

BUUCTF-Misc-No.5

原创

水星Sur 于 2020-07-04 16:25:26 发布 1137 收藏 1

分类专栏: [BUUCTF Misc Python](#) 文章标签: [python](#) [信息安全](#)

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/pone2233/article/details/107126086>

版权



[BUUCTF 同时被 3 个专栏收录](#)

21 篇文章 2 订阅

订阅专栏



[Misc](#)

22 篇文章 0 订阅

订阅专栏



[Python](#)

5 篇文章 0 订阅

订阅专栏

文章目录

[比赛信息](#)

[内心os \(蛮重要的\)](#)

Misc:

- [\[BSidesSF2019\]zippy | SOLVED |](#)
- [\[ACTF新生赛2020\]明文攻击 | SOLVED |](#)
- [\[GKCTF2020\]Pokémon | SOLVED |](#)
- [\[GWCTF2019\]huyao | SOLVED |](#)
- [\[SCTF2019\]电单车 | SOLVED |](#)
- [\[GKCTF2020\]code obfuscation | SOLVED |](#)
- [\[GKCTF2020\]Harley Quinn | SOLVED |](#)
- [\[SUCTF2018\]followme | SOLVED |](#)
- [voip | SOLVED |](#)
- [BJD&DAS CTF 2020 | Questionnaire | SOLVED |](#)
- [BJD&DAS CTF 2020 | /bin/cat 2 | SOLVED |](#)
- [\[*CTF2019\]otaku | SOLVED |](#)
- [\[UTCTF2020\]docx | SOLVED |](#)
- [\[UTCTF2020\]basic-forensics | SOLVED |](#)
- [\[UTCTF2020\]zero | SOLVED |](#)

比赛信息

比赛地址: Buuctf靶场

内心os(蛮重要的)

对不起我就是鸽子本精Σ(っ °Д °;)っ呜呜呜

Misc:

[BSidesSF2019]zippy | SOLVED |

下载，使用**foremost**，分离得到一个压缩包，



图片已做防盗链处理
请在原文件中访问该图片

发现是带锁的



图片已做防盗链处理
请在原文件中访问该图片

，发现了**zip**的密码

```
unzip -P supercomplexpassword flag.zip
```



图片已做防盗链处理
请在原文件中访问该图片

CTF{this_flag_is_your_flag}

[ACTF新生赛2020]明文攻击 | SOLVED |

下载解压，发现一个带锁的zip和一张图片，图片放进winhex



图片已做防盗链处理
请在原文件中访问该图片

最下面发现一个zip就少了一个pk头



图片已做防盗链处理
请在原文件中访问该图片

到这一步我们了解一下明文爆破的必要条件

flag.zip

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加 解压到 测试 查看 删除 查找 向导 信息 扫描病毒 注释 自解压格式

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
flag.txt	17	17	文本文档	2019/12/2 20:22	B0C530D8

res.zip

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加 解压到 测试 查看 删除 查找 向导 信息 扫描病毒 注释 自解压格式

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
secret.txt *	19	33	文本文档	2020/1/14 13:14	483344C3
flag.txt *	17	29	文本文档	2019/12/2 20:22	B0C530D8

