web

Cosmos的博客

这一关真的卡了我好久……先是学git，然后配环境，Linux又不熟，疯狂试错……

先看见github，百度查到git泄漏，并找到工具GitHacker。于是在Linux装环境（Python版本低、Git没有……），之后用

```
python GitHack.py http://www.example.com/.git/
```

获得网站文件（git名词太多，忘了叫啥了……好像是工作区和暂存区啥的。。。）

```
[root@izuf6ajm9b61u88n8oc3e6z cosmos_hgame_n3ko_co_]# ls
index.html  static
```

打开index.html，却还是原先的网站。用 git log查看一下commit，也只有

```
[root@izuf6ajm9b61u88n8oc3e6z cosmos_hgame_n3ko_co_]# git log
commit 051894e2ed400a7195008f7022a241e68f5a1335 (HEAD -> master)
Author: Auto-Deploy <e99plant@vidar.club>
Date:   Tue Jan 7 18:54:09 2020 +0800

    init
```

一个commit，进去找了一番发现还是没什么东西。

之后用ls -a，发现了还有个隐藏文件夹.git

```
[root@izuf6ajm9b61u88n8oc3e6z cosmos_hgame_n3ko_co_]# ls -a
.  ..  .git  index.html  static
[root@izuf6ajm9b61u88n8oc3e6z cosmos_hgame_n3ko_co_]# cd .git
[root@izuf6ajm9b61u88n8oc3e6z .git]# ls
config  description  FETCH_HEAD  HEAD  hooks  index  info  logs  objects  ORIG_HEAD  refs
```

进去后发现东西不少，就一个个慢慢看……

在config里发现了

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

远程仓库有个url，下载来看一下，然后探索后发现

```
[root@izuf6ajm9b61u88n8oc3e6z 8LTUKCL83VLhXbc]# git log
commit 6d66acf3227cf85d5ba2ea55df9bc164f953541d (HEAD -> master, origin/master, origin/HEAD)
Author: FeYcYodhrPDJSru <53310630+FeYcYodhrPDJSru@users.noreply.github.com>
Date:   Tue Jan 7 18:32:32 2020 +0800

    init

commit f79171d9c97a1ab3ea6c97b3eb4f0e1551549853
Author: FeYcYodhrPDJSru <53310630+FeYcYodhrPDJSru@users.noreply.github.com>
Date:   Tue Jan 7 18:32:14 2020 +0800

    new file

commit 02bb67805ab0f6f9cbe92ab5dc9269e99f0bf361
Author: John Wu <524306184@qq.com>
Date:   Tue Jan 7 18:09:05 2020 +0800

    init
```

第二个commit非常可疑！check一下

```
[root@izuf6ajm9b61u88n8oc3e6z 8LTUKCL83VLhXbc]# ls
flagggggggggg  index.html  static
```

进去是个base64加密的密文（明着写base64加密……），找个网站解密得flag.

接头霸王

第一条，要从“https://vidar.club/”访问。

意识到是加HTTP请求头的题，跟着加呗。

Referer:https://vidar.club/

第二条提示来了，要从localhost访问。

于是——X-Forwarded-For:127.0.0.1

第三条：用Cosmos 浏览器访问

于是——User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Cosmos

即将已有的User-Agent后面的浏览器改为Cosmos即可

第四条：用POST方法

于是——更改请求头即可

第五条：flag要在2077年后更新，请等待。

刚开始我以为是有date首部设为2077年以后，但发送过去并不成功。

发现：

Name	Value
HTTP/1.1	200 OK
Content-Length	1231
Content-Type	text/html; charset=UTF-8
Date	Sun, 19 Jan 2020 04:26:00 GMT
Last-Modified	Fri, 01 Jan 2077 00:00:00 GMT
Server	HGAME 2020
Server	Apache/2.4.29 (Ubuntu)
Vary	Accept-Encoding
Connection	close

有一个Last_Modified首部为2077年一月一日。

然后查到有两个请求首部：If-Modified-Since和If-Unmodified-Since

前者给出一个时间，如果该时间以后修改过就将资源返回，未修改过返回304；

后者给出一个时间，如果该时间以后未修改过就返回资源，修改过返回412；

但奇怪的是，我用If-Modified-Since，时间怎么设都没用，只有用后者才能得到正确结果，不知道为什么……总之这是最后一道关，过了flag就返回了。

后来才知道前者只能用GET和HEAD头，因此改成POST以后前者失效了

Code is exciting

这题也是私聊Annevi大佬才得到的提示。Web也得看内容说话，不能想当然啊~

进去之后先是一个403愣了一下，先F12查看，有一段吐槽，于是得知是302跳转（当时看见那个building site以为会和去年一样有什么备份文件或者其它的东西，还各种奇奇怪怪的文件名试了半天，还好没拿字典跑……），用Burpsuite抓包，看见了302跳转前的页面是405 not allowed，这里我也卡了很久。我以为是找错页面了，然后百思不得其解。后来私聊浩哥问思路，他说405错误也是一个信息。百度以后才得知405是请求方式出了问题

405概念

> 请求行中指定的请求方法不能被用于请求相应的资源。该响应必须返回一个Allow 头信息用以表示出当前资源能够接受的请求方法的列表。 鉴于 PUT，DELETE 方法会对服务器上的资源进行写操作，因而绝大部分的网页服务器都不支持或者在默认配置下不允许上述请求方法，对于此类请求均会返回405错误。

