

BUUCTF-MISC-[ACTF新生赛2020]base64隐写

原创

[D2FuZ3hpYW9xaW4](#) 于 2022-01-14 22:23:57 发布 148 收藏

分类专栏: [CTF MISC base](#) 文章标签: [其他 python](#)

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/Briteny_/article/details/122503287

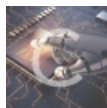
版权



[CTF](#) 同时被 3 个专栏收录

1 篇文章 0 订阅

订阅专栏



[MISC](#)

1 篇文章 0 订阅

订阅专栏



[base](#)

1 篇文章 0 订阅

订阅专栏

1、题目描述:

题目

解题快手榜



[ACTF新生赛2020]base64 隐写

1

得到的 flag 请包上 flag{} 提交。

 attachment.tar

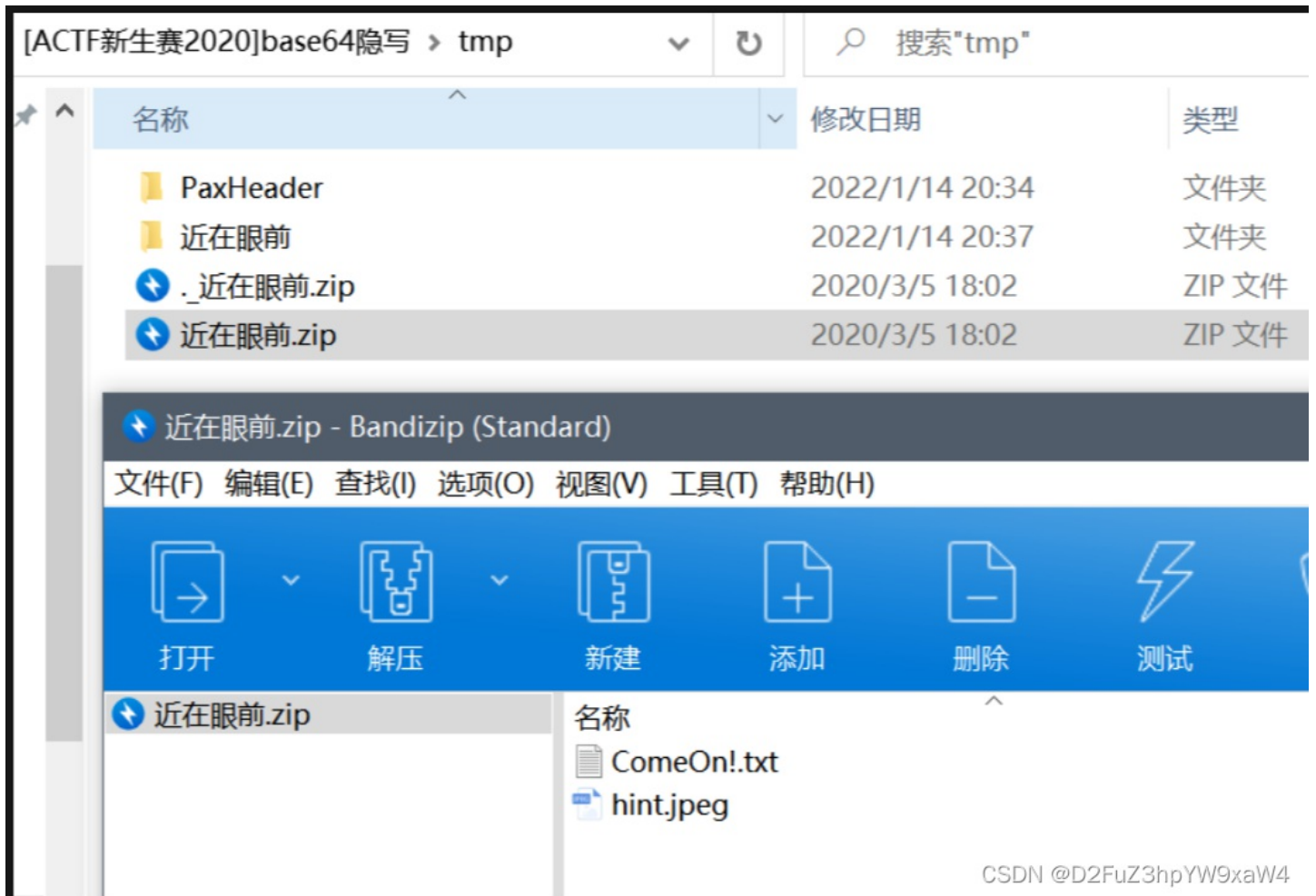
Flag

提交

CSDN @D2FuZ3hpYW9xaW4

2、下载文件

打开之后发现一个文件夹，两个压缩包，文件夹里的压缩包打不开，而_近在眼前.zip也打不开，只有近在眼前.zip可以打开



近在眼前文件夹



二维码没什么东西，ComeOn!.txt中是一堆base64编码

如下所示：

STJsdVkyeDFaR1U4YVc5emRISmxZVzArQ2c9PQ1=

STJsdVkyeDFaR1U4YzNSeWFXNW5QZ289DQ==
STJsdVkyeDFaR1U4WTNOMGNtbHVaejRLDV==
STJsdVkyeDFaR1U4WTNOMFpHbHZQZ289DQ==
STJSbFptbHVau0J0wVhodU1EazVPUW89Dd==
Q2c9PQ1=
ZFh0cGjtY2dibUZ1W1h0d1lXTmxJSE4wWkRzSw1=
Q2c9PQ0=
WTJoaGNpUmpZVnR0wVhodVhTd2dZMkpiY1dGNGJsmDdDZz09DU==
Q2c9PQ1=
YVc1ME1HTnrjQ2h6ZEhKcGjtY2dZU3dnYzNSeWFXNW5JR21wQ2c9PQ2=
ZXdvPQ1=
SUNBZ31HbG1LR3V1YzJsn1pTZ3BJRDRnWk1emFYcGxLQ2twQ2c9PQ3=
SUNBZ01DbWdJb0J5W1hSMWntNGdNVHNLDb==
SUNBZ01HVnNjM1VnYVdZb11TNXphWHBsS0NrZ1BDQm1Mbk5wZw1Vb0tTa0sN
SUNBZ01DQwdJQ0J5W1hSMWntNGdMVEU3Q2c9PQ0=
SUNBZ01HVnNjM1VLDd==
SUNBZ01Ic0sN
SUNBZ01DQwdJQ0JtYjNjb2FXNTBJR2tnUFNBd095QnBJRHdnWVM1emFYcGxLQ2s3SUDrckt5a0sN
SUNBZ21DQwdJQ2I3Q2c9PQ2=
SUNBZ11DQwdJQ1FnSUNBZ2FXWW9ZVnRwWfNB0E1HSmJhVjBwQ2c9PQ1=
SUNBZ01DYwdJY0FnSUNBZ01DYwdJSEpsZEhWeWJpYXRNVHNLDY==
