Web

谁吃了我的flag

这个作为 web 第一题确实有点吓人。 直接上来是一个 html 文件，给了一半的 flag。因为是 html，又没有 JavaScript，所以搞不了什么小动作。 试了下`.git`、`.DS_Store`都不行。 那么问题来了，剩下的 flag 在哪呢？ 根据题干`是因为昨天没有好好关机吗T_T`，自己原来剪视频时，总是会遇到 Premiere 崩掉的情况，此时会产生一个临时文件。 百度了一下`ctf 临时文件`，发现 vim 编辑器是会生成临时文件的，文件名为`.文件名.swp`， 尝试访问`.index.html.swp`，下载下来后打开即可得到完整的 flag：

```
hgame{3eek_diScl0Sure_fRom+wEbsit@}
```

换头大作战

头就是请求头咯~ 显示`POST`一个`want`，之后根据提示来一步步加请求头。 还是推荐使用 Chrome 插件 Restlet Client。 `X-Forwarded-For`为`127.0.0.1`，`Referer`为`www.bilibili.com`，`Cookie`为`admin`。 但是，**这里的User-Agent好像有点问题**，当时就是卡在这里了，不知道是题目的问题还是我太菜了。 直接设置`User-Agent`为`Waterfox/50.0`是不行的，最后是改为`use Waterfox/50.0`才可以，就很懵逼。

```
hgame{hTTp_HeaDeR_iS_Ez}
```

very easy web

简单的代码审计。 代码中把`GET`参数`id`进行了`urldecode`，而浏览器同样也会帮我们先做一遍`urldecode`，因此我们要传`urlencode`两遍的`vidar`。 那么问题来了，PHP里面`echo(urlencode('vidar'));`出来的还是 vidar。 好像英文字母这些`urldecode`都是本身啊；这里就需要我们手动自己对照 URL 编码的表格得出来。 可以参考这个http://www.w3school.com.cn/tags

/html_ref_urlencode.html 两遍 URL 编码后得出`%2576%2569%2564%2561%2572`，访问`http://120.78.184.111:8080/week1/very_ez/index.php?id=%2576%2569%2564%2561%2572`拿到 flag：

```
hgame{urlDecode_Is_GoOd}
```

can u find me?

进入后日常 F12，转到`f12.php`。 说是要`POST` password，日常瞄一眼响应头，找到`password: woyaoflag`， 然后`POST`过去就行。 转到`http://47.107.252.171:8080/toofast.php`，说我们速度 too fast，那就是要抓包咯。 Charles 走一波，得到 flag：

```
hgame{f12_1s_aMazIng111}
```

