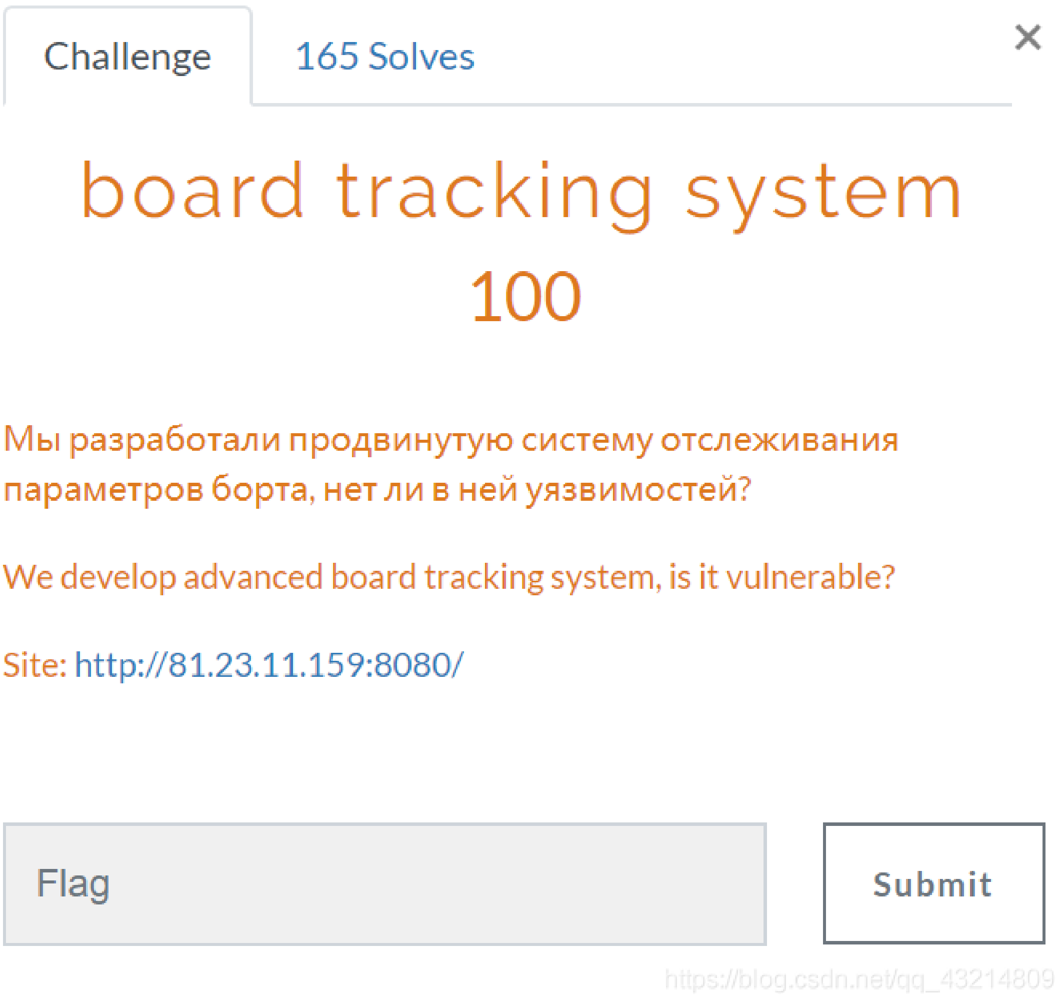
Dragon Scimitar Conference 2019

做过题目的基本上都会到CONTACT US这个页面去，因为在这里可以进行输入名字的操作，也只可能在这个地方出点问题，（由于UTCTF的网站关闭了，我做题的时候也没多少图片，所以只能脑补）大佬的思路是先传参："?name[]=aaa"然后会出现一个关于MD5的提示（我提交的时候后怎么没看到(・ε・ '))，之后的操作我就知道了什么是大佬和我这种菜鸡的区别，对问题的把握以及解读差距还是很大的。我还是没懂他咋得出来的结论，大佬最后落脚点是在php弱类型比较上，因为根据题目提示是要使得输入的字符串和其本身MD5计算后哈希值一致，通过php的这个特性，可以轻易的构造true出来，可以通过在前面添加0e绕过，最后输入得到flag。