SUNBZ11DQwdJQ1FnSUNBZ2FXWW9ZVnRwWfNBK11HSmJhVjBwQ2c9PQ1=
SUNBZ01DXwdJX0FnSUNBZ01DXwdJSEpsZEhWeWJpXXhPd289DX==
SUNBZ01DQwdJQ0I5Q2c9PQ0=
SUNBZ01DdwdJd0J5W1hSMWntNGdNRHNLd==
SUNBZ01IMEsN
Z1FvPQ2=
Q2c9PQ1=
ZG05cFpDQn1awE5sZEdFb2MzUn1hVzVuSudFcENnPT0N
ZXdvPQ1=
SUNBZ11HMWxiWE5sZENoa11Td3dMSE5wZW1Wd1ppaGpZU2twT3dvPQ1=
SUNBZ21HWnZjaWhwYm5RZ2FTQT1JREE3SUDrZ1BDQmhMbk5wZw1Vb2tUc2dhU3NyS1FvPQ2=
SUNBZ01Ic0sN
SUNBZ01DQwdJQ0JqWVZ0cFhTQT1JR0ZiWVM1emFYcGxLQ2t0TVMxcFhTQXRJQ2N3SnpzSw0=
SUNBZ01IMEsN
Z1FvPQ2=
Q2c9PQ0=
ZG05cFpDQn1awE5sZEdJb2MzUn1hVzVuSudJcENnPT0N
ZXdvPQ3=
SUNBZ11HMWxiWE5sZENoa11pd3dMSE5wZW1Wd1ppaGpZawtwT3dvPQ1=
SUNBZ01HWnZjaWhwYm5RZ2FTQT1JREE3SUDrZ1BDQm1Mbk5wZw1Vb0tUc2dhU3NyS1FvPQ0=
SUNBZ01Ic0sN
SUNBZ11DQwdJQ1JqWwX1cFhTQT1JR1piWk1emFYcGxLQ2t1TVMxcFhTQXRJQ2N3SnpzSw1=
SUNBZ11DQwdJQ1F2DNCeWFXNTBaaWdpS1dRaUxHTm1XMmxkS1RzSw1=
SUNBZ01IMEsN
Z1FvPQ3=
Q2c9PQ3=
YzNSeWFXNW5JR1ZrWkNoemRISnBibWnNwVn3Z2MzUn1hVzVuSudJcEx5L2xpcURtczVVSsw1=
ZXdvPQ2=
SUNBZ01ISmxjM1YwWVNoaEtUc3ZMK2FLaXVhdmorUzRnT1M5amVXdG1PZWfoT1d0bCt1c3B1V1BtT2FJa09hVnNPV3RseXpsdWJia3VKVG1pb3Jt
bGJEbnU0VgXqNDNvdjRmbW5hWgt2ci9rdw83bGtJN3BuYUxrdUkzbGtJemt2WTNtbGJEbm03amxpcUFLDW==
SUNBZ01ISmxjM1YwWVloaUtUc0sN
SUNBZ01HWnZjaWhwYm5RZ2FTTT1JREE3SUDrZ1BDTnRZwGh1T31CcEt5c3BDZz09DT==
SUNBZ01Ic0sN
SUNBZ01DQwdJQ0JqWVZ0cFhTQT1JR05oVzJsZE1Dc2dZMkpiYVYwN0NnPT0N
SUNBZ01DQwdJQ0JwWmloa11WdHBYU0ErUFNBuUtYc0sN
SUNBZ01DTwdJt0FnSUNBZ1kyRmJhU3N4WfNBc1BTTXhPd289DT==
SUNBZ01DQwdJQ0FnSUNBZ1kyRmJhVjBnTFQwZ01qc0sN
SUNBZ01DQwdJQ0I5Q2c9PQ0=

