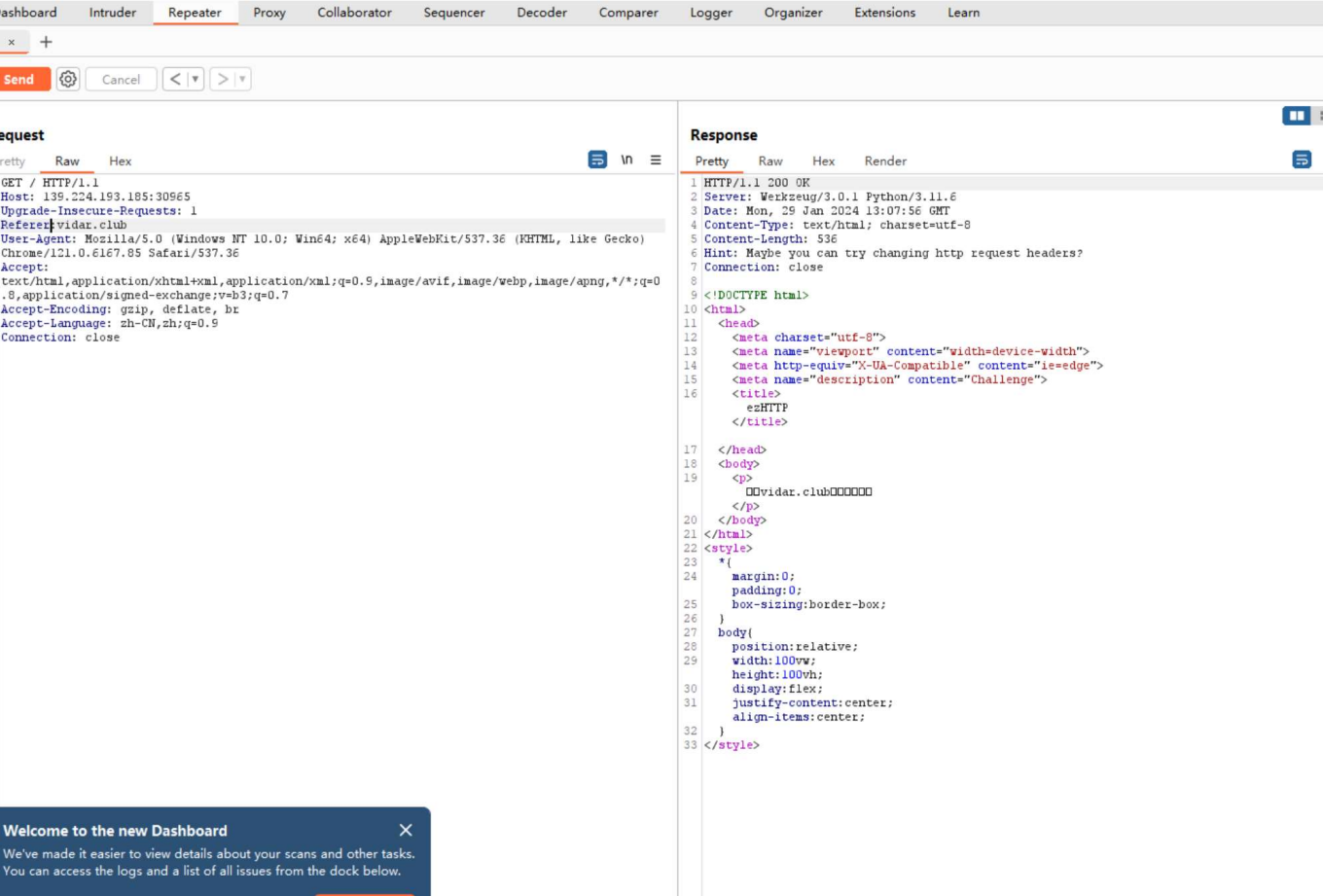


# hgame week1

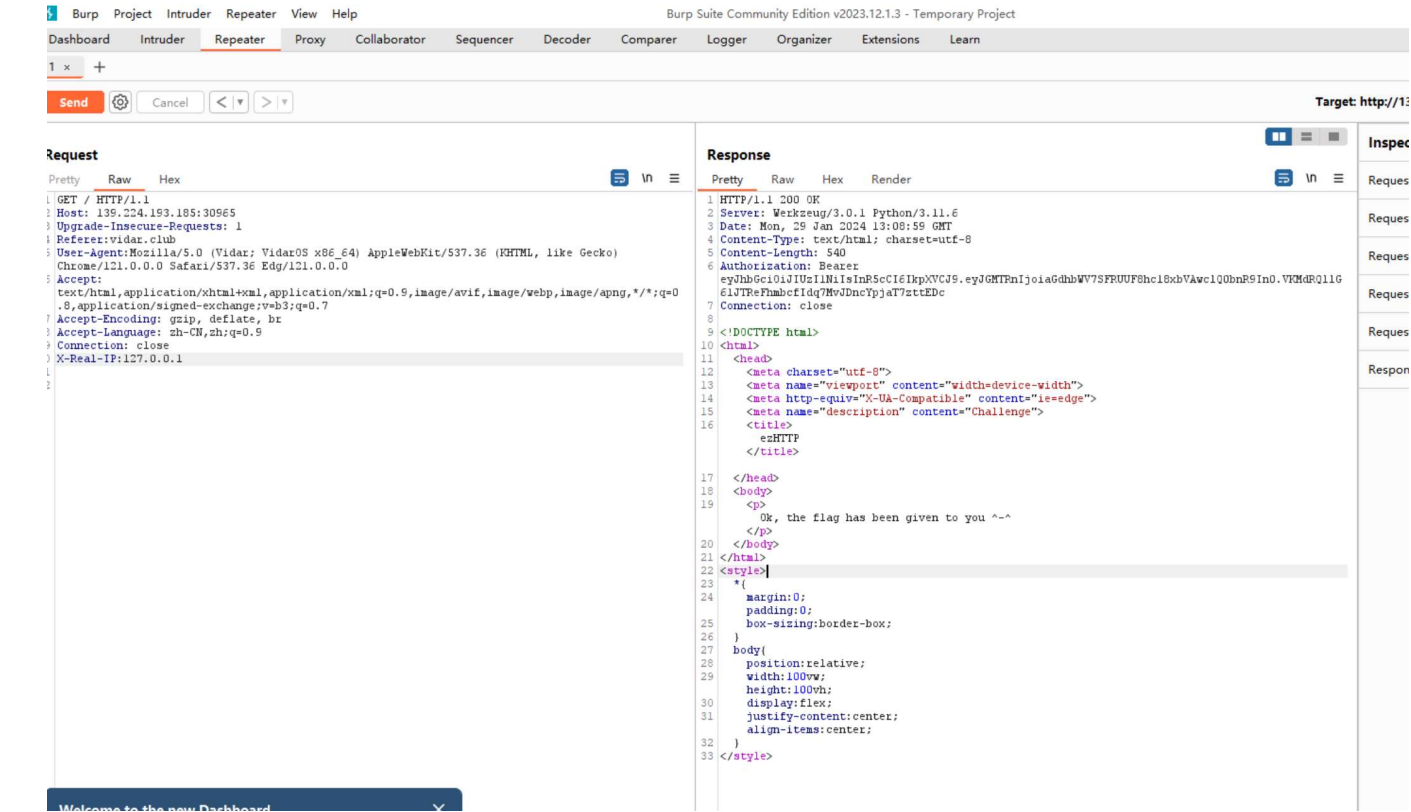
## web

### ezhttp

题目要求从vidar.club进入



修改请求头之后,根据题目修改use-agent并从本地访问,这里普通的 X-Forwarded-For: 127.0.0.1, 或client-ip:127.0.0.1都不可以, 查阅资料, 我们发现了X-Real-IP:127.0.0.1可以实现功能



刚开始以为是需要抓包，尝试了半天结果是上方给的密文，试了一下base64解码，得到flag

bypass it

题目要求我们登录，但是首先要注册，点击注册会根据弹窗返回登陆页面，根据提示，我们禁用js，成功登录

←

→

↻

⚠ 不安全 | 47.100.137.175:32757/login.html

🖨 PTA | 程序设计类实...

📄 竞赛平台

📁 misc工具

📁 php

📁 web语法

📁 ct练习

📁 linux

📄 HDU统一身份认证...