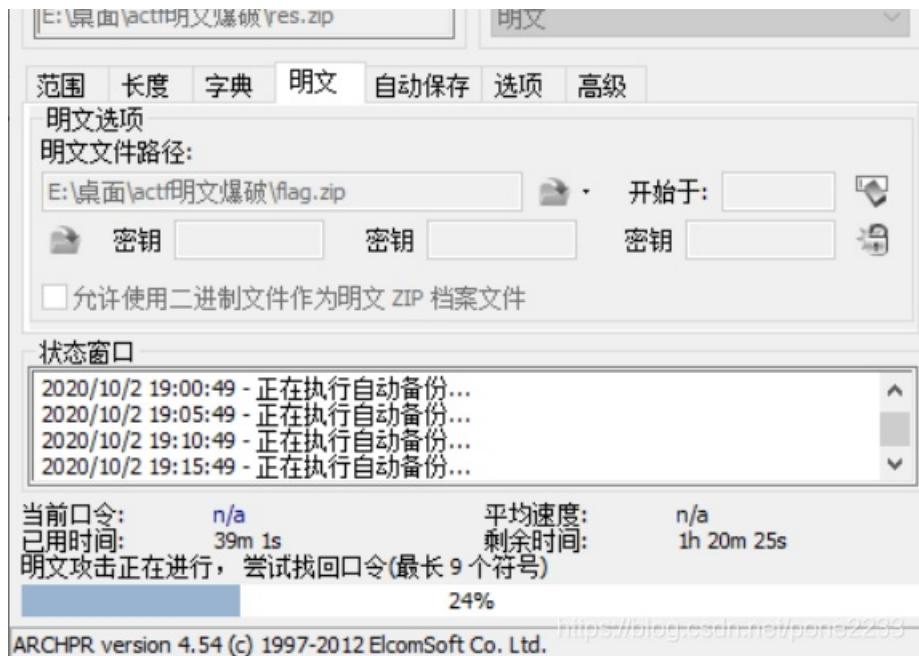
重要的还是CRC要一致，接下来我们就可以进行使用软件跑一下

ARCHPR 4.54 - 24%

文件(F) 恢复(R) 帮助(H)

打开 开始! 停止 基准测试 升级 帮助 关于 退出

加密的 ZIP/RAR/ACE/ARJ 文件 攻击类型



大概跑了30分钟，可能很短就可以了，我这时候去吃个饭回来点击停止按钮他会有一个保存页面弹出，保存一下就有无密码的zip了

名称	大小	压缩后大小	类型
secret.txt	19	21	文本文档
flag.txt	17	17	文本文档

flag就出来拉

ACTF{3te9_nbb_ahh8}

[GKCTF2020]Pokémon | SOLVED |

下载软件，在下载一个模拟器，走到103街道就能看到flag
我推荐使用金手指呦~



flag{PokEmon_14_CutE}

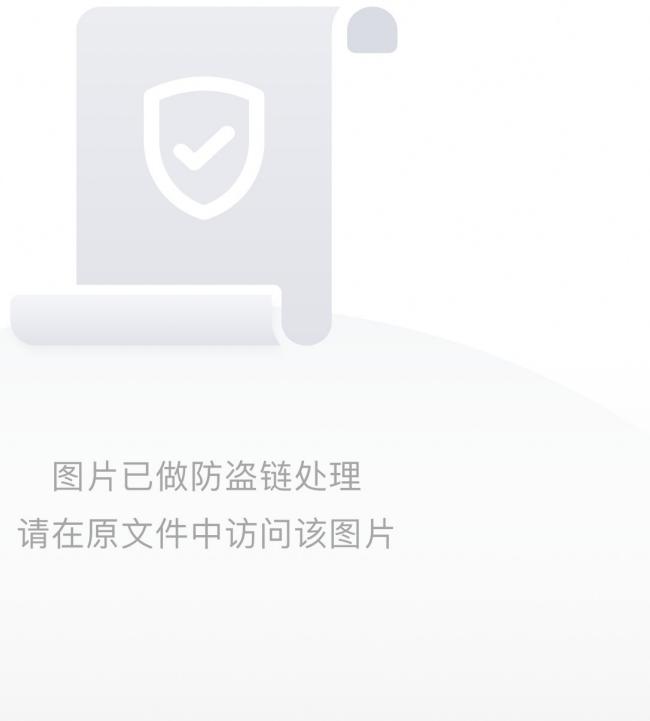
图片已做防盗链处理
请在原文件中访问该图片

[GWCTF2019]huyao | SOLVED |

一看2张png，一看就是盲水印，

```
python decode.py --original <原始图像文件> --image <图像文件> --result <结果文件>
```

然后发现了这个



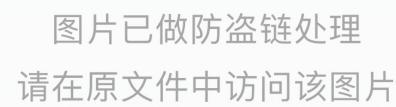
```
flag{BMW_1s_c00l}
```

[SCTF2019]电单车 | SOLVED |



图片已做防盗链处理
请在原文件中访问该图片

这个找到文献资料



图片已做防盗链处理
请在原文件中访问该图片



图片已做防盗链处理
请在原文件中访问该图片

然后flag 是地址嘛，就是中间那段

```
01110100101010100110
flag{01110100101010100110}
```

[GKCTF2020]code obfuscation | SOLVED |

首先他给我们一个二维码，我们修复一下



图片已做防盗链处理
请在原文件中访问该图片

扫描得到
base(gkctf)