SUNBZ01IMESN

SUNBZ01HSnZiMndnWm14aFp5QT1J5FJ5ZFdvN0NnPT0N

SUNBZ31ITjBjbWx1Wn1Cak1EMGdJaUk3Q2c9PQ3=

SUNBZ01HwnZjaWhwYm5RZ2FTQT1JRzFoZuc0Z0xTQXhPeUJwSUQ0OU1EQTdJR2t0TFNrSw0=

SUNBZ01Ic0sN

SUNBZ01DQwdJQ0JwWm1obWJHRm5JQ11tSud0aFcybGRJRDA5SURBcENnPT0N

SUNBZ01DQwdJQ0FnSUNBZ1kyOXVkr2x1ZFdvN0NnPT0N

SUNBZ01DQwdJQ0JwWm1oa1lWdHBYU0FoUFNBd0tRbz0N

SUNBZ01DcWdJc0FnSUNBZ1pteGhaeUE5SUdaaGJITmxPd289Dc==

SUNBZ21DQwdJQ2JqSUNZOU1DaGpZVnRwWfNBck1DY3dKeWs3Q2c9PQ2=

SUNBZ01DQwdJQ0F2TDNCeWFXNTBaaWdpS1dRaUxHTmhXmMxkS1RzSw0=

SUNBZ01IMESN

SUNBZ01HbG1LR1pzWvdjcENnPT0N

SUNBZ11DQwdJQ1J5W1hSMWntNGdJkFpT3dvPQ1=

SUNBZ01HVnNjM1VLDX==

SUNBZ01DdWdJd0J5W1hSMWntNGdZenNLDd==

Z1FvPQ0=

Q2c9PQ0=

YzNSeWFXNW5JRzFwYm5Vb2MzUn1hVzVuSudFc01ITjBjbWx1Wn1CaUtTOHY1WwVQNXJPVknNPT0N

ZXdvPQ0=

SUNBZ01HbHVkQ0J6YVdkdU1EMGdNRHNLdQ==

SUNBZ01HbG1LR050Y0NoaExDQm1LU0E5UFNBdE1Ta3ZMMkhrdUszbHVwVG9yN1hsc1pqb3ZvUGxwS2ZubW9UcGdxUGt1S3JtbGJBSw0=