> 至此，web 题就 ak 了！花了半个小时，如果中途键盘没有因为虚拟机而崩掉搞不好还能更快 23333

RE

一点都不会的菜鸡，只能挑简单的做。

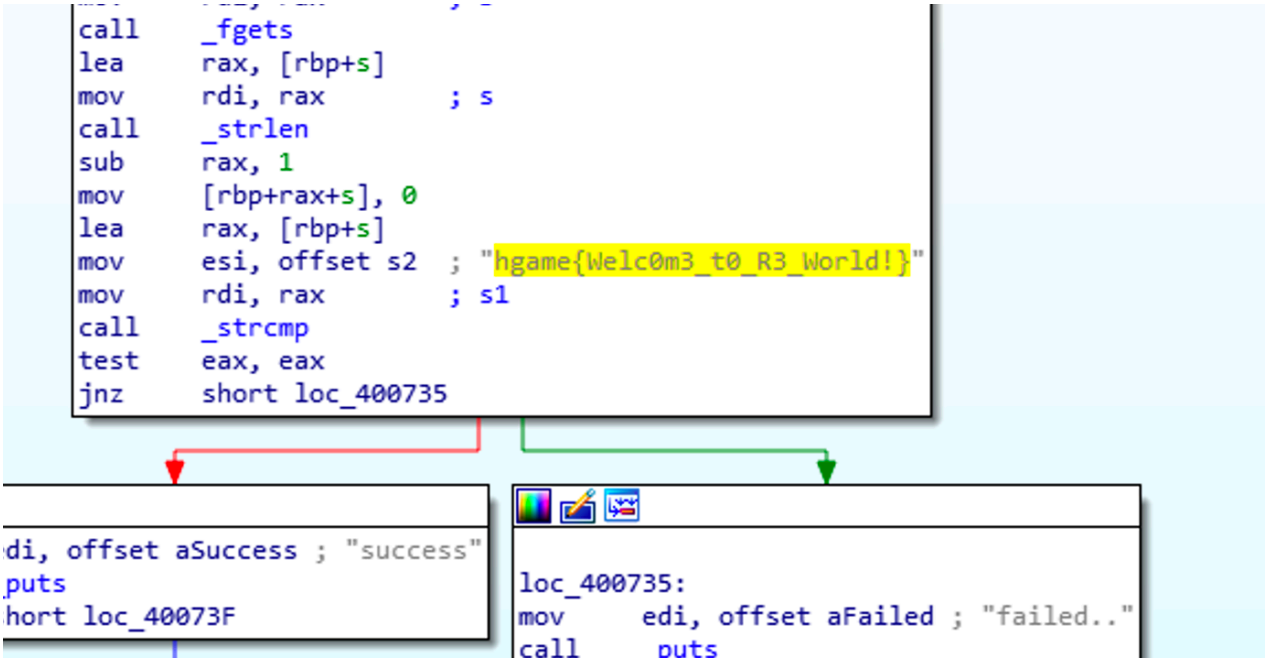
brainfxxker

挺硬核的一道题。 brainfucker 看起来虽然比较扯，但是正如 wiki 上所说，“Brainfuck程序很难读懂。尽管如此，Brainfuck图灵机一样可以完成任何计算任务。”所以……这就十分底层了。 菜鸡的我只能大概搞得懂一点，但是，**这并不妨碍我们自己先随便试试。** 第一个字母输入个`a`，不对，再猜`b`，诶！对了！ 然后就是脑洞了：总共有要输入 9 个字符，题目是 brainfuck 也是 9 个字符，那 flag 会不会就是这个？ 结果还真是，只是稍微换了下字母混淆了一下，比如改下大小写啊，`i`换成`!`，都是些常见的套路。 然后，我们就可以很轻松的猜出flag，笑死。

```
hgame{bR4!NfUcK}
```

HelloRe

下载下来直接拖进 IDA，然后一个个翻找到 hgame 的 flag 就行。




```
hgame{Welc0m3_t0_R3_World!}
```

Pro的Python教室(一)

跟 Python 无关，直接把`enc2 = 'SGVyZV8xc18zYXN5Xw=='` `base64_decode`一下，然后拼成最后的 flag 就行。 这题只是给个示例而已，传达一种思想。要来真的肯定是那种复杂的一匹的加密算法。

```
hgame{Here_1s_3asy_Pyth0n}
```

PWN

还是一点都不会，只会最简单的。

aaaaaaaaaa

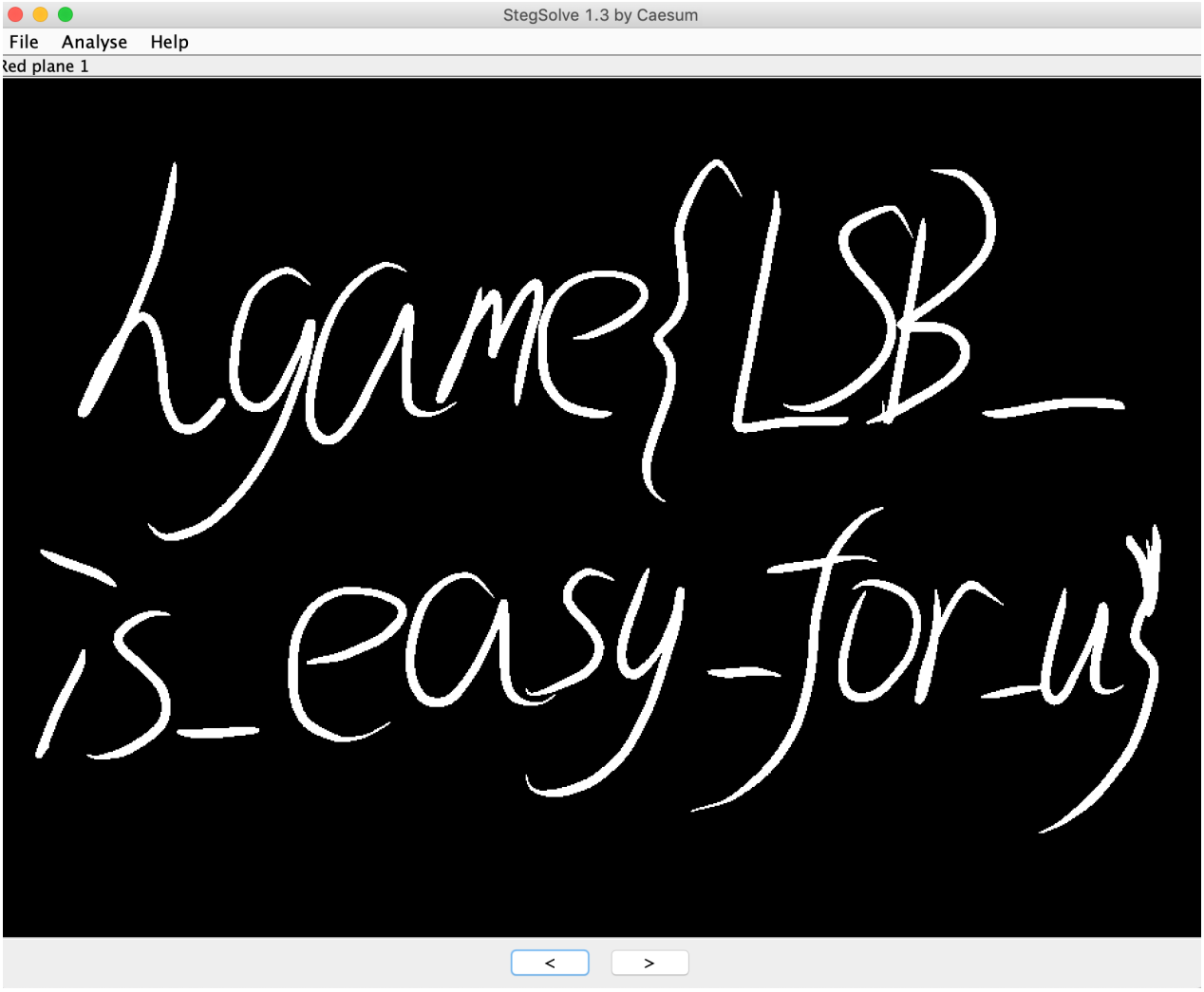
`nc`连上后，疯狂输入`a`，然后就会反弹出 Shell，然后`cat flag`即可。 感觉背后的原理就是接受输入的函数不安全，然后就溢出了吧。

```
hgame{Aa4_4aA_4a4aAAA}
```

