




BUUCTF: 蜘蛛侠呀

原创

末初  于 2020-11-12 15:45:32 发布  1252  收藏 5

分类专栏: [CTF_MISC_Writeup](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/mochu7777777/article/details/109645038>

版权



[CTF_MISC_Writeup](#) 专栏收录该内容

246 篇文章 45 订阅

订阅专栏

<https://buuoj.cn/challenges/#%E8%9C%98%E8%9B%9B%E4%BE%A0%E5%91%80>

Challenge

204 Solves



蜘蛛侠呀

1

将你获得的明显信息md5加密之后以flag{xxx}的格式提交。注意: 得到的flag 请包上 flag{} 提交

 fa90d5eb-20...

Flag

Submit

<https://blog.csdn.net/mochu7777777>

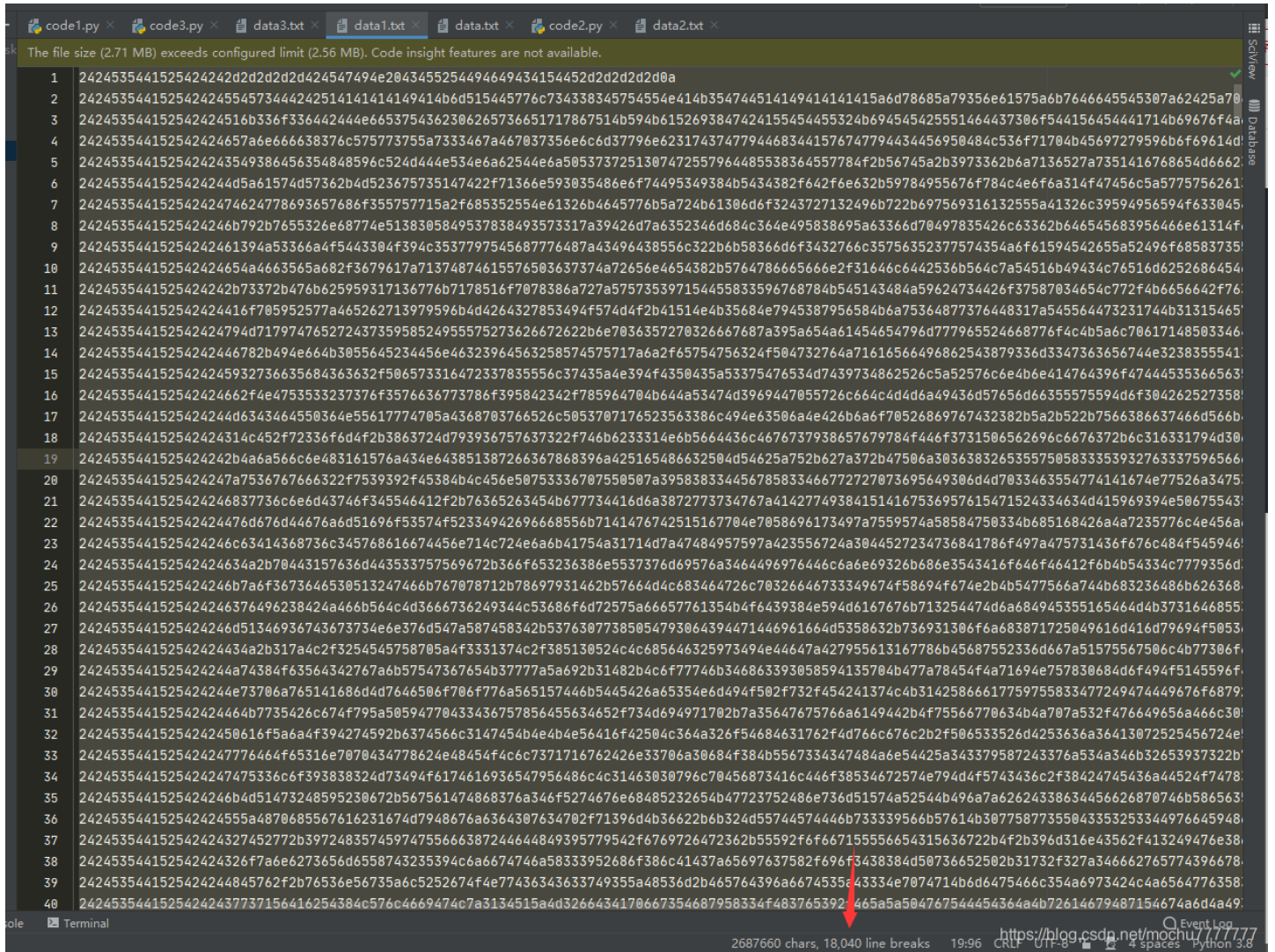
out.pcap


```

with open('data.txt', 'r') as file:
    res_list = []
    lines = file.readlines()
    print('[+]去重之前一共{0}行'.format(len(lines)))
    print('[+]开始去重, 请稍等.....')
    for i in lines:
        if i not in res_list:
            res_list.append(i)
    print('[+]去重后一共{0}行'.format(len(res_list)))
    print(res_list)

with open('data1.txt', 'w') as new_file:
    for j in res_list:
        new_file.write(j)