SUNBZ01Ic0sN

SUNBZ01DQwdJQ0J6YVdkdU1EMGdNVHNLdQ==

SUNBZ01DQwdJQ0J6ZEhKcGJtY2dkR1Z0Y0NBOU1HRTdDZz09DQ==

SUNBZ01DQwdJQ0JoSUQwZ11qc0sN

SUNBZ01DQwdJQ0JpSUQwZ2RHVnRjRHNLdQ==

SUNBZ01IMESN

SUNBZ01ISmxjM1YwWVNoaEtUc0sN

SUNBZ01ISmxjM1YwWlloaUtUc0sN

SUNBZ01HwnZjaWhwYm5RZ2FTQT1JREE3SudrZ1BDQnRZwGh1T3lCcEt5c3BDZz09DQ==

SUNBZ01Ic0sN

SUNBZ01DQwdJQ0JwWm1oa1lWdHBYU0E4Sud0aVcybGRLWHLNLdQ==

SUNBZ01DQwdJQ0FnSUNBZ1kyRmJhU3N4WfNBdFBTQXhPd289DQ==

SUNBZ01DQwdJQ0FnSUNBZ1kyRmJhVjBnS3owZ01qc0sN

SUNBZ01DQwdJQ0I5Q2c9PQ0=

SUNBZ01DQwdJQ0JqWVZ0cFhTQT1JR05oVzJsZE1DMGdZMkpiYVYwN0NnPT0N

SUNBZ01IMESN

SUNBZ01HSnZiMndnWm14aFp5QT1J5FJ5ZFdvN0NnPT0N

SUNBZ01ITjBjbWx1Wn1Ca01EMGdJaUk3Q2c9PQ0=

SUNBZ01HwnZjaWhwYm5RZ2FTQT1JRzFoZuc0Z0xTQXhPeUJwSUQ0OU1EQTdJR2t0TFNrSw0=

SUNBZ01Ic0sN

SUNBZ01DQwdJQ0JwWm1obWJHRm5JQ11tSud0aFcybGRJRDA5SURBcENnPT0N

SUNBZ01DQwdJQ0FnSUNBZ1kyOXVkr2x1ZFdvN0NnPT0N

SUNBZ01DQwdJQ0JwWm1oa1lWdHBYU0FoUFNBd0tRbz0N

SUNBZ01DQwdJQ0FnSUNBZ1pteGhaeUE5SUdaaGJITmxPd289DQ==

SUNBZ01DQwdJQ0JrSUNZOU1DaGpZVnRwWfNBck1DY3dKeWs3Q2c9PQ0=

SUNBZ01DQwdJQ0F2TDNCeWFXNTBaaWdpS1dRaUxHTmhXmMxkS1RzSw0=

SUNBZ01IMESN

SUNBZ01HbG1LR1pzWvdjcENnPT0N

SUNBZ01DQwdJQ0J5W1hSMWntNGdJkFpT3dvPQ0=

SUNBZ01HVnNjM1VLdQ==

SUNBZ01DQwdJQ0J5W1hSMWntNGdaRHNLdQ==

Z1FvPQ0=

Q2c9PQ0=

YzNSeWFXNW5JRzExYkNoemRISnBibWnNzUN3Z2MzUn1hVzVuSUhrCEx50Xo1TGk2NTZ5bTVZkZm1TDJQ02c9PQ0=