🔍 SQL注入靶场sqli-la...

🏠 BUUCTF

🔒 希望 Bitwarden 为您保存这个密码吗?

从不

编辑

保存

✕

用户登录

• 用户名:

• 密 码:

• ☐ 7天内自动登录

•

设置

⌵ 首选项

工作区

实验

忽略列表

设备

限制

位置

快捷方式

符号服务器

首选项

☒ 自动完成

☒ 括号匹配

☒ 代码折叠

显示空白字符:

无

☒ 调试时以内联方式显示变量值

☒ 触发断点时聚焦源面板

☒ 启用 CSS 源映射

☒ 允许滚动超出文件尾

☒ 使用调试信息调试 wasm 时，如果可能，wasm 字节码上暂停

☐ 允许DevTools 从远程文件路径加载资源，射。出于安全原因，默认为禁用。

默认缩进:

4 个空格

元素

☐ 显示用户代理阴影DOM

☒ 自动换行


☒ 显示HTML 条注释

☒ 悬停时显示DOM 节点

☒ 显示详细检查工具提示

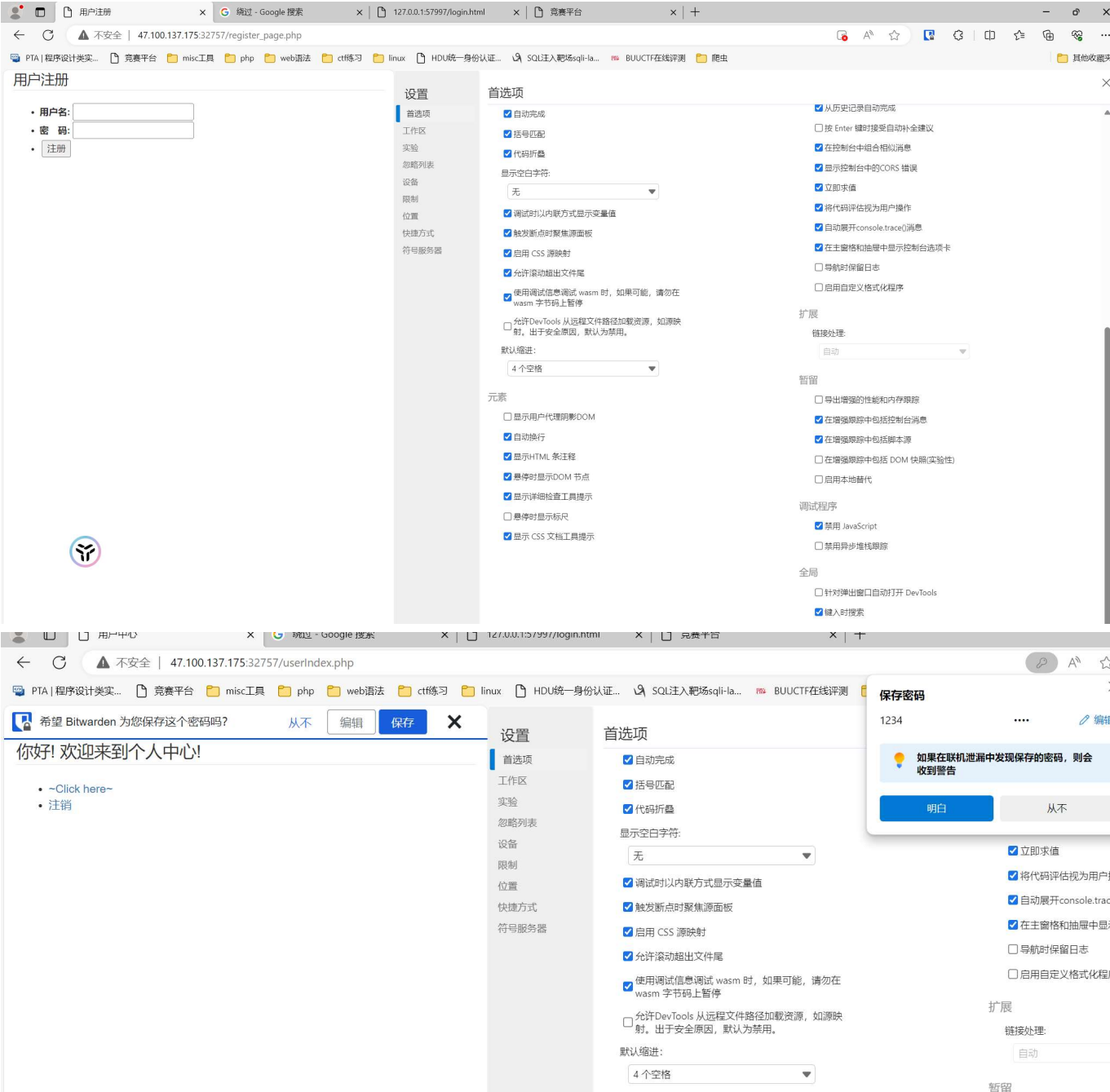
☐ 悬停时显示标尺

☒ 显示 CSS 文档工具提示



file:///D:/vscode\_project/wp/week1/sh4ll0t.html

3/10



Select Courses

这题模拟的是选课系统，随机会有课放出来，因此我们搞个爬虫判断课程是否有余量并不断发送请求

```
import json
import requests
for i in range(5):
    while(1):
        a=requests.get('http://47.100.137.175:30125/api/courses')
        b=json.loads(a.content)
        c=b['message']
        d=c[i]
        check=d['is_full']
        choice=d['status']
        if(check==False):
            headers ={
                "Content-Type": "application/json;charset=utf-8"
```

```

    }
    payload={"id":i+1}
    url="http://47.100.137.175:30125/api/courses"
    e=requests.post(url,data=json.dumps((payload)),headers=headers)
    print(e)
    a=requests.get('http://47.100.137.175:30125/api/courses')
    b=json.loads(a.content)
    c=b['message']
    d=c[i]
    check=d['is_full']
    choice=d['status']
    if(choice==True):
        print(i)
        break

```

2048\*16

游戏要求我们玩到2048\*16的分数即可获得flag,网页源代码是混淆过的js代码, 我们想要分析出在游戏胜利的时候会输出什么内容, 我们查找关键字game

```

g[h(432)][h(469)] = function(x) {
    var n = h