大佬的具体writeup：https://graneed.hatenablog.com/entry/2019/03/11/122020

2.board tracking system：（AeroCTF）

题目描述如下：



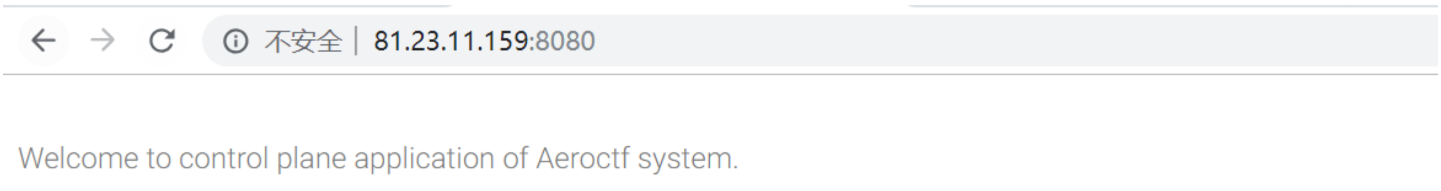
board tracking system

100

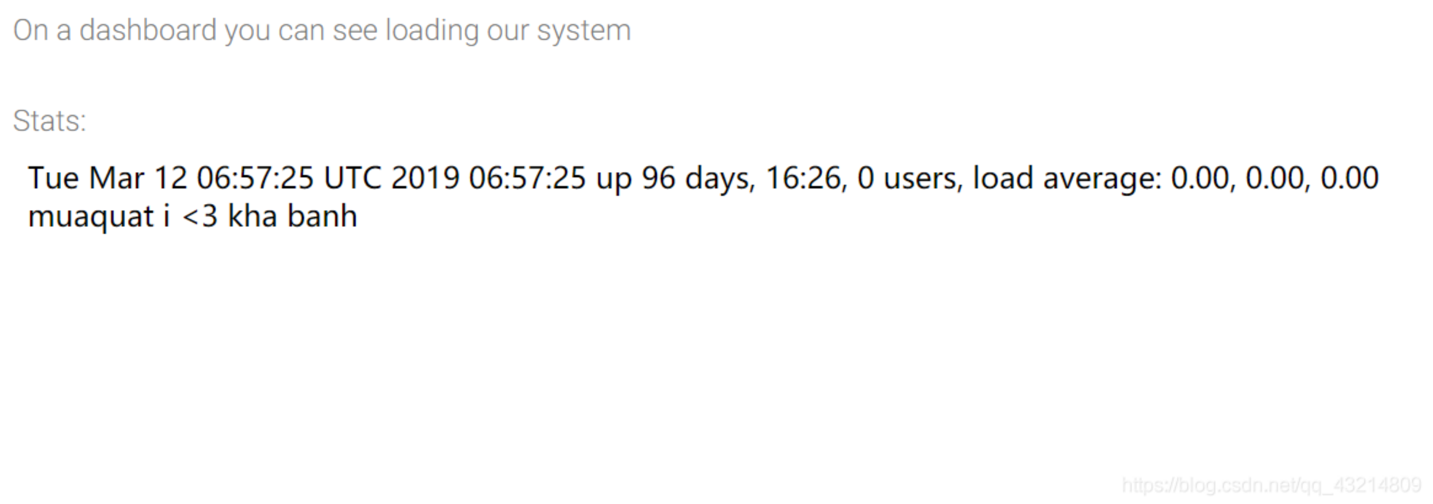
Мы разработали продвинутую систему отслеживания параметров борта, нет ли в ней уязвимостей?

We develop advanced board tracking system, is it vulnerable?

Site: http://81.23.11.159:8080/



Welcome to control plane application of Aeroctf system.



On a dashboard you can see loading our system

Stats:

```
Tue Mar 12 06:57:25 UTC 2019 06:57:25 up 96 days, 16:26, 0 users, load average: 0.00, 0.00, 0.00
muaquat i <3 kha banh
```

我在一开始拿到这道题的时候是没有思路的，我的工具箱里好像没有找到能解决这种题目的工具，说来惭愧，还是知识的问题，了解的太少，以前没见过，这次就当长见识了，涨姿势。

因为链接关闭了，我也不好直接说，有兴趣的可以直接去看大佬writeup：https://rawsec.ml/en/AeroCTF-2019-write-ups/

3.VisageNovel：（UTCTF)

这道题我一开始就没看，也是比较遗憾的。因为心思一直在第一道web题目上，到后来比赛结束也没看上一眼，但最后解出的人数也是蛮少的，直接附上writeup链接好了：

https://graneed.hatenablog.com/entry/2019/03/11/121643

三、小结：

1.这次比赛挺集中的，加上自己的时间也不是很多，所以有的题目也没顾上仔细思考，以后的话尽量降低这样的做题频率，还是先弄明白前面的一部分知识，再去了解下一部分，现在还是有点着急了，应该先去丰富丰富学识。

2.还是像上次讲的，差距还是有的，既要往前看，也要多看看自己，有些天赋差距那真是比不上，付出和时间的差距也比不上，但有些知识上的差距是可以弥补的，自己以后也要多注意。

3.日常感谢乐于分享writeup的大佬们，谢谢你们的分享；无论是在ctftime上的，还是在github上的，本人在这里一并感谢。