ZXdvPQ0=

SUNBZ01HbHVkQ0JzW1c1NE1EMGd1QzV6YVhwbEtDazdDZz09DQ==

SUNBZ01HbHVkQ0JzW1c1NU1EMGd1UzV6YVhwbEtDazdDZz09DQ==

SUNBZ01HbG1LR3hsYm5nZ1BUMGdNU2tLDQ==


```

def get_base64_diff_value(s1, s2):
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    res = 0
    for i in xrange(len(s2)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return res

def solve_stego():
    with open('ComeOn!.txt', 'rb') as f:
        file_lines = f.readlines()
        bin_str = ''
        for line in file_lines:
            steg_line = line.replace('\n', '')
            norm_line = line.replace('\n', '').decode('base64').encode('base64').replace('\n', '')
            diff = get_base64_diff_value(steg_line, norm_line)
            print diff
            pads_num = steg_line.count('=')
            if diff:
                bin_str += bin(diff)[2:].zfill(pads_num * 2)
            else:
                bin_str += '0' * pads_num * 2
            print goflag(bin_str)

def goflag(bin_str):
    res_str = ''
    for i in xrange(0, len(bin_str), 8):
        res_str += chr(int(bin_str[i:i + 8], 2))
    return res_str
print(res_str)

if __name__ == '__main__':
    solve_stego()

```

这个脚本在我的Windows下一直报错，后来发现是版本问题，是在python2的环境下进行的，由于我的Windows下没有python2，所以直接用kali，将base64文件和脚本复制到kali中，直接运行就行。需要注意的是base64文件的路径。

```

—(root@kali)-[/home/ustinian]
└─# python2 /home/ustinian/桌面/1.py
1
File "/home/ustinian/桌面/1.py", line 12
SyntaxError: Non-ASCII character '\xe6' in file /home/ustinian/桌面/1.py on line 12, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details

```

就是编码的问题还需要在py文件最前面加上#encoding:utf-8

```
#encoding:utf-8
def get_base64_diff_value(s1, s2):
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    res = 0
    for i in xrange(len(s2)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return res

def solve_stego():
    with open('/home/ustinian/桌面/come.txt','rb') as f:
        file_lines = f.readlines()
        bin_str = ''
        for line in file_lines:
            steg_line = line.replace('\n', '')
            norm_line = line.replace('\n', '').decode('base64').encode('base64').replace('\n', '')
            diff = get_base64_diff_value(steg_line, norm_line)
            print(diff)
            pads_num = steg_line.count('=')
            if diff:
                bin_str += bin(diff)[2:].zfill(pads_num * 2)
            else:
                bin_str += '0' * pads_num * 2
            print(goflag(bin_str))

def goflag(bin_str):
    res_str = ''
    for i in xrange(0, len(bin_str), 8):
        res_str += chr(int(bin_str[i:i + 8], 2))
    return res_str
    print(res_str)

if __name__ == '__main__':
```

CSDN @D2FuZ3hpYW9xaW4

问题解决，得到flag

```
└─(root@kali)-[/home/ustinian]
└─# python2 /home/ustinian/桌面/1.py
1
0
5
A
0
A
13
AC
1
AC
1
AC
0
ACT
4
ACT
1
ACT
2
ACTF
1
ACTF
3
ACTF
11
ACTF{
0
ACTF{
0
```


ACTF{
13
ACTF{
0
ACTF{
0
ACTF{
2
ACTF{6
1
ACTF{6
8
ACTF{6
1
ACTF{6a
7
ACTF{6a
0
ACTF{6a
13
ACTF{6as
0
ACTF{6as
2
ACTF{6as
1
ACTF{6as
0
ACTF{6as
1
ACTF{6ase
1
ACTF{6ase
2
ACTF{6ase
0
ACTF{6ase
0
ACTF{6ase
0
ACTF{6ase
2
ACTF{6aseb
0
ACTF{6aseb
0
ACTF{6aseb
3
ACTF{6aseb
1
ACTF{6aseb
0
ACTF{6aseb4
0
ACTF{6aseb4
1
ACTF{6aseb4
1
ACTF{6aseb4

ACTF{6aseb4_f33!

7

ACTF{6aseb4_f33!

13

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}

0

ACTF{6aseb4_f33!}


```
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}  
0  
ACTF{6aseb4_f33!}
```

最后

flag{6aseb4_f33!}