MISC

Hidden Image in LSB

唔……这个是花了四五个小时，真的是难受。 首先是疯狂在 GitHub 上 LSB 的项目，clone 下来后又直接在本地装了一堆依赖，把本地的 Python 环境搞得乱七八糟。 然后发现还是不行。 也在网上找到了 stegsolve 这个软件，奈何网上的使用教程是错的，我一直在 Data Extract 里面疯狂尝试，结果什么也出不来。 最后发现…… 只要多点击几下下方的箭头就可以了，被自己蠢死QAQ

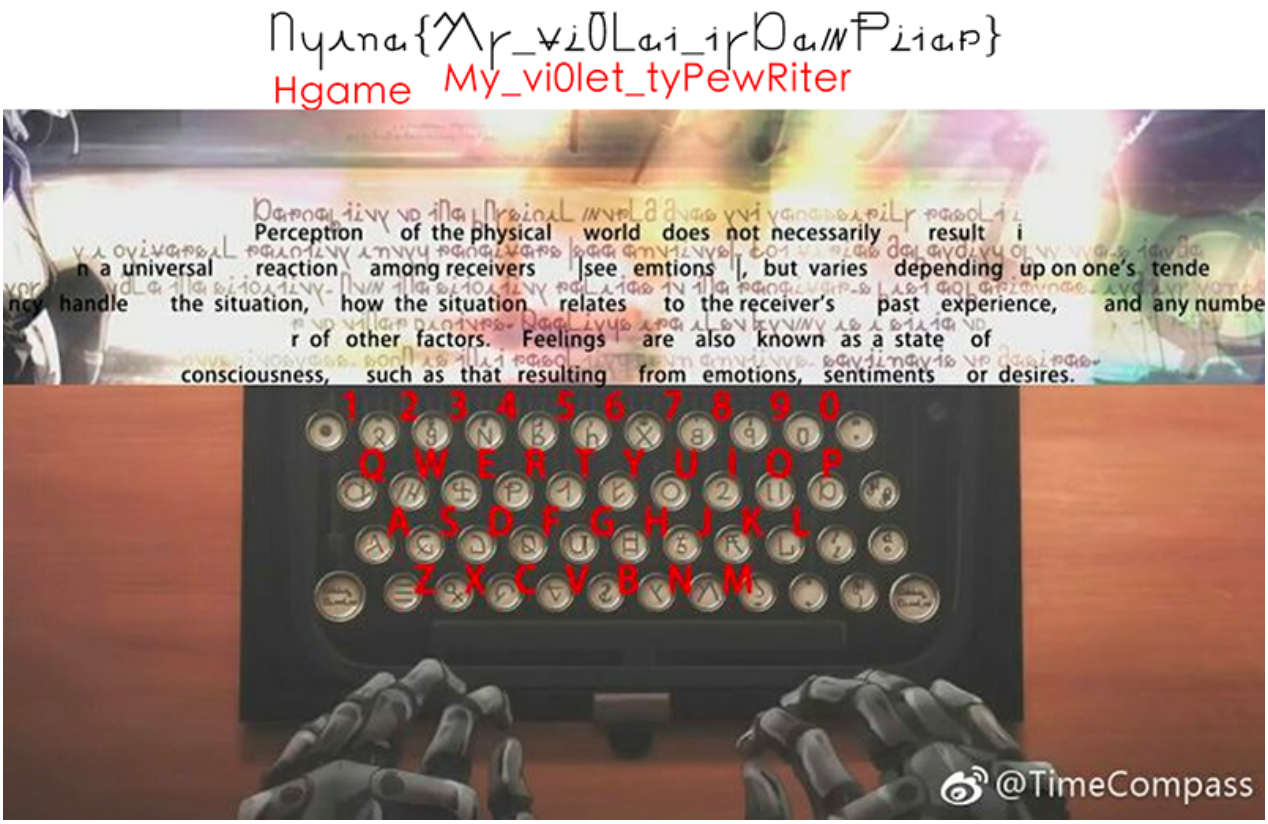



```
hgame{LSB_is_easy_for_u}
```

打字机

死宅真恶心！死宅真恶心！死宅真恶心！ 明显就是欺负我看番看得少23333

