

# hgame week1

## web

### cosmos的博客


这是一个文件泄露的题目，出题人暗示用了git，用了git以后会生成一个.git文件，在这个文件夹中的config文件里存放了github的url地址

← → ↻ ⓘ 不安全 | cosmos.hgame.n3ko.co/.git/config



```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  url = https://github.com/FeYcYodhrPDJSru/8LTUKCL83VLhXbc
  fetch = +refs/heads/*:refs/remotes/origin/*
```

根据url找到github上的仓库，在init区域找到base64编码后的flag文件，解码后即可得到flag


**init**  
🔑 master

 FeYcYodhrPDJSru committed 15 days ago Verified 1 parent f79

Showing 1 changed file with 0 additions and 1 deletion.

▼ 1  flagggggggggg 

...    ...    @@ -1 +0,0 @@

1     - base64 解码: aGdhbWV7ZzF0X2x1QGtfMXNfZGFuZ2VyMHVzXyEhISF9

...    ...

### 接头霸王

这是一道考察http头部的题目，根据要求，构造http请求头部,并且试用post请求  
使用hackbar构造

LOAD   SPLIT   EXECUTE   TEST   SQLI   XSS   LFI   SSTI   ENCODING   HASHING   THEME

encytpe  
Enable POSTapplication/x-www-form-urlencoded

ADD HEADER

Body

Name

Value

☒

If-Unmodified-Since

▼

Wed, 22 Jan 2027 05:40:12 GMT

×

Name

Value

☒

User-Agent

▼

www.cosmos.com

×

Name

Value

☒

X-Forwarded-For

▼

localhost

×

Name

Value

☒

Referer

▼

https://vidar.club/

×

## code world

一进去是显示403，查看源代码后发现是302跳转的问题，直接访问index.php也访问不了。网上查询了一下之后，尝试了一下post请求访问index，意外访问成功...

进入index之后显示是要通过url传参，解决一下url编码问题，构造a=3%2b7

## cxk打篮球

查看了网页源代码，感觉像是和前端漏洞有关，后面用bp抓包，发现直接改分数，然后send过去，flag就出来了...虽然出题人本意好像不是如此（滑稽

Send   Cancel   < >

Target: http://cxk.hgame.ws22.cc

Request

Raw   Params   Headers   Hex

POST /evmno HTTP/1.1  
Host: cxk.hgame.ws22.cc  
Content-Length: 47  
Accept: \*/\*  
X-Requested-With: XMLHttpRequest  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36  
Content-Type: application/x-www-form-urlencoded; charset=UTF-8  
Origin: http://cxk.hgame.ws22.cc  
Referer: http://cxk.hgame.ws22.cc/  
Accept-Encoding: gzip, deflate  
Accept-Language: zh-CN,zh;q=0.9  
Cookie: \_\_cfduid=dbcb41d41e5020b9e722ca5b70e05c1979177d47  
Connection: close  
  
score=20000000;c4ddc7cb99a77b35f951691ba282dba0

Response

Raw   Headers   Hex   Render

HTTP/1.1 200 OK  
Date: Wed, 22 Jan 2020 05:59:24 GMT  
Content-Type: text/plain; charset=utf-8  
Content-Length: 49  
Connection: close  
Access-Control-Allow-Origin: http://cxk.hgame.ws22.cc  
Vary: Origin  
CF-Cache-Status: DYNAMIC  
Alt-Svc: h3-24="448"; ma=86400, h3-28="448"; ma=86400  
Server: cloudflare  
CF-RAY: 555f4d36da56d8d9-A05  
  
hgame{j4nA8cIpg\_will\_tell\_yOu\_somethin9\_u5eful777}

## misc

## 欢迎参加hgame

签到题，题目给了一串编码，base64解码后发现是摩斯密码，对照表即可得出flag

## 壁纸

题目下载解压后是一张图片，notepad++打开后，发现有地方写着flag.txt，于是怀疑藏了压缩包，把后缀名改成zip后成功打开，密码是图片的p站id，在网上找了个上传p站图片显示id的网站成功拿到密码，打开压缩包里的flag.txt，发现是/u开头的编码，但是后面只有两位16进制数，在百度上找到Unicode字符列表，一一对应得出flag

# 克鲁苏神话

一开始看hint要用到7zip以为是伪加密，用winhex打开发现并不是，然后查了下压缩包在ctf中的加密，发现一种明文攻击方式。外面和压缩包里有bacon.txt，而且crc32值也一样，采用的是同一种加密方法，7zip是用在给外面的bacon.txt文件压缩，因为明文攻击需要采用同一种压缩方式，之后用工具archpr明文攻击，得到解压缩后的文件。

