

## 2020CTF 解题思路

队伍: \*\*

队员: CxrjdyI \*\*\*

BH3GEI\*\*\*

### 一、something so fast

思路: 解压文件后得到一个 .gif 文件, 每一帧都是一个二维码。利用 <https://zh.bloggif.com/gif-extract?id=a2baa5235403e274622ce0848197e96f> 提供的在线逐帧分解.gif 文件服务, 得到九张二维码图片, 每张扫描后得到 flag 的一部分。顺序连接得到 flag 为:

Spirit{8c5c6150-c6bf-4473-ad6a-380016e29ced}

### 二、YLBNB

思路: 解压后得到 .png 图片文件, 使用 Windows PowerShell 的 Format-Hex ./great\_captcha.png 命令得到图片文件的十六进制。

肉眼观察 (在任一文本编辑器 ctrl+f 查找) 后可以发现 flag 在第 105 行, 上下文为:

```
104. 00000650 70 68 6F 74 6F 73 68 6F 70 3A 4C 61 79 65 72 54 photoshop:LayerT
105. 00000660 65 78 74 3D 22 53 70 69 72 69 74 7B 69 5F 31 5F ext="Spirit{i_1_
106. 00000670 6C 5F 4C 5F 49 5F 6F 5F 4F 5F 30 5F 43 5F 63 5F l_l_I_o_O_0_C_c_
107. 00000680 57 5F 77 7D 22 2F 3E 20 3C 2F 72 64 66 3A 42 61 W_w}"/> </rdf:Ba
108. 00000690 67 3E 20 3C 2F 70 68 6F 74 6F 73 68 6F 70 3A 54 g> </photoshop:T
109. 000006A0 65 78 74 4C 61 79 65 72 73 3E 20 3C 2F 72 64 66 extLayers> </rdf
110. 000006B0 3A 44 65 73 63 72 69 70 74 69 6F 6E 3E 20 3C 2F :Description> </
111. 000006C0 72 64 66 3A 52 44 46 3E 20 3C 2F 78 3A 78 6D 70 rdf:RDF> </x:xmp
```

故 flag 为:

Spirit{i\_1\_l\_l\_I\_o\_O\_0\_C\_c\_W\_w}

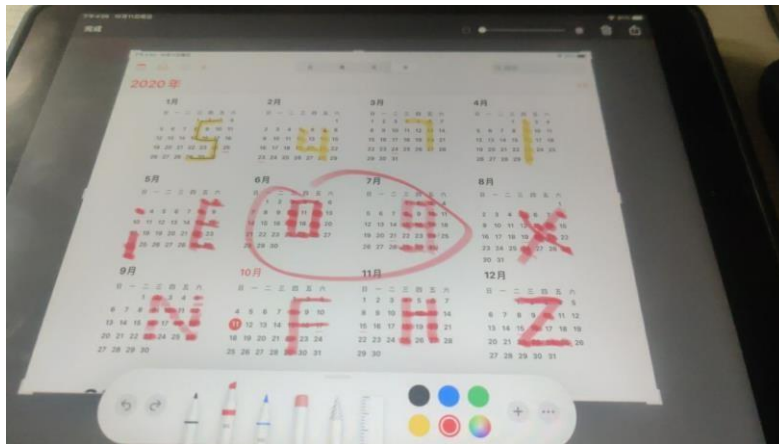
### 三、锯斤拷

思路: 在 dev c++ 内以 GBK 格式保存字符串“铸筹绋得刂峽得刂谿得涪及铸璿铸巖綽得擣紹铸巖綽得旃絨铸巖縱铸巖綽得囉峽得哋統得標結”到一个文本文件, 再用 vscode 以 utf-8 模式打开此文本文件, 即可得到” S p i r i t { P L z - u s e - u t f - 8 - p u r e l y }”, 根据题意得到 flag 为:

Spirit{P L z - u s e - u t f - 8 - p u r e l y }

#### 四、大佬的学习计划表

把每一个日期都画在日历上可以得到如下图样：



[(圈是误画，)其中第五位字符处产生了两个字母"l" "E"与数字"6"之间的歧义(将"i"拼到"E"的右边就是"6"),多次实验得到其表意为"6"] 最终得到全部字符为" 5471609XNFBHZ"，故 flag 为：

Spirit{5471609XNFBHZ}

#### 五、双重保险

利用网址 <https://www.toolnb.com/tools/pyc.html> 提供的服务对题目提供的压缩包内部的 double\_check.pyc 进行反编译得到如下代码：

```

1. def xor(a, b):
2.     return bytes([x[0] ^ x[1] for x in zip(a, b)])
3. def load_asset():
4.     return open('data', 'rb').read()
5. def check(data, key1, key2):
6.     key1 = int(key1)
7.     cipher = data[key1:key1 + 26]
8.     return xor(key2.encode(), cipher) == 'Kvbm4aeoZzR5upGgKjqPE39ovM'
9.
10. if __name__ == '__main__':
11.     data = load_asset()
12.     key1 = input('Plz input password 1:')
13.     key2 = input('Plz input password 2:')
14.     try:
15.         result = check(data, key1, key2)
16.     except Exception as e:
17.         try:
18.             result = False
19.         finally:
20.             e = None
21.         del e
22.
23.     if result:
24.         print('Correct!')
25.     else:
26.         print('Nope. Try again!')

```

解析后，利用如下代码试出 flag：

```

1. def xor(a, b):
2.     return bytes([x[0] ^ x[1] for x in zip(a, b)])
3. def load_asset():
4.     return open('data', 'rb').read()
5. def check(data, key1, key2):
6.     key1 = int(key1)
7.     cipher = data[key1:key1 + 7] # "Spirit{" 共 7 个字符
8.     return xor(key2.encode(), cipher) == 'Kvbm4ae'
9. # 返回值应为 "Spirit{" 加密的结果
10.
11. data = load_asset()
12. total = 'Kvbm4aeoZzR5upGgKjqPE39ovM'
13. crack = data[374543:374569] # 根据前七位定位到 data 二进制文件的 374543 位处
    flag 开始, 直到 374569 位, 此区间内与 flag 相同, 共 27 个字符。
14. k = 0
15. for j in total:
16.     for i in range(0, 127):
17.         if xor(i, crack[k]) == (total[k].encode()):
18.             print(chr(i))
19.             k+=1
20.             break
21. # 比对读取的二进制文件和尝试字符值 i 的值, 穷举后满足条件输出该位字符。试验出 flag 内
    所有字符后跳出循环

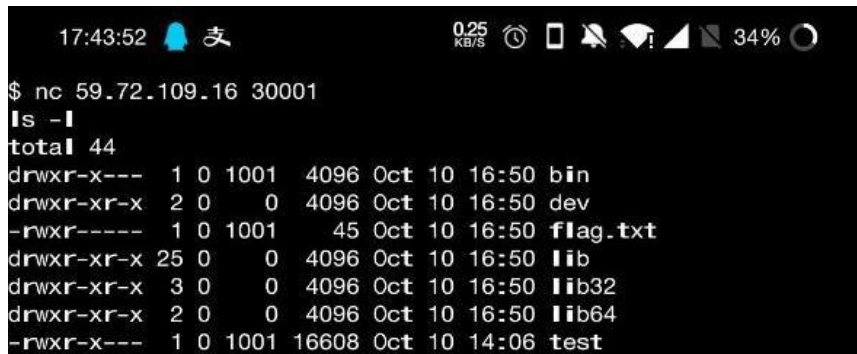
```

可得输出结果为:

Spirit{prune\_1s\_1mp0rtant}

## 六、shell

在安装了 netcat 的 linux 环境下输入 nc 59.72.109.16 30001 命令, enter 之后继续输入 ls -l, 得到当前目录下的所有文件列表:



```

17:43:52 支 0.25 KB/s 34%
$ nc 59.72.109.16 30001
ls -l
total 44
drwxr-x--- 1 0 1001 4096 Oct 10 16:50 bin
drwxr-xr-x 2 0 0 4096 Oct 10 16:50 dev
-rwxr----- 1 0 1001 45 Oct 10 16:50 flag.txt
drwxr-xr-x 25 0 0 4096 Oct 10 16:50 lib
drwxr-xr-x 3 0 0 4096 Oct 10 16:50 lib32
drwxr-xr-x 2 0 0 4096 Oct 10 16:50 lib64
-rwxr-x--- 1 0 1001 16608 Oct 10 14:06 test

```

继续使用 cat 命令得到 flag:

```

cat flag.txt
Spirit{045cca15-3035-46d4-9f61-4ecec0d255b}

```

七、签到题：凯撒密码，一种简单的替换密码。使用 Android app “Caesar cipher” 或其他 web 解密服务可直接得到 25（因为只有 26 个英文字母，故只有 25 个方案）个不同的结果：

根据字符串片段"spirit"可知 flag 为：  
spirit{sign\_in\_with\_caesar?}

SHIFTS	
Key: 1	Llskbn{bzg_bg_gbma_vtkk?}
Key: 2	Kkajel{kayf_aj_oaiz_uwkg?}
Key: 3	Jgata{jase_ah_naky_trjhl?}
Key: 4	Ilytjlljywd_yd_mylx_squqh?}
Key: 5	Haxge{hxvc_xc_kiiv_rplhg?}
Key: 6	Gdewehlgwub_wb_kwiv_qusgd?}
Key: 7	Fcevegltz_vz_jvgu_pnlhe?}
Key: 8	Edadu{eusz_uz_lufl_onqemd?}
Key: 9	Datcie{dry_ty_hoes_nodic?}
Key: 10	Czsod{csox_sx_gsd_rmkock?}
Key: 11	Bytac{brow_rv_hoa_lnhla?}
Key: 12	Avazab{aqv_ov_nghp_kimalz?}
Key: 13	Zepypa{pna_pu_dpa_jhahy?}
Key: 14	Yvazae{pant_at_paan_kjkyax?}
Key: 15	Xurwery{xnls_ns_brym_hjfw?}
Key: 16	Wtmvma{emkr_mr_amad_gelwew?}
Key: 17	Vkuwvljq_lq_zwkl_tdrvdu?}
Key: 18	Ukrtkv{klip_lip_ykvi_asqucl?}
Key: 19	Tqjau{tpe_jo_xua_dhttas?}
Key: 20	Spirit{sign_in_with_caesar?}
Key: 21	Rnqahsjtllm_hm_vhag_badrza?}
Key: 22	Onpogr{age_gl_uqrl_aycqpp?}
Key: 23	Pmkof{pldk_lk_tqez_zxbpxo?}
Key: 24	Omeo{ped_sl_sepd_ywaown?}
Key: 25	Nldmdo{ndbi_sl_doc_xvznm?}

## 八、小猪佩奇

根据猪圈密码：

A	B	C	J	K	L
D	E	F	M	N	O
G	H	I	P	Q	R
<del>S</del>	<del>T</del>	<del>U</del>	<del>W</del>	<del>X</del>	<del>Y</del>
<del>V</del>			<del>Z</del>		

并把 flag 图样中的点状图换为与点数相等的数字，可得 flag 为：

Spirit{FFF9A7C5577A}

## 全部 flag：

Spirit{8c5c6150-c6bf-4473-ad6a-380016e29ced}

Spirit{i\_1\_l\_l\_o\_o\_0\_c\_c\_w\_w}

Spirit{P Lz - u s e - u t f - 8 - p u r e l y}

Spirit{5471609XNFHZ}

Spirit{prune\_1s\_1mp0rtant}

Spirit {045cca15-3035-46d4 9f61-4ecec0d255b}

Spirit{sign\_in\_with\_caesar?}

Spirit{FFF9A7C5577A}