    , e = x ? "game-won" : n(443)
    , t = x ? s0(n(439),
    "V+g5LpoEej/fy0nPNivz9SswHIhGaD0mU8CuXb72dB1xYMrZFRA1=QcTq6JkWK4t3") : n(453);
    this[n(438)][n(437)].add(e),
    this[n(438)][n(435)]("p")[-1257 * -5 + 9 * 1094 + -5377 * 3].textContent = t
}

```

通过分析, 猜测输出的可能是s0(n(439),

"V+g5LpoEej/fy0nPNivz9SswHIhGaD0mU8CuXb72dB1xYMrZFRA1=QcTq6JkWK4t3"),但是未知的变量和函数还需要逐步查找 我们拼接出这一行的关键信息的相关代码

```

function $( ) {
    var x = ["debu", "charAt", "game-over", "push", "tile", "3218200jObBXv",
    "gger", "bestContainer", "firstChild", "chain", "4992592cFfFKg",
    "updateBestScore", "Game over!", "add", "score-addition", ".best-container",
    "over", ".tile-container", "scoreContainer", "counter", "clearMessage", "tile-",
    "tile-merged", "appendChild", "remove", "1457704JdCGrI", "apply",
    "clearContainer", "message", "11358450AckHq", "init", "requestAnimationFrame",
    "addTile", "applyClasses", "\\+\\+ *(?:[a-zA-Z_$][0-9a-zA-Z_$]*)", "value",
    "while (true) {}", "call", "length", "querySelector", "indexOf", "string", "div",
    "tile-new", "function *\\( *\\)", "setInterval", "2589jWZtI", "updateScore",
    "class", "createElement", "score", '{}.constructor("return this")()',
    "4321134sPxlgc", "stateObject", "positionClass", "action", "terminated", "won",
    "tile-position-", "constructor", "join", "fromCharCode", "forEach",
    "textContent", "normalizePosition", "continueGame", "previousPosition",
    "bestScore", "3224mBKYMJ", "1522395ywebnW", "prototype", ".score-container",