图片已做防盗链处理
请在原文件中访问该图片

从二维码中分离出了rar



图片已做防盗链处理
请在原文件中访问该图片

给了一个提示，看看刚刚扫描到的。



图片已做防盗链处理
请在原文件中访问该图片

用base58就得到了rar的密码

解压下来一个png一个1文件

1文件打开一看就是js加密，我们解密一下就是这样了送上解密网站<https://tool.lu/js>



图片已做防盗链处理
请在原文件中访问该图片

然后打开png



图片已做防盗链处理
请在原文件中访问该图片

```
import string
s = "$Bn$Ai$An$Ac$Al$Au$Ad$Ae$Bk$Cc$As$At$Ad$Ai$Ao$By$Ah$Ce$Ai$An$At$Bk$Am$Aa$Ai$An$Bs$Bt$Cn$Ap$Ar$Ai$An$At$Bs$Bm$Aw$Dd$Al$Ac$Da$Am$Ae$C1$De$Ao$Cl$Dj$Ak$Ac$At$Df$Bm$Bt$Cb$Ar$Ae$At$Au$Ar$An$Bk$Da$Cb$Cp"
ll = s.split('$')
list1 = ['Bk', 'Bm', 'Bn', 'Bs', 'Bt', 'By', 'Cb', 'Cc', 'Ce', 'Cl', 'Cn', 'Cp',
'Da', 'Db', 'Dc', 'Dd', 'De', 'Df', 'Dg', 'Dh', 'Di', 'Dj']
list2 = [ ' ', '''', '#', '(', ')', '.', ',', '<', '>', '_', '{', '}', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' ]
list3 = []
list4 = []
s = string.ascii_lowercase
for i in s:
    list3.append('A%s%i')
    list4.append(i)
#print(list3, '\n', list4)

t = ''
for i in range(0, len(ll)):
    for j in range(0, len(list1)):
        if ll[i]==list1[j]:
            t += list2[j]
    for k in range(0, len(list3)):
        if ll[i]==list3[k]:
            t += list4[k]
print(t)
```

