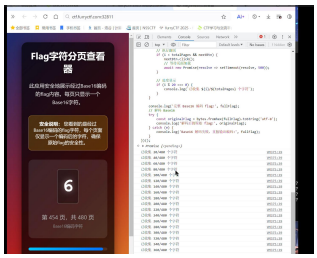
# FlagReader

1.脚本收集 480 个字符（或者一页页翻 hah




2.原字符串：

3636373537323732373934333534343637423332333136353633333433323632363632444363433
3933323331324433343632333833313244333936323635333232443633333433313336333036333
3336333836333332363336333244333633313333333333373632333536353244333836343335333
3332443334363536333635324436323331333533353244333533373338363136363332333636353
3335363433333634324436343633336333632333836343635333232443332363336323339324433
3433353631333432443339333033363631324433373632333636323635333436363633363236363
366236363744

3.Base16 解码 2 次：

furryCTF{21ec42bf-d921-4b81-9be2-c4160c68c2cc-61337b5e-8d53-4ece-b155-578af26e5d3d-dccb8de2-2cb9-45a4-906a-7b6be4fcbfbf}

# PyEditor

1.题目提供了 app.py 源码，其中有一段被注释但未删除的关键代码：

# Hey bro, don't forget to remove this before release!!!
import os
import sys
flag_content = os.environ.get('GZCTF_FLAG', '') os.environ['GZCTF_FLAG'] = '' try: with open('/flag.txt', 'w') as f: f.write(flag_content)
except: pass

2. 沙箱通过 ast.parse 解析代码，拦截了部分函数模块等。

3. 尝试：

1）在 safe_exec() 中读取 /flag.txt（safe_exec() 执行时，写入 /flag.txt 的代码还未运行，导致 No such file or directory）

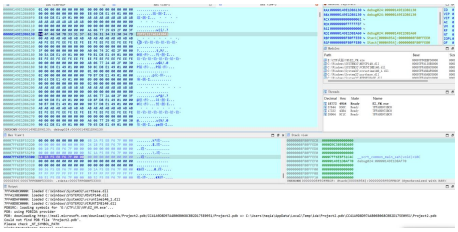
2）动态导入与循环等待，循环阻塞导致 safe_exec() 无法结束，写入 /flag.txt 的代码永远不会执行

直接读取环境变量，a = __import__ os = a('os') print(os.environ.get('GZCTF_FLAG', ''))绕过沙箱检测，直接读取到环境变量中的 flag。

# ezvm

1. 本来准备偷懒直接打开记事本发现有 POFP{327a6c4304}~
直到我交上去。。。(谢谢


2. 所以开头的一定是个幌子。按照一般流程脱壳（没有）->idax64 里打开找了一遍没找到 flag 有关的字符串（除了上面那串）
3. 我打开 main 函数。。。啊，虚拟机。从指令集 v26 动态生成正确的 flag，存储在 v5 内存中。所以要去找 v5。
4. 在汇编窗口找到 call operator new（分配堆内存）后的 mov rbx, rax，确认这行就是 v5 = v3 的汇编实现，rbx 对应 v5、rax 是堆内存返回值；
5. 定位到程序的逐字节验证汇编逻辑（loc_7FF6EBF512F0），在 IDA 中对该地址下断点，运行程序并输入任意字符（如 a）触发断点
6. 断点命中后，在 IDA 寄存器面板直接读取到 rax = 0x00000149E1D86130，确认该值就是 v5 的运行时堆内存地址（验证逻辑中 rax 始终指向 v5）
7. 跳转到 v5 的内存地址后，查看 IDAHex dump 窗口的十六进制 + ASCII 双列显示，直接读取到内存中存储的 ASCII 字符串为 POFP{317a614304}



# 赛后问卷

填就行了

# ezmd5

分析：这是一道 PHP 弱类型与数组绕过 MD5 比较 题目。题目要求同时满足两个矛盾的条件**$user !== $pass：要求两个变量的值不完全相等（类型和值都要不同）。**

**md5($user) === md5($pass)：要求两个变量的 MD5 哈希值完全相等（类型和值都要相同**

思路：（试）采用 “数组绕过 MD5 函数” 。利用 PHP 的 md5() 函数在处理数组（Array） 时，不会报错，而是会返回 NULL。

当 $user 和 $pass 都是数组时：

md5($user) 结果为 NULL

md5($pass) 结果为 NULL

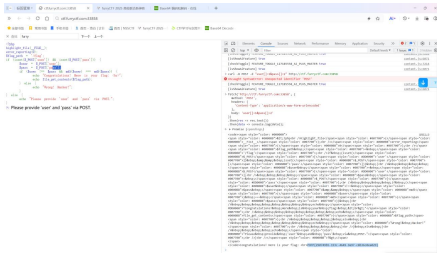
此时 md5($user) === md5($pass) 成立（NULL === NULL）。同时，两个不同的数组（或相同数组在严格比较下）满足 $user !== $pass

我构造 body: 'user[]=1&pass[]=2'

user[]=1：将 user 构造成一个包含元素 1 的数组

pass[]=2：将 pass 构造成一个包含元素 2 的数组

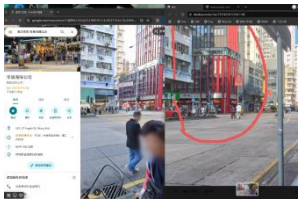
结果：绕过了服务器的校验逻辑，成功触发了 echo file_get_contents($flag_path); 代码，读取到了 flag。



# 独游

1. 提取图中有效信息：DDKing、红色地标建筑、红色临时巴士站牌，线路为 281E、287D、287X、最最重要的是左边绿色招牌的袁記雲餃。字体大部分是繁体字，上网搜发现是香港那边的。锁定香港。
2. 去大众点评上找，就找到了一模一样的店

3. 位置基本锁定，上谷歌地球



4. 得出 flag

Ps:其实 CCPreview 做出来了。。不过已经是赛后了