把打字机的图放 Google 搜了一下，发现居然是《紫罗兰永恒花园》里面的。 瞬间气死，我还以为是什么上古文字。 因为打字机图上的字母都是大写的字母，小写的字母并没有。 在 Google 上我找到了网友对动画中一封信的翻译，其中的大小写字母对照的挺多的。 根据这张图，然后连蒙带猜出 flag：




```
hgame{My_vi0let_tyPewRiter}
```

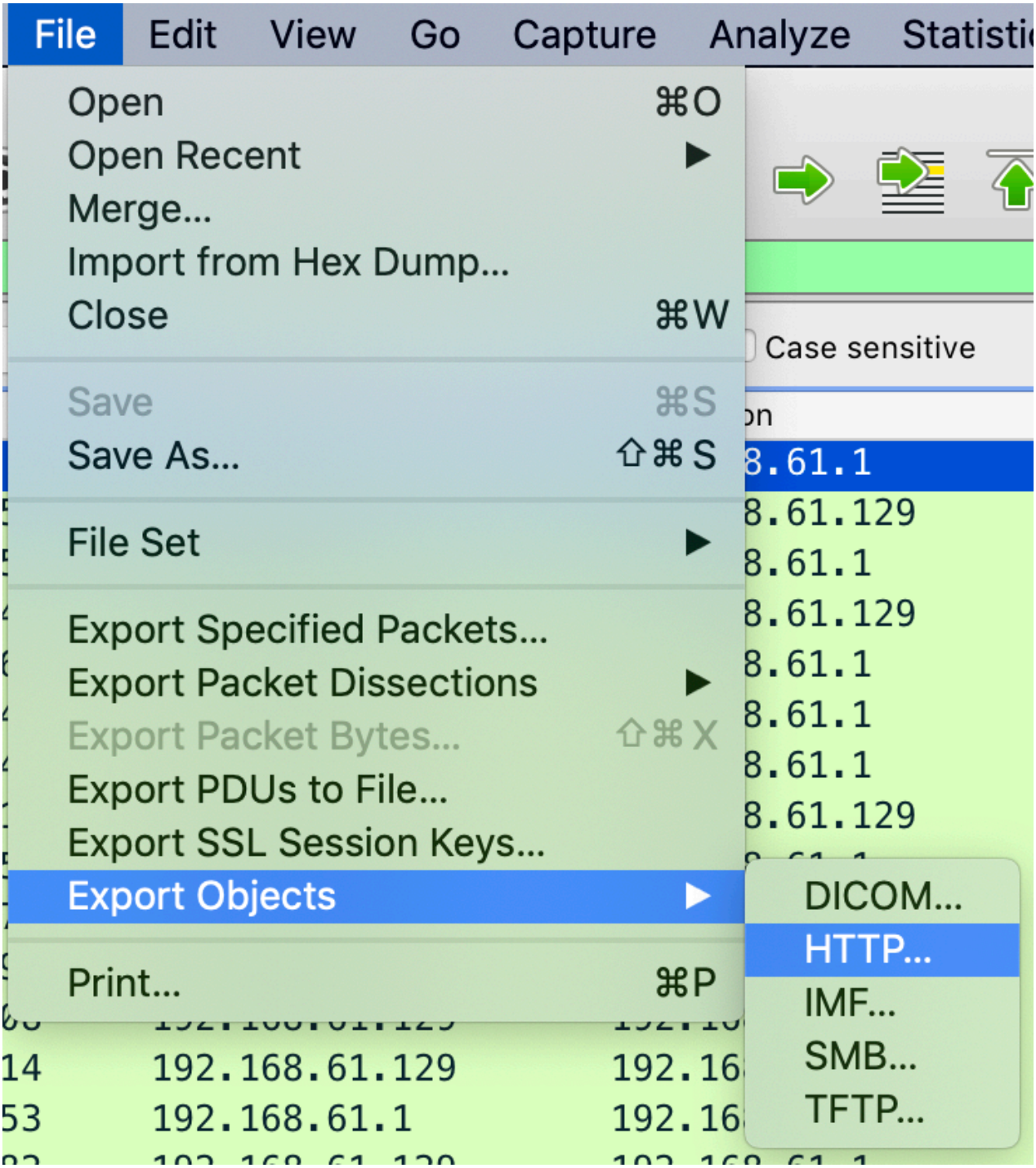
Broken Chest

这个文件下载下来后，丢进 binwalk 看了下，发现有点东西。 但是却一直解压不出来，提示我的 Python 缺少库，然而 macOS 上的 brew 好像并不能装这个库。 便开始忙着用 Docker 开 Ubuntu 虚拟机，配环境。 最后居然。。。 因为这是个 zip 文件，然后文件头那里缺了一点点，因此就损坏了。 自己做一个 zip 文件，比对一下文件头，把缺少的部分用 WinHex 加上去即可解压，密码在压缩包简介那里给出，得到 flag：