文件里一个word一个bacon.txt，word打开需要密码，于是自然想到密码藏在bacon.txt里。一开始没有注意到文件名是培根的意思，思索了很久，后来发现是培根密码，大写字母为b小写字母为a，解密后得出word密码。打开word后没有看到flag，百度了下word在ctf中的隐写，把显示中设置为隐藏文字，flag就出现了

## 签到题proplus

这题考的是各种编码还有栅栏与凯撒加密

根据password.txt里的提示，是先进行为分为3组的栅栏加密再进行左移5位的凯撒加密，因为一开始不知道网上有现成的解密网站，自己用py写了个凯撒解密... 栅栏再操作下，跑出英文句子与密码

```
def kaisa(str):
    word = ''
    if str.isalpha():
        if str.isupper():
            word = chr((ord(str) - 65 - 5 + 26) % 26 + 65)
        else:
            word = chr((ord(str) - 97 - 5 + 26) % 26 + 97)
    else:
        word = str
    return word

def main():
    with open('Password.txt', 'r', encoding='utf-8') as f:
        str_line = f.readline()
        first = str_line[0:50]
        sec = str_line[51:101]
        thir = str_line[102:152]
        word = []
        for i in range(0, 50):
            one = kaisa(first[i])
            two = kaisa(sec[i])
            three = kaisa(thir[i])
            word.append(one + two + three)
        print(word[i], end="")

    print()

    psd = 'EMAQETAUQMPVBHVD'
    f = 'EMAQET'
    s = 'AUQMP'
    t = 'VBHVD'
    rs = ''
    for i in range(5):
        rs += f[i] + s[i] + t[i]
    rs += 'T'
    print(rs)
```

```
if __name__ == '__main__':
    main()
```

Many years later as he faced the firing squad, Colonel Aureliano Buendia was to remember that distant afternoon when his father took him to discover ice.

解压后打开OK.txt，发现是ook编码，直接用在线工具解密，解密后得出base32编码；再用工具解密base32，解密得出看到最后两个等号，感觉是base64，直接解密发现不行出现乱码，但是最上面显示png三个字母，于是感觉应该是解密出一张图片；最终找到一个base64转图片的网站，解密出一个二维码，扫码得出flag

## 每日推荐

题目下载解压后是一个pcapng文件，查询后得知用wireshark打开，发现是一大堆流量包；考虑到http中用post传送比较重要的文件，于是就过滤一下看看有没有压缩包之类的，然后就发现可疑的地方

http.request.method=="POST"						
No.	Time	Source	Destination	Protocol	Length	Info
1996	21.009896	192.168.146.132	192.168.146.1	HTTP	1091	POST /wp-admin/admin-ajax.php HTTP/1.1 (application/x-www-form-urlencoded)
3048	28.449637	192.168.146.132	192.168.146.1	HTTP	790	POST /wp-admin/async-upload.php HTTP/1.1 (application/x-zip-compressed)
7258	56.542139	192.168.146.132	192.168.146.1	HTTP	1094	POST /wp-admin/admin-ajax.php HTTP/1.1 (application/x-www-form-urlencoded)
7357	67.747289	192.168.146.132	192.168.146.1	HTTP	1314	POST /index.php?rest_route=%2Fwp%2Fv2%2Fposts%2F13&_locale=user HTTP/1.1 (ap
7393	70.622019	192.168.146.132	192.168.146.1	HTTP	526	POST /wp-admin/admin-ajax.php HTTP/1.1

追踪http流后发现这个php文件里可能藏了一个zip，就在导出对象：http中导出了这个文件。

一开始我直接把文件后缀改成zip，也能正常打开，提示密码六位数字，直接选择密码爆破；但是用工具爆破密码时却说这个不是zip文件，然后我用winhex打开发现开头不是50 4B 03 04，而是在后面一点；就用notepad++打开，把上面的部分和最后面不属于zip文件的部分给删掉，然后再用工具爆破就成功了。

song.zip

-----WebKitFormBoundaryGjwmn57vGB5LC1Kb  
Content-Disposition: form-data; name="action"

