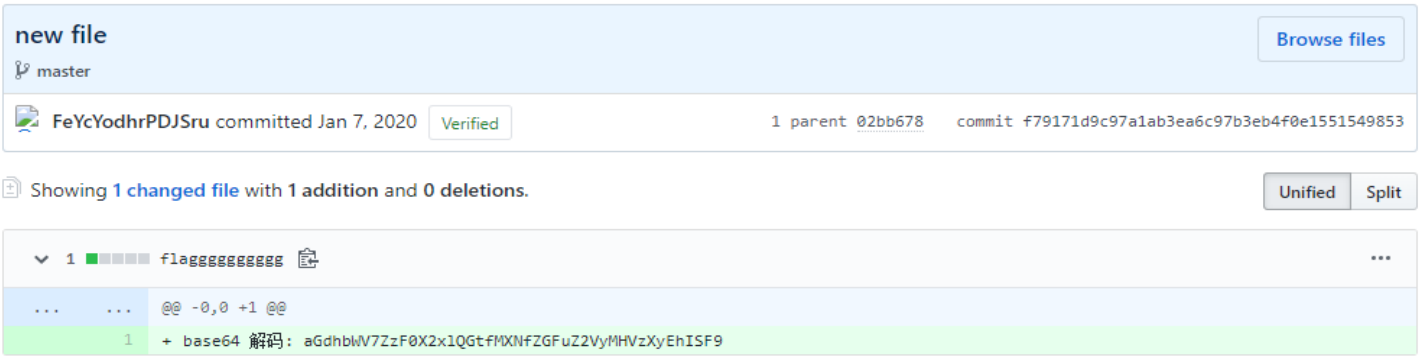
Legrandk 的 HGAME2020 Week 1 WP

1：WEB

1）：Cosmos 的博客

从提示可见可能为 Git 泄露。

使用 GitHack 脚本，可以获取到网站的文件，在文件夹中隐藏的.git 文件夹的 config 文件中，记载着这个项目所在的 GitHub 地址，在这个项目中可以获取到一份叫 new file 的文件，打开可以看到一串 base64 编码



解码这串 Base64 编码，便可以得到 flag

2）：接 头 霸 王

开始提示：You need to come from https://vidar.club/.

使用 HTTP 请求头的 Referer：

添加 Referer:https://vidar.club/

请求之后提示：You need to visit it locally.

使用 HTTP 请求头的 X-Forwarded-For：

添加 X-Forwarded-For:127.0.0.1

请求后提示：You need to use Cosmos Brower to visit.

使用 HTTP 请求头的 User-Agent：

在 User-Agent 项中添加 Cosmos Browser

请求后提示：Your should use POST method :)

把 HTTP 的请求方式从 GET 改为

请求后提示：

The flag will be updated after 2077, please wait for it patiently.

使用 HTTP 请求的 If-Unmodified-Since：

把 If-Modified-Since 改为 If-Unmodified-Since,值仍为 Fri, 01 Jan 2077 00:00:00 GMT

请求后得到 flag

最后的 Burp Suite 中的请求头如下：

```
POST / HTTP/1.1
Host: kyaru.hgame.n3ko.co
Referer:https://vidar.club/
X-Forwarded-For:127.0.0.1
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Cosmos Browser Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
If-Unmodified-Since:Fri, 01 Jan 2077 00:00:00 GMT
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close
```

3）：Code World

对它进行抓包，发现进行了一次 302 跳转，原因是 405 Not Allowed

```
HTTP/1.1 302 Found
Server: nginx/1.14.0 (Ubuntu)
Date: Mon, 20 Jan 2020 14:34:46 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 211
Connection: close
Location: new.php

<html>
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
<center><h1>405 Not Allowed</h1></center>
<hr><center>nginx/1.14.0 (Ubuntu)</center>
</body>
</html>
```