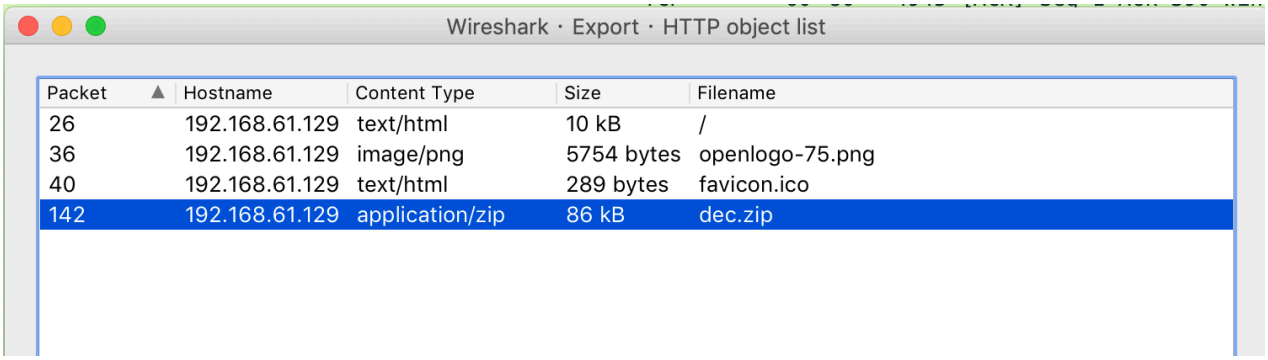
```
hgame{Cra2y_D1aM0nd}
```

Try

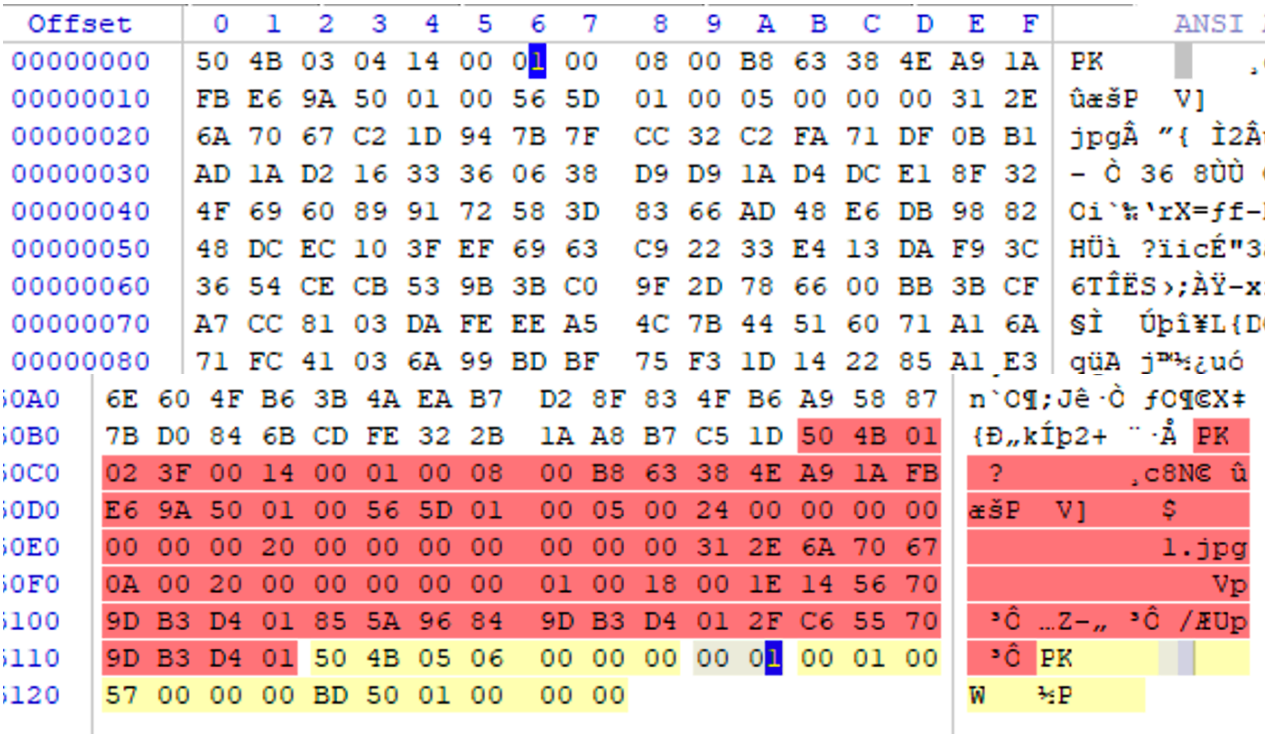
这个题做了很久也是自己太菜了。 首先把下载下来的 try-it.pcapng 拖进 Wireshark 分析。 只看 HTTP 请求，我们可以看到有一个 dec.zip 的文件。然后我便傻乎乎的把整个请求给保存下来然后用 WinHex 删掉前面的请求头，发现并不能打开。 正确的方法应该是：File -> Export Objects -> HTTP