```



将十六进制数据转为字符

```

import binascii

with open('data1.txt', 'r') as file:
    with open('data2.txt', 'wb') as data:
        for i in file.readlines():
            data.write(binascii.unhexlify(i[:-1]))

```



```

1  $$START$$-----BEGIN CERTIFICATE-----
2  $$START$$UesDBBQAAAAIAKmqTEwLsC84WTUNAK5GDQAIAAAAZmxhZy5naWZkvFUE0zbBZp0
3  $$START$$Qk3o3dBDNFsUcB0besfQqxgQKYKaRi8GBAUEDU2KIEEBUQFD70oTAVEDAqKigoJi
4  $$START$$eznff87LWwSuZs3FzFp75nLm7ynb1t7GyDh4AWgGyD4E1PHLSogpKEIryYKoiAMU
5  $$START$$5I8dV5HHYLRMDNSNjbTNjPsrQ0trUydHU86EWx0+VtZ+9s6+jq6RzsQAgheMfb2
6  $$START$$MzAWMw6+MR6usQGB/q6nY05HnotISI8KT4B/d/nc+YXIUgoxLNoj10GELZUuuba2
7  $$START$$tb6xi6Who5WwqZ/hSRUNA2kFEwkZrKa0mo2Crq2Ikr+iu1a2UA2L9YIVY0c0EC
8  $$START$$ky+vU2nhwNq80XISx8I5s1z9BmzcR4mhl6NIX8iZc6mpIX5B1c6+deEh9VFna10i
9  $$START$$a9JS6j0TC009L57yuEhvwHzCiD8U12+kX6mo42vL5ucR7ut5JoaVEBZRIohX75Y
10 $$START$$eJFcVzh/6yazq7HtaUvP67JrenFT8+WdxFefn/1dldbSkVlzTQkICLVQmbRhdTj
11 $$START$$+s7+GkbYY1q6wkqxQopx8jrzWW59qTEX3YghxKTQCHJYb64Bo7Xp4eLw/KFvd/v8
12 $$START$$AopYRwzFRbq9yYkMBd2XSIOWMO+AQNK5hNyE8yVXKjy6Hw7dH1zTudG21tk11Tep
13 $$START$$MyqvtvRrCsYXRIUWRsbfRb+np65rp2fghz9ZeJaEFTYmwyERFhwoLKLZLpaqHP3FK
14 $$START$$Ffx+INfK0uR4EnF29dV2XWUeqzj/euGV20P62vJqaefIhbT8y3m3G66VtN285UA4
15 $$START$$Y2sf5hNcc/Pes1dr3x5UL7CZ90CPCZS7TVSMt9sHbRLZRwLnKnAGd9oGDE56ecR
16 $$START$$f/NGS27705vcgsxo9XB4/xYdpKdJSGM9iDpUnrflMMjICmWemf5UuYMo0BbRsXZ
17 $$START$$hCfEP6NUawtpZChp7fRlPspqvR5c8lInCpJNBkjopRh1vt28+Z+R+uf8f7FmVfK
18 $$START$$lLE/r3om0+8crMy96uv72/tkb31NkVdClFvsy8evyx0Do71Pebi1fv7+1lc1yM0L
19 $$START$$+JvLnH1awjCnd0Q8rf6xh9jBQeHf2PMTbZu+bz7+GPj0682e5WPX3592v37Yefn
20 $$START$$Zu6vvf2/u99/E8KLEnPu36puPP29X83DVxx3FgrnpsivI0mMp3F5TWAAGNwRj4u1
21 $$START$$h7s1nmCto4UFA/+v6RcEKgw4Amj8rw74vzABwI8AQAgSiWatRC4cMAYi9NPgUCR
22 $$START$$GmgDgmjQioSW0R3IBifhUkqAGgBQqppNpXiasIzuYwJXXG6P3KhqBjJr5wLNEjP
23 $$START$$LcAchs14WhaftEnqLrNjkAuJ1qMzGHIWVzB5VrJ0DRr4shAxoIzGW1CoglHOTYFP
24 $$START$$cJ+pD1WcmD53uuig+6oe268nU77miWj4dIivDljni2Khn5CAodoFA/kKT3Lwy5m8
25 $$START$$kzo6sde0Q2GfKvpxq+xiy1F+WfMLh4drLp2fS3IgDXi0gN+KTWjtkh26HkbchC
26 $$START$$cvIb8BjFkVLM6fbsI4LShomrWZfewa5K0d98NYMaggkq2TGMjhIE5QeFMK71dhU6
27 $$START$$mQ4i6t6ssNn7mTzXtX4+Sv0w8PTy0d9DqD1afMSXc+si10ojh8qrPIamAmy10PSf
28 $$START$$CJ+1zL/2TTWxpZ0317L/8Q0RLlhVF2YsINDdzByUa1gxkEhuR3mfzQUUGPLKw0og
29 $$START$$Jt80cVBCvzkWTsgeK7wzZi+1H+Lowtk4hc90XYA5pKGzxE0Jq1Nux0hMoIQEYoC
30 $$START$$NspjvQAhmMvFPopowjVQWDkTEBje5NmIOP/s/EBA7L18XfawYuX3GrIGDIgohy/
31 $$START$$FKw5B1g0yZPYGpC46uxVEV4e/sMiIqp+z5dvuvjaID+0uVgpcKJpzS/GfIejfL0P
32 $$START$$PaoZj09BtY+ctV1LGEKKNVAv0BPL6J2oThF1v/MvlgL+/Pe3RmBScj6A0rRTVrNZ
33 $$START$$wvF0eInppCGxbNHE0LlsgqgbBn3pj0h08KUG3CGHJnTbZ43yXrc7jS34k2e972+y
34 $$START$$tu3l09882MsI0ata16TyVHLL1F00yLpEhsALDo8SFRWNYMOWCCl/8B6ECjDR0tx4
35 $$START$$kMQG2HYR0g+VuaGhh7j4oRtgnhHR2eKGr7RHnsmQWJRTKIjzbbC8cdVbhptkXecT
36 $$START$$UZHphUgab1gMyHgjcdv4p/q9mK6b+k2MUtEtDks39VkwK0wXw5PC52S4IvdYHc
37 $$START$$2tRw+9rH5tYtuVf8rDdHI9WyT/g1rdr6+UY/ofqUUFt1V6r+0+9m1ncV/A2IGn8k
38 $$START$$2oznbsemeXt259LjfttjX39Rho8LACze1v7X/1o488MPsfRP+1s/2z4fbvWt9fxL
39 $$START$$HEv/+vSnVsZLRRg0NwCcC07I5ZHSm+FWd9jftSZC3NptqKmduFL5JisBLJedwcX9
40 $$START$$7sqVAT8LWLFiGLz14QZM2fCAppfsThyX30H7e9/FZP6gTDT6JKraFyHqTg3mJI6
41 $$START$$Ceqb02vCn4LRJLlAJglwYtTcJLcu5RY/EtP2K/w4jND/Em5pcF8EibaUepVCEvC