改变方法为 POST 重新请求后的返回数据：

```
HTTP/1.1 403 Not Allowed
Server: nginx/1.14.0 (Ubuntu)
Date: Mon, 20 Jan 2020 14:36:59 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Location: new.php
Content-Length: 161

<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的相加，参数为a<br><br>现在,需要让结果为10</center>
```

提示需要进行 人 鸡 验 证，通过 URL 传参使 a 的值为 10

加入参数?a=5%2b5 （其中%2b 为'+'的 URL 编码）

请求头如下：

```
POST /?a=5%2b5 HTTP/1.1
Host: codeworld.hgame.day-day.work
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
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
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close
```

从返回包中得到 flag

4）：🐴尼泰玫

30000 分过关，就不老实打球了。

浏览器按 F12，审计 JS 代码，在 game.js 里发现了有关分数控制的代码，在第一行把 storageScore 和 globalScore 的初值改到 30000 以上，保存。

然后开启游戏，等球掉到地上，你会发现分数变成了 30000+，浏览器弹窗得到 flag，结束。

2：MISC

1）欢迎参加 HGame！ （签到题）

题设的字符串为

```
Li0tIC4uLi0tIC4tLi4gLS4tLiAtLS0tLSAtLSAuIC4uLS0uLSAtIC0tLSAuLi0tLi0g
Li4tLS0gLS0tLS0gLi4tLS0gLS0tLS0gLi4tLS4tIC4uLi4gLS0uIC4tIC0tIC4uLi0t
```

为 Base64 编码，解码得到

```
.-- ...-- .-.. -.-. ----- -- . ..--.- - --- ..--.- ..--- ----- ..--- ----- ..--.- .... --. .- -- ...--
```

可知它是摩尔斯电码，按照对照表对其解码（某些在线解码不会解字符）

得到：W3LC0ME_TO_2020_HGAM3

按照题目提示加上 hgame{}，得到 flag

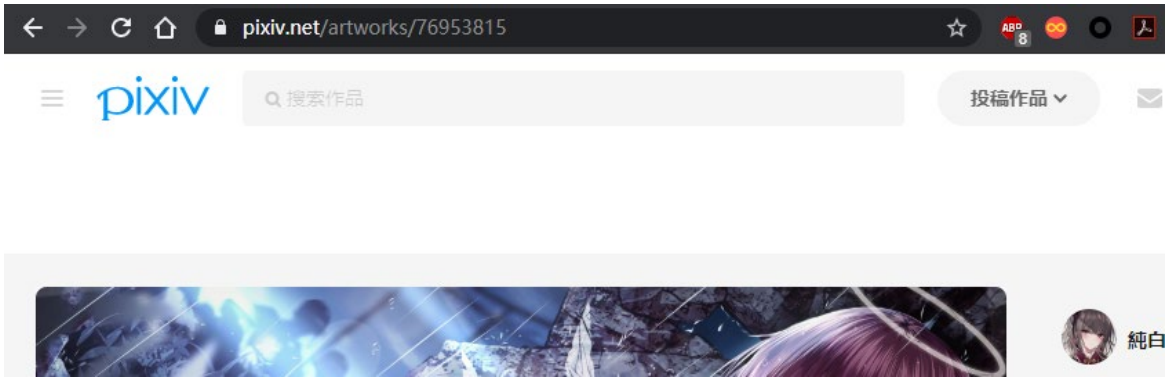
2）壁纸

下载附件打开，得到一张 jpg：

用 binwalk 或者直接改后缀名为.zip 可以得到一份加密的压缩文件。

根据提示：Password is picture ID 以及图片的文件名：Pixiv @純白可憐.jpg

去 Pixiv 网站上搜索 純白可憐 画师，找到这张画，在 URL 中可以看到这张画的 id：76953815



把它作为密码解压压缩文件，得到 flag.txt，得到了一串 Unicode 编码的字符串：

```
\u68\u67\u61\u6d\u65\u7b\u44\u6f\u5f\u79\u30\u75\u5f\u4b\u6e\u4f\u57\u5f\u75\u4e\u69\u43\u30\u64\u33\u3f\u7d
```