得到flag{w3lc0me_4o_9kct5}

[GKCTF2020]Harley Quinn | SOLVED |

解压文件看到一个音乐，和图片，看见他的详细



图片已做防盗链处理
请在原文件中访问该图片

知道了要听到最后面，似乎听到了按键电话的声音，然后使用之前看到大佬使用的软件点我呦，记住这个软件，最好删除其他杂音，就单独取出音频



图片已做防盗链处理
请在原文件中访问该图片

就获得了，#22283334447777338866#

用9键盘打一下



图片已做防盗链处理
请在原文件中访问该图片

ctfisfun,然后百度一下zip中的



图片已做防盗链处理
请在原文件中访问该图片



图片已做防盗链处理
请在原文件中访问该图片

最终得到了宝藏



图片已做防盗链处理
请在原文件中访问该图片

flag{Pudd1n!!_y0u_F1nd_m3!}

[SUCTF2018]followme | SOLVED |

下载文件，然后最基础到处http



图片已做防盗链处理
请在原文件中访问该图片

发现登录信息，从这一步大致猜到，是密码就是flag，可是有很多图片可以，都保存一一分析



图片已做防盗链处理
请在原文件中访问该图片

最终找不到，有用的，最后就看http，最后找到了flag



图片已做防盗链处理
请在原文件中访问该图片

SUCTF{password_is_not_weak}

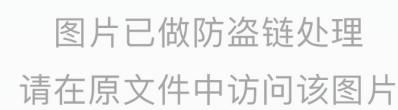
voip | SOLVED |

下载文件，百度一下，知道了这个通话数据



图片已做防盗链处理
请在原文件中访问该图片

看到RTP



图片已做防盗链处理
请在原文件中访问该图片

分析一下



图片已做防盗链处理
请在原文件中访问该图片



图片已做防盗链处理
请在原文件中访问该图片

在这一段有非常清楚的几个单词就是flag了

9001IVR
报上flag
flag{9001IVR}

BJD&DAS CTF 2020 | Questionnaire | SOLVED |

需要翻墙，打开源码一看



图片已做防盗链处理
请在原文件中访问该图片

输入正确的地名就有一小段的flag

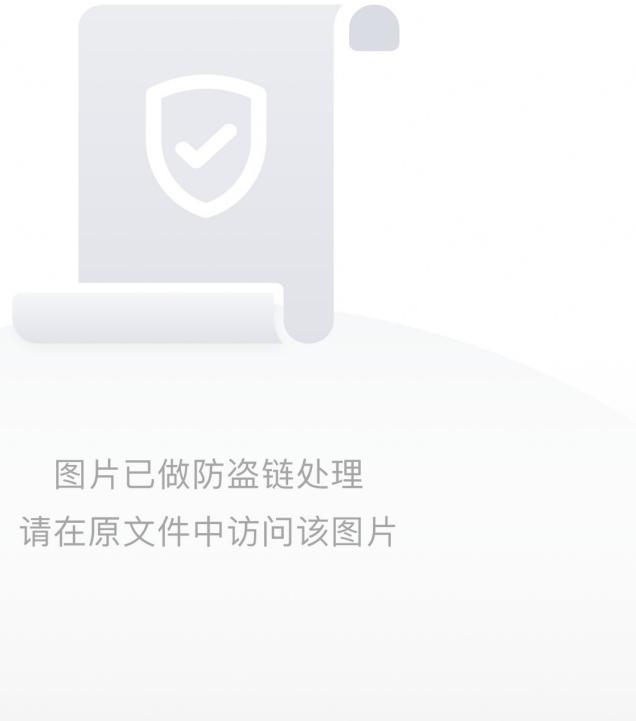


图片已做防盗链处理
请在原文件中访问该图片

d41d8cd98f00b204e9800998ecf8427e

BJD&DAS CTF 2020 | /bin/cat 2 | SOLVED |

打开图片发现颜色有些不一样，然后调一下色



扫码结果

```
m1ao~miao~mi@o~Mia0~m!a0~m1a0~~~  
md5  
9b84eb9e7107ffafebeb1000e8c05322
```