然后选择我们的文件就好了！

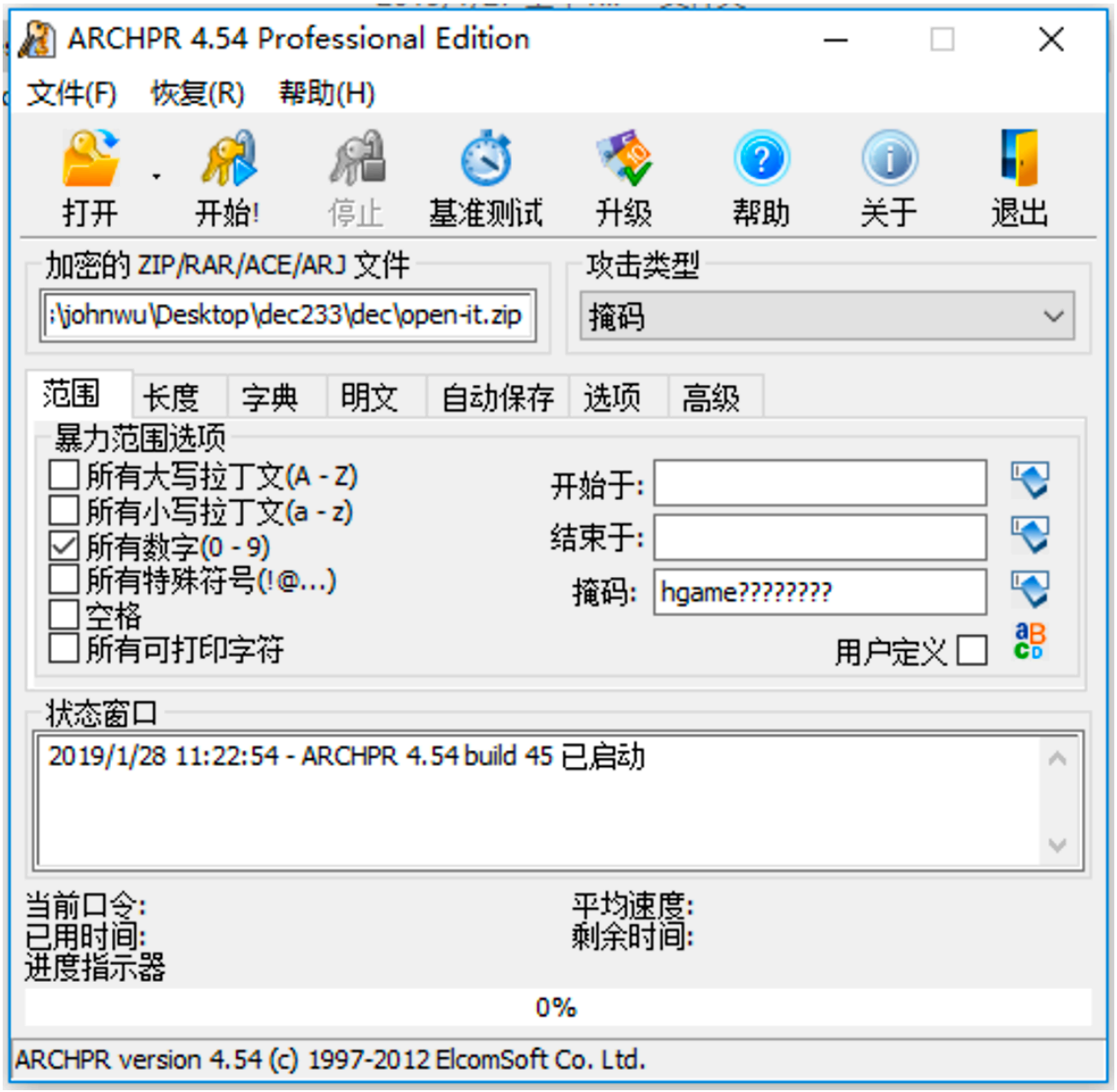


解压后发现得到了一个 password.txt 文件和一个带密码的 open-it.zip 文件。 用 WinHex 或 HexFriend 打开 open-it.zip 检查一下是否是伪加密，