upload-attachment

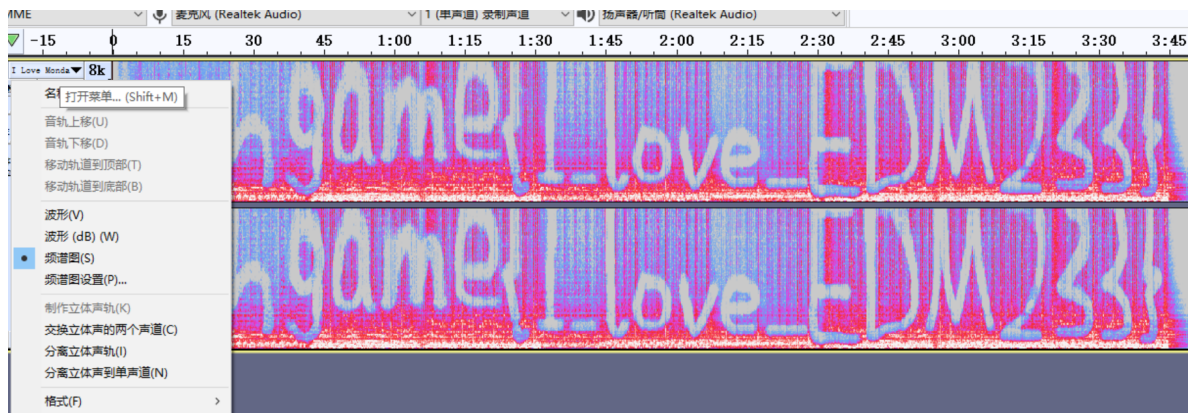
-----WebKitFormBoundaryGjwmn57vGB5LC1Kb  
Content-Disposition: form-data; name="\_wpnonce"

8b76e8452a

-----WebKitFormBoundaryGjwmn57vGB5LC1Kb  
Content-Disposition: form-data; name="async-upload"; filename="song.zip"  
Content-Type: application/x-zip-compressed

PK.....C....-P.....}~...C.....I Love Mondays.mp3.....AE...P...4[.zf..y.ko.M.....n..h..  
[O!..j`.....+~\.^Q..C... .S7...z4k..[]^.....-.-1|F....L....  
.../...z.a..?2..}n.....oR....KSj]...S|.....  
/...'...W.`y-..bR...#.....L.....N...  
d..Q...w.-b...u.-...`.VI.....d.^..D.. 1\*...\.U  
teh 1

解压缩之后是一个mp3文件，百度了一下音频隐写，用audacity打开音频，换成频谱图之后，就出现了flag



## crypto

### infantRSA

根据题目名字的提示，百度RSA加密，然后观察题目py的加密脚本，根据RSA写出解密脚本

```
def main():
    num = 275698465082361070145173688411496311542172902608559859019841
    p = 681782737450022065655472455411
    q = 675274897132088253519831953441
    e = 13
    k = (p - 1) * (q - 1)
    i = 1
    while (k * i + 1) % e != 0:
        i += 1
    rs = (k * i + 1) // 13
    rs = pow(num, rs, p*q)
    print(rs)
    b = int.to_bytes(rs, 25, byteorder='big')
    print(b.decode())
if __name__ == '__main__':
    main()
```

```
39062110472669388914389428064087335236334831991333245
[]hgame{t3Xt600k_R5A!!!}
```

## Affine

看懂py的加密方式后，写出对应的解密脚本

另外关于A, B两个数，因为flag前面几位是hgame开头，我是根据这个联立方程求出A, B的值...

```
TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
def co(num):
    i = 0
    rs = TABLE.find(num) + 62 * i - 14
    while rs % 13 != 0:
        i += 1
        rs = TABLE.find(num) + 62 * i - 14
    rs = rs // 13
    return rs
def maim():
```

```

MOD = len(TABLE)
flag = 'A8I5z{xr1A_J7ha_vG_TpH410}'
print(MOD)
rs = ''
for i in flag:
    if i not in TABLE:
        rs += i
    else:
        num = co(i)
        rs += TABLE[num]
        print(rs)
if __name__ == '__main__':
    main()

```

```

02
hgame{M4th_u5Ed_iN_cRYpt0}

```

## reorder

根据nc 47.98.192.231 25002，尝试了下发下是对输入进行重新排序，直接回车会出现乱序flag；并且重新打开后，重新排序方式是不一样的；关键在于，乱序方式和之前对输入的重新排序一样，于是只要输入和flag相同长度的一串字符串，观察排序方式即可排序出正常的flag。写了个脚本帮忙排序如下

```

def main():
    str_1 = '1234567890qwertyuiopasdfghjklzxc' # 输入flag等长字符串
    rs_1 = '9y642ter80531wq7gcspixlzfhaoukjd' # 输出

    str_rs = '$L{mgpImUteah5+jA}me_!n!TTRP30iu' # 乱序flag
    rs = [0]*32
    for i in range(len(str_1)):
        num = rs_1.find(str_1[i])
        rs[i] = str_rs[num]

    for i in rs:
        print(i, end='')

if __name__ == '__main__':
    main()

```

```

hgame{jU$t+5ImpL3_PeRmuTATi0n!!}

```