```

```

"actuate", "getElementsByTagName", "tile-super", "classList", "messageContainer",
"I7R8ITMCnzbCn5eFIC=6yliXfzN=I5NMnz0XIC==yzycysi70ci7y7iK", "tileContainer"];
    return $ = function() {
        return x
    }
    ,
    $()
}
function F(x, n) {
    var e = $();
    return F = function(t, r) {
        t = t - (-4073 * 1 + 84 * -39 + 7766);
        var a = e[t];
        return a
    }
    ,
    F(x, n)
}
var h=F;
console.log(h(442))
function s0(x, n) {
    for (var e = h, t = 0, r, a, o = 0, c = ""; a = x[e(442)](o++); ~a && (r = t
% (-1 * 445 + -324 + -1 * -773) ? r * (-64 * 33 + -6548 + 8724) + a : a,
    t++ % (-268 * -25 + 166 * -37 + -277 * 2)) ? c += String[e(423)](7397 + 173 *
13 + 1 * -9391 & r >> (-2 * t & 1573 + -2423 * 1 + -856 * -1)) : 3978 + -26 *
153)
        a = n[e(481)](a);
    return c
}
var n = h;
console.log(s0(n(439),
"V+g5LpoEej/fy0nPNivz9SswHIhGaD0mU8CuXb72dB1xYMrZFRAI=QcTq6JkWK4t3") )

```

但是并没有输出答案，查找猜测可能是源代码把x打乱了

```

(function(x, n) {
    for (var e = F, t = x(); ; )
        try {
            var r = -parseInt(e(470)) / 1 + -parseInt(e(466)) / 2 +
parseInt(e(487)) / 3 * (parseInt(e(430)) / 4) + parseInt(e(446)) / 5 +
parseInt(e(493)) / 6 + -parseInt(e(431)) / 7 + parseInt(e(451)) / 8;
            if (r === n)
                break;
            t.push(t.shift())
        } catch {
            t.push(t.shift())
        }
    }
)($, -1 * -639371 + -997 * 937 + 896117 * 1);

```

加入到测试代码后成功输出flag

jhat

java小白对界面可以说是一无所知，只能根据现有的hint，进行查阅资料 jhat

jhat是jdk内置的工具之一。主要是用来分析java堆的命令，可以将堆中的对象以html的形式显示出来，包括对象的数量，大小等等，并支持对象查询语言。使用jmap等方法生成java的堆文件后，使用其进行分析。

又粗查了一些简单的oql语法,长得和sql有点像



查到了相关用法，根据hint,查到资料可以使用exec()函数，查询flag但是由于输出会调用ToString方法，导致无法输出内容，查阅资料得到payload。