```

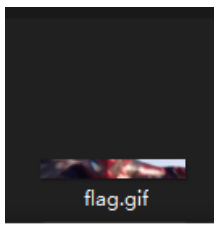
去掉首尾两行，再去掉 \$\$START\$\$ 和换行，Python将base64解码以字节流形式写成 zip

```

import base64

with open('data3.txt','rb') as file:
    with open('res.zip','wb') as new_file:
        new_file.write(base64.b64decode(file.read()))

```



flag.gif



时间隐写，真是我见过最骚的隐写，Ubuntu 下使用 identify

```
identify -format "%T" flag.gif  
20505020505020502050202020502020202050502050205020502050505050502020505020202050205050205020206666
```

20 替换为 0

50 替换为 1

```
011011010100010000110101010111110011000101110100
```

```
>>>int('011011010100010000110101010111110011000101110100',2)  
120139720634740
```

```
>>> hex(120139720634740)  
'0x6d44355f3174'
```

```
>>> binascii.unhexlify('6d44355f3174')  
b'mD5_1t'
```

```
>>> hashlib.md5('mD5_1t'.encode('utf-8')).hexdigest()  
'f0f1003afe4ae8ce4aa8e8487a8ab3b6'
```

```
flag{f0f1003afe4ae8ce4aa8e8487a8ab3b6}
```