[*CTF2019]otaku | SOLVED |

解压zip，发现有一个带锁的，还是伪加密，不过我推荐是同binwalk -e 分离，拒绝一切花里胡哨。



图片已做防盗链处理
请在原文件中访问该图片

打开flag



图片已做防盗链处理
请在原文件中访问该图片

是一个加密的，看来要在word 中找到密文，这一段无论无何都无法复制，我这边使用格式刷，直接复制到txt中



图片已做防盗链处理
请在原文件中访问该图片



图片已做防盗链处理
请在原文件中访问该图片

然后就可以明文爆破了，（PS：这里有一些问题，不能用UTF-8的编码要用，GBK，就默认使用ANSI就行了。）然后就可以爆破了



图片已做防盗链处理
请在原文件中访问该图片

My_waifu

得到了zip密码了



图片已做防盗链处理
请在原文件中访问该图片

在看图软件中用LSB找到了flag，或者使用很方便的隐写工具也行这个是下载地址哦



图片已做防盗链处理
请在原文件中访问该图片

*ctf{vI0l3t_Ev3rg@RdeN}

[UTCTF2020]docx | SOLVED |

下载文件，然后docx文件里面搜不到flag，我们就直接修改他的后缀名docx改成zip



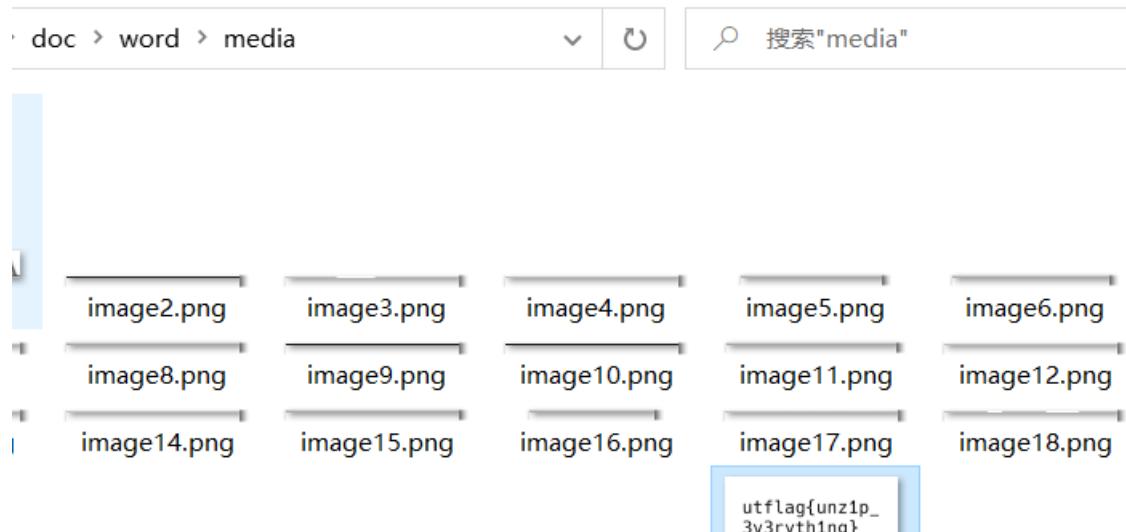
<https://blog.csdn.net/pone2233>

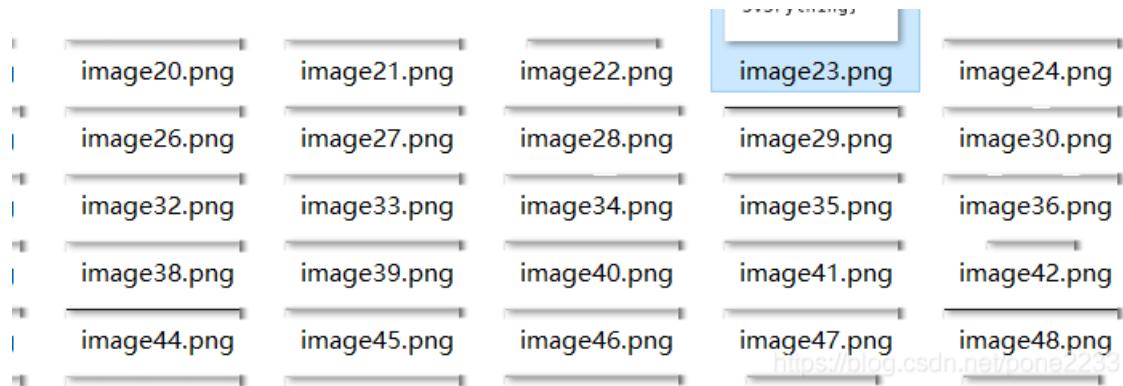
在所有文件中搜索flag



<https://blog.csdn.net/pone2233>

啥也没有，在文件夹中图片，找到了





<https://blog.csdn.net/pone2233>

utf8flag{unz1p_3v3ryth1ng}

<https://blog.csdn.net/pone2233>

flag{unz1p_3v3ryth1ng}

[UTCTF2020]basic-forensics | SOLVED |

下载，发现打不开，然后拖进winhex发现就是普通的文本，然后查找一下就发现了

```
2040 imaginary ideas and qualities to Cervantes, they show no perception of
2041 the quality that ninety-nine out of a hundred of his readers would rate
2042 highest in him, and hold to be the one that raises him above all
2043 rivalry.
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055 utf8flag{fil3_ext3nsi0ns_4r3nt_r34l}
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069 To speak of "Don Quixote" as if it were merely a humorous book would be
2070 a manifest misdescription. Cervantes at times makes it a kind of
2071 commonplace book for occasional essays and criticisms, or for the
2072 observations and reflections and gathered wisdom of a long and stirring
2073 life. It is a mine of shrewd observation on mankind and human nature.
```



<https://blog.csdn.net/pone2233>

flag{fil3_ext3nsi0ns_4r3nt_r34l}

[UTCTF2020]zero | SOLVED |

下载文件用010打开

attachment.txt

0 10 20 30 40 50 60 70 80 90 100 110

1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus quis tempus ante, nec vehicula mi. Aliquam nec nisi ut neque interdum auctor. Aliquam felis orci, vestibulum sit amet ante at, consectetur lobortis eros. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In finibus magna mauris, quis auctor libero congue quis. Duis sagittis consequat urna non tristique. Pellentesque eu lorem id quam vestibulum ultricies vel ac purus.

在打开txt对比

1.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus quis tempus ante, nec vehicula mi. Aliquam nec nisi ut neque interdum auctor. Aliquam felis orci, vestibulum sit amet ante at, consectetur lobortis eros. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In finibus magna mauris, quis auctor libero congue quis. Duis sagittis consequat urna non tristique. Pellentesque eu lorem id quam vestibulum ultricies vel ac purus.

第 1 行, 第 966 列 100% Windows (CRLF) UTF-8 https://blog.csdn.net/pone2223

起初以为是字屏统计，然后发现不是，会报错，然后看看题目zero 是0？

百度一下，找不到，去github上面看到了一个关于零的加密，零宽度字符加密然后就有flag了

Text in Text Steganography Sample

Original Text: (length: 709)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus quis tempus ante, nec vehicula mi. Aliquam nec nisi ut neque interdum auctor. Aliquam felis orci, vestibulum sit amet ante at, consectetur lobortis eros. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In finibus magna mauris, quis auctor libero congue quis. Duis sagittis consequat urna non tristique. Pellentesque eu lorem id quam vestibulum ultricies vel ac purus.

Hidden Text: (length: 32)

utf8flag{whyNOT@sc11_4927aajbqk14}

Steganography Text: (length: 965)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus quis tempus ante, nec vehicula mi. Aliquam nec nisi ut neque interdum auctor. Aliquam felis orci, vestibulum sit amet ante at, consectetur lobortis eros. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In finibus magna mauris, quis auctor libero congue quis. Duis sagittis consequat urna non tristique. Pellentesque eu lorem id quam vestibulum ultricies vel ac purus.

[Download Stego Text as File](#)

https://blog.csdn.net/pone2223

flag{whyNOT@sc11_4927aajbqk14}

完~