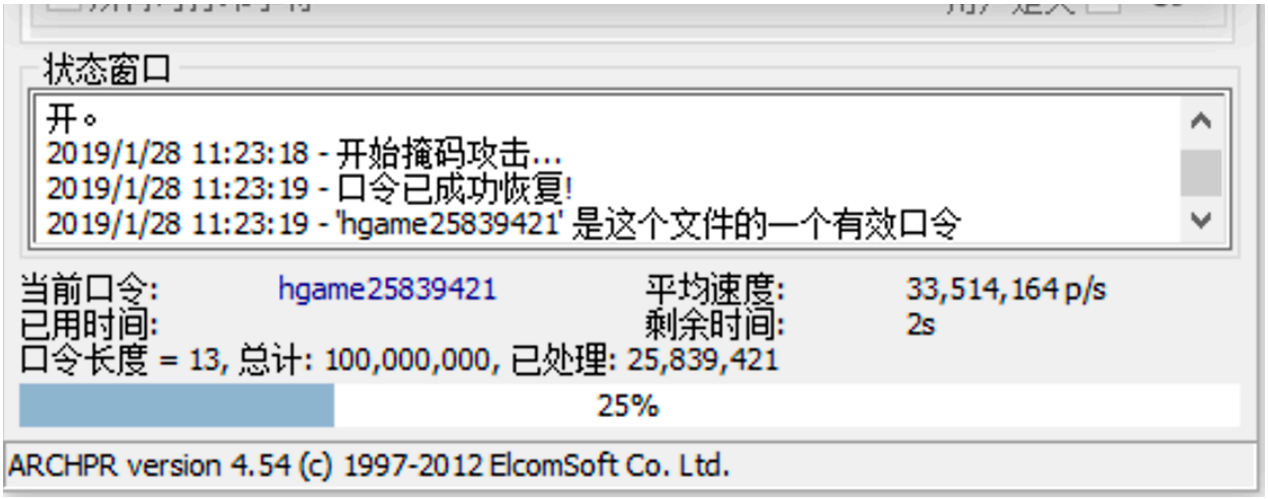
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI
00000000	50	4B	03	04	14	00	01	00	08	00	B8	63	38	4E	A9	1A	PK ¸c8N©
00000010	FB	E6	9A	50	01	00	56	5D	01	00	05	00	00	00	31	2E	ûæšP V] 1.
00000020	6A	70	67	C2	1D	94	7B	7F	CC	32	C2	FA	71	DF	0B	B1	jpgÂ ”{ Ì2ÂúqßB±
00000030	AD	1A	D2	16	33	36	06	38	D9	D9	1A	D4	DC	E1	8F	32	- Ò 36 8ÙÙ ÔÜá 2
00000040	4F	69	60	89	91	72	58	3D	83	66	AD	48	E6	DB	98	82	Oi`‰‘rX=ƒf-HæÛ˜‚
00000050	48	DC	EC	10	3F	EF	69	63	C9	22	33	E4	13	DA	F9	3C	HÜì ?ïicÉ"3ä Úù<
00000060	36	54	CE	CB	53	9B	3B	C0	9F	2D	78	66	00	BB	3B	CF	6TÎËS›;À-xf »;Ï
00000070	A7	CC	81	03	DA	FE	EE	A5	4C	7B	44	51	60	71	A1	6A	§Ì Úþî¥L{DQ`q¡j
00000080	71	FC	41	03	6A	99	BD	BF	75	F3	1D	14	22	85	A1	E3	qüA j™½¿uó " ¡ã
00A0	6E	60	4F	B6	3B	4A	EA	B7	D2	8F	83	4F	B6	A9	58	87	n`O¶;Jê·Ò ƒO¶©X‡
00B0	7B	D0	84	6B	CD	FE	32	2B	1A	A8	B7	C5	1D	50	4B	01	{Ð„kÍþ2+ ¨·Å PK
00C0	02	3F	00	14	00	01	00	08	00	B8	63	38	4E	A9	1A	FB	? ¸c8N© û
00D0	E6	9A	50	01	00	56	5D	01	00	05	00	24	00	00	00	00	æšP V] $
00E0	00	00	00	20	00	00	00	00	00	00	00	31	2E	6A	70	67	1.jpg
00F0	0A	00	20	00	00	00	00	00	01	00	18	00	1E	14	56	70	Vp
0100	9D	B3	D4	01	85	5A	96	84	9D	B3	D4	01	2F	C6	55	70	³Ô …Z–„ ³Ô /ÆUp
0110	9D	B3	D4	01	50	4B	05	06	00	00	00	00	01	00	01	00	³Ô PK
0120	57	00	00	00	BD	50	01	00	00	00							W ½P

可以看到开头结尾都有这个`0001`，说明这是用 7z 压缩的真的带密码加密的压缩包。 通过 password.txt 的提示我们可以知道，压缩包的密码是`hgame`后面接 8 位字符，用 ARCHPR 进行暴力破解。 选择“掩码”，然后输入`hgame????????`，首先只跑一遍所有数字尝试一下。


ARCHPR 4.54 Professional Edition
文件(F) 恢复(R) 帮助(H)
打开 开始! 停止 基准测试 升级 帮助 关于 退出
加密的 ZIP/RAR/ACE/ARJ 文件
:\johnwu\Desktop\dec233\dec\open-it.zip
攻击类型
掩码
范围 长度 字典 明文 自动保存 选项 高级
暴力范围选项
所有大写拉丁文(A - Z)
所有小写拉丁文(a - z)
所有数字(0 - 9)
所有特殊符号(!@...)
空格
所有可打印字符
开始于:
结束于:
掩码: hgame????????
用户定义
状态窗口
2019/1/28 11:22:54 - ARCHPR 4.54 build 45 已启动
当前口令:
已用时间:
进度指示器
平均速度:
剩余时间:
0%
ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.