对它解码就可以得到 flag

3）克苏鲁神话

解压下载到的压缩文件，得到 Bacon.txt 和加密过的 Novel.zip。把 Bacon.txt 用 7zip 进行压缩，压缩后的 Bacon.txt 的 CRC32 值以及大小与 Novel.zip 中的一样，遂使用 ARCHPR 进行明文攻击。攻击后可以得到一份加密的 doc 文件。

Bacon.txt 中有一串大小写紊乱的字符串，联系文件名猜想是培根密码。将原句小写转为 a，大写转为 b，得到一串有 75 个字符构成的 a、b 字符串，对它进行培根密码解码，得到 doc 文件的密码：FLAGHIDDENINDOC

打开 doc 文件，在 Word 的开始->选项->显示->始终在屏幕上显示这些格式标记 中勾选隐藏文字，在文件末尾就可以看见 flag 了。

4）签到题 ProPlus

解压下载得到的压缩文件，得到 Password.txt 和有加密的 OK.zip

对 Password.txt 中的字符串先按每组 3 个字符进行栅栏密码解码，再以位移为 5 对它进行凯撒密码解码，得到一句英文语句以及 zip 文件的密码：EAVMUBAQHQMVEPDT。解压 zip 文件，得到 ok.txt，获得了一串 Ook!编码，对它进行解码，得到一串 Base32 编码，再对它解码得到一串 Base64 编码，对它以 16 进制解码，得到一串以 89 50 4E 47 开头的 16 进制符号，可知它是一张 png 图片的 16 进制码，用 winhex 把这串 16 进制码转换成 png 图片，得到一张二维码：  扫描即可获得 flag

3：Reverse

1） maze

下载得到的内容拖入 IDA64 内，查看 main 函数按 F5 查看伪代码。

可见迷宫按照

w：+64

a：-4

s：+64

d：+4

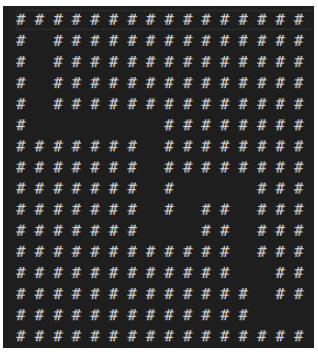
的规则走，遂猜测地图大小是 16*16.

看以下语句：

```
if ( v5 < (char *)&unk_602080 || v5 > (char *)&unk_60247C || *(_DWORD *)v5 & 1 )
  goto LABEL_22;
```

可见移动路径应在 unk_602080 至 unk_60247C 之间，且只走偶数。

查看该段内存，得到一串由 00 和 01 构成的 16 进制符号。以每 4 字节为一单位，每单位倒序组成一个 2 进制数，组成 16*16 的一个矩阵。把奇数数字表示为‘#’，偶数数字表示为空格，可得迷宫：



运行这段程序，输入路径，即可得到 flag

4：Crypto

1）.InfantRSA

用 python2 的 gmpy2 库计算，脚本如下。

```
import gmpy2

p = gmpy2.mpz(681782737450022065655472455411)
q = gmpy2.mpz(675274897132088253519831953441)
e = gmpy2.mpz(13)
phi_n = (p - 1) * (q - 1)
d = gmpy2.invert(e, phi_n)
print ("private key:")
print (d)

c = gmpy2.mpz(27569846508236107014517368841149631154217290260855985901984l)
print ("plaintext:")
print (pow(c,d,p*q))
~
~
~
```

算出来的 plaintext 就是 m

再用 python3 的 int.to_bytes 函数把 m 转化成 byte，脚本：

```
m = (39062110472669388914389428064087335236334831991333245)

print(int.to_bytes(m,length=100,byteorder='big',signed=False))
```

运行，得到 flag