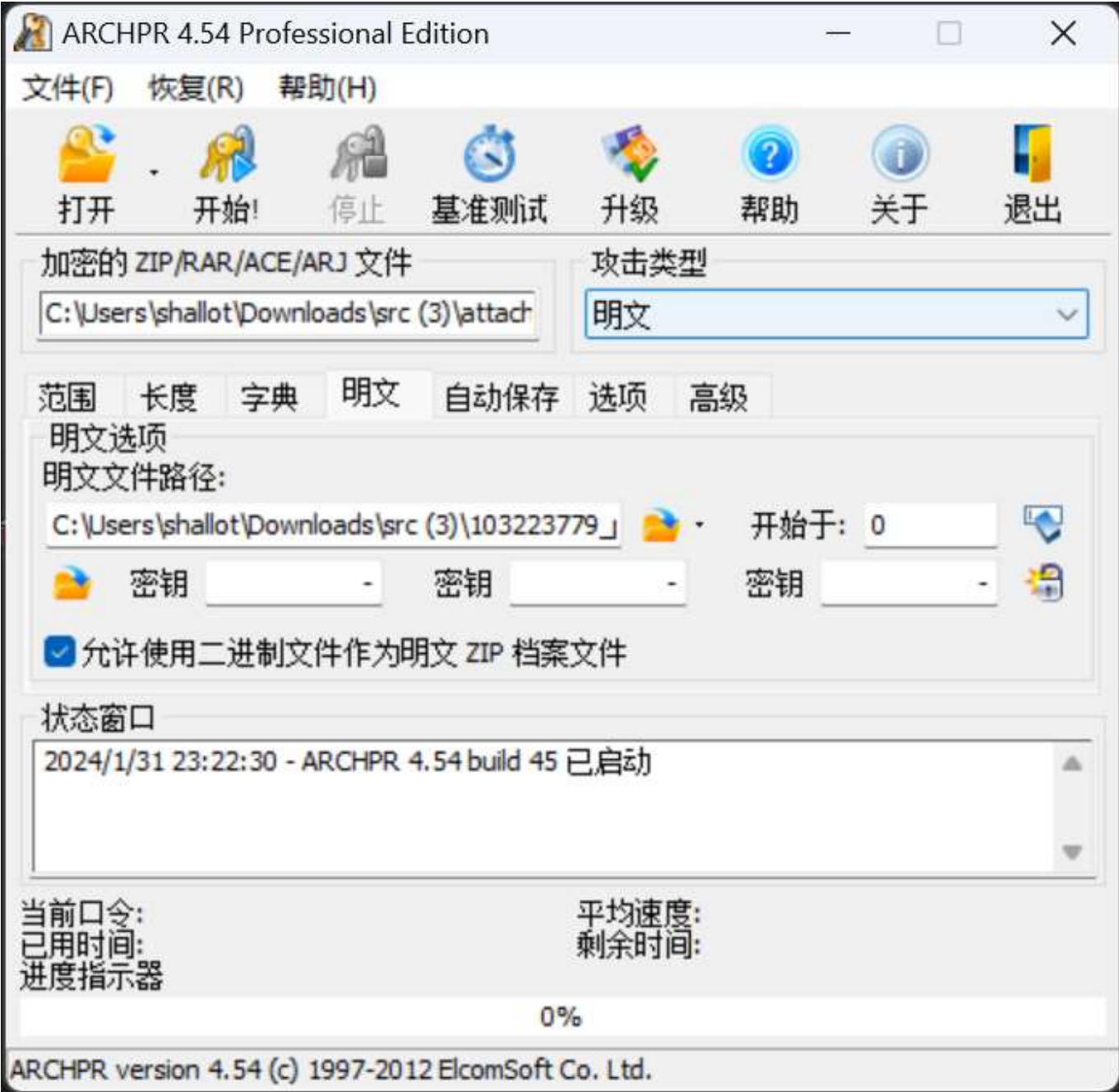


misc

simple\_attack

下载附件之后发现是一个压缩包和一个图片的组合，上网查找了一下有关压缩包的Misc知识，联想可能是明文爆破，





但是出现了报错，询问了学长，发现明文需要用src，爆破成功。打开photo.txt，发现该文本文件是一个以image/png;base64开头的乱码，在查阅资料后，查到需要我们把文本内容复制到浏览器里面，得到flag。

希尔希尔希尔

打开图片是乱码，猜测是修复图片长宽，载入crc32脚本，查验出图片原来的数据，把图片拖到010editor,修改数据，得到原本的图片。



```
importos.py
1  import os
2  import binascii
3  import struct
4
5  crcbp = open("secret.png", "rb").read()    #打开图片
6  for i in range(2000):
7      for j in range(2000):
8          data = crcbp[12:16] + \
9              struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
10         crc32 = binascii.crc32(data) & 0xffffffff
11         if(crc32 == 0x121b804d):    #图片当前CRC
12             print(i, j)
13             print('hex:', hex(i), hex(j))
```

问题 输出 调试控制台 终端 端口

```
PS C:\Users\shallot\Downloads\111> & C:/Users/shallot/AppData/Local/Programs/Python/Python312/python.exe c:/Users/shallot/Down
1394 1999
hex: 0x572 0x7cf
PS C:\Users\shallot\Downloads\111> 
```

```
import os
import binascii
import struct

crcbp = open("secret.png", "rb").read()    #打开图片
for i in range(2000):
    for j in range(2000):
        data = crcbp[12:16] + \
            struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        if(crc32 == 0x121b804d):    #图片当前CRC
            print(i, j)
            print('hex:', hex(i), hex(j))
```

看文件大小，直觉先binwalk一下，确实有隐藏文件，给了好几个文件，折腾了半天，原来就一个secret.txt，根据题目提示，可能是希尔密码，但是缺少密钥，但是还原后的图片还没有用上，原来是还有一层lsb隐写，哇咔咔，密钥到手了。