不到一秒钟就出来啦！


ARCHPR 4.54 - 25%
文件(F) 恢复(R) 帮助(H)
口令已成功恢复!
Advanced Archive Password Recovery 统计信息:
总计口令 25,839,421
总计时间 773ms
平均速度(口令/秒) 33,427,452
这个文件的口令 hgame25839421
十六进制口令 68 67 61 6d 65 32 35 38 33 39 34 32 31
保存...
确定



输入密码解压后，可以看到一张**石原里美老婆**的照片。 用 binwalk 看一下，发现里面藏着个 1.docx 的 Word 文档。

```
johnwu@JohnMBP ~/Downloads binwalk 1.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
566           0x236           LZMA compressed data, properties: 0xD0, dictionary size: 10747904 bytes, uncompressed size: 274877906944 bytes
79837         0x137DD         Zip archive data, at least v2.0 to extract, compressed size: 9447, uncompressed size: 12178, name: 1.docx
89408         0x15D40         End of Zip archive, footer length: 22
```

提取出来后用 Word 打开，发现是空白的，但是上面很明显有内容。 直接全选复制，然后粘贴到随便一个文本编辑器，便得到了 flag：

```
hgame{59d28413e36019861498e823f3f41406}
```

> 好耶！misc 也 ak 了！

CRYPTO

Mix

这道是 AC 学长出的题。因为一些原因，这道题很容易得到假 flag 。 当时我疯狂怀疑人生。在群里 oyeye 给了提示后才发现不对劲2333 描述里面一下就是摩斯密码，随便在网上找个在线解码，得到：

```
744B735F6D6F7944716B7B6251663430657D
```

然后 base 编码全部试一下，发现是 base16：

```
tKs_moyDqk{bQf40e}
```

看过去年 HGame 的 Writeup，了解到这是栅栏密码。根据 flag 的格式，最后解出来的应该为 `xxxxx{xxxxxxxxxxx}` 。 推荐这个网站 `https://www.qqxiuzi.cn/bianma/zhalanmima.php` 当每组字数为 9 时，解出来为：

```
tsmyq{Q4eK_oDkbf0}
```

符合题意。这一看就是凯撒密码，然后我就一脚踩进坑里面了2333 我是用 CTF 在线工具 这个站 `http://ctf.ssleye.com/` 里面的凯撒密码来解密的。 这个站的问题就在于，凯撒编码后所有字母全部都变成了小写：

```
hgame{e4sy_crypt0}
```

然而我们的密码原文中是有大写字母的。所以我需要手动对比一下，把小写换成大写才是最终的 flag：

```
hgame{E4sY_cRypt0}
```

狡猾狡猾~

Base全家

很有意思的一道题，正规解法应该是用 Python 写脚本的。 但是因为我是个菜鸡，所以用 Python 手动解决。期间总共做了四遍！ 我们知道，题目里给的那一串就是疯狂各种 Base 编码搞出来的。 我们可以用 Python `base64` 库中的 `base64.b16decode``base64.b32decode``base64.b64decode` ，来对其分别进行相应的解码。 如果用错的解码的方式，Python 会报错并抛出一个异常。 然后我可以**用try语句**或者用**你的眼睛**捕获到异常然后再换一种解码方式就行。 我只有菜鸡的做法，也就是我的方法。用我的眼睛捕获异常2333，再手动更换解码方式。

```
print(base64.b32decode(base64.b64decode(base64.b64decode(base64.b64decode(ba
se64.b32decode(base64.b16decode(base64.b16decode(base64.b16decode(base64.b16
decode(base64.b64decode(base64.b16decode(base64.b64decode(base64.b32decode(b
ase64.b16decode(base64.b32decode(base64.b16decode(base64.b16decode(base64.b1
6decode(base64.b64decode(base64.b64decode(a))))))))))))))))))))))
```

还是太菜了..... 最后得到字符串

```
base58 : 2BAja2VqXoHi9Lo5kfQZBPjq1EmZHGEudM5JyDPREPmS3CxrpB8BnC
```

在网上找到了一个 base58 解码的 JavaScript 实现：`https://blog.csdn.net/github_39132847/article/details/83624463` 解码后得到 flag：

```
hgame{40ca78cde14458da697066eb4cc7daf6}
```

> 嗯，然后其实还有些题感觉可以做的。只是回到老家后事挺多的，便咕咕咕了。 ak 了 web 和 misc，也算是及格了23333