改为POST以后，需要URL传参，格式是a+b。依旧是百度到URL编码会将“+”变为“space"，必须用%2B，使用后即得到flag。

Summary:不得不说只看一遍《图解HTTP》还是太水了，这周这几道题懂了以后才知道都是最简单的类型，但就是对HTTP各个方面的陌生才让我纠结了如此之久，不得不说是很令人沮丧的一件事情。学习任重道远啊，HTTP还得再多去了解，有条件最好《HTTP权威指南》读一读。想以Web为方向，基本功这么差怎么行，这才第一周啊……希望能挺过去四周。。。

鸡你太美

最后一天才做出来的……一直不是很了解这种东西该从何入手。后来与Moesang私聊后，得知和国庆那道几百万分的题是一个级别的题，于是去看JS代码……

这个其实算是我一直困惑又非常愚蠢的问题。我一直觉得JS是从服务器加载过来的，为什么在Chrome的source修改内容会对实际页面产生影响呢？之前那个题是在默认HTML改的，这个是在源文件改的，我一直搞糊涂了。

其实这两个本质上好像都是一样的……加载过来的JavaScript，它运行的是在前端，后端处理应该有其它语言。当我把Chrome的缓存开启后，网页在缓存过期前都是不会再向服务器请求数据的。除非传说中的web2.0，交互式网页啥的。。（emmmm有点在乱说了，）

好的最后在main函数里，找到一个类，把类的初始值都赋了三万分，再开游戏，游戏结束就得到flag了。

crypto

InfantRSA

RSA嘛，毕竟我大二了，信安数学也刚刚学过，而且题目给的条件真的很多，就是解一下就出来的题目。然而……

这题看着正规点的办法应该是这样的，利用Python的gmpy2库进行计算，真的很容易，以下皆为做题时的吐槽

由于对Python的不熟悉，中间折腾了好长一段时间。我先自己写的Python RSA求解代码（不得不说惊到我了，Python没有引入任何库的情况下，对大数计算的速度如此之快，真的好用……），在计算c^d时炸了，跑不出来，私聊Lurkrul后叫我再去看看RSA……后来意识到还有个模平方的算法，害，怪不得信数学不好。

又自己写了个模平方，然后将结果对照着给出的字符转换代码转回去，发现乱码了……

之后各种尝试，国内国外RSA在线计算都试了一遍，各种无法计算、算法不同，更有甚者，这么大数据输进去把网站给轰出源码报错信息来了……

我也尝试安装gmpy2库，然后死活导入不进来，万分无奈还是走上了自己写RSA的路子。

利用了网上的扩展欧几里得算法，颤颤巍巍的写了一天，调完了bug，总算能用了……

代码像裹脚布，太臭长了就不放了，最后做出来总还是开心的。

网上在线计算RSA的时候发现，其实我对于密码学这一块根本不懂。比如数字签名、证书这种东西。网上的RSA解密都是以证书和公钥、密文这些来的，与信数那种最简单无脑的RSA根本不一样。学密码学任重而道远。密码学据说也不能算一个主方向，那更好了，就当是辅助Web和对数学的一丢丢记挂组合起来而想学的东西吧。

Affine

入目而来的python代码，先读……

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import gmpy2
from secret import A, B, flag
assert flag.startswith('hgame{') and flag.endswith('}')

TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)

cipher = ''
for b in flag:
    i = TABLE.find(b)
    if i == -1:
        cipher += b
    else:
        ii = (A*i + B) % MOD
        cipher += TABLE[ii]

print(cipher)
# A8I5z{xr1A_J7ha_vG_TpH41O}
```

总之就是从secret库中引入三个参数，A，B和flag，然后已知一个TABLE，根据这些东西对flag进行加密，加密结果以注释的形式给出。

加密的过程就是，TABLE表包含所有大小写字母和数字，不在此内的字符原封不动，在此内的字符进行一定规则的轮换。

```
flag = ''
cipher = 'A8I5z{xr1A_J7ha_vG_TpH41O}'
for b in cipher:
    i = TABLE.find(b)
    if i == -1:
        flag += b
    else:
        flag += 'a'
```

反正……照着给出的代码复制就是了……虽然不知道AB是啥，但这样先看看flag的格式……

aaaaa{aaaa_aaaa_aa_aaaaaa}

算是意外之喜？hgame的flag都是这种形式，于是hgame对应于A8I5Z，根据这个计算AB（我用最蠢的两层循环……）

```
flag = 'hgame'
flag1 = 'A8I5z'
for A in num:
    for B in num:
        cipher = ''
        for b in flag:
            i = TABLE.find(b)
            if i == -1:
                cipher += b
            else:
                ii = (A*i + B) % MOD
                cipher += TABLE[ii]
        if cipher == flag1:
            print(A, B)
```

我算出来A，B有好几组解，不过后来发现都是模 6 2 同余的。

解密代码是

```
def origin(ii):
    index = 0
    while(index <= 62):
        if (13*index + 14 - ii) % 62 == 0:
            return index
        index = index + 1

for b in cipher:
    i = TABLE.find(b)
    if i == -1:
        flag += b
    else:
        ii = TABLE.find(b)
        flag += TABLE[origin(ii)]
```

该题干掉了。

Reorder

这题真的，给人整懵了……看题目有Oracle，查半天，太复杂看不懂，无果。而且nc连接以后试，输一堆乱七八糟的东西，输着输着跳出来一个"RUAAAA！！！"，emmmm……然后跳出一串字符串，看不懂的字符串，猜测是与flag有关。后来多试了几次，结合题目，确实有规律。Reorder不就重新排序嘛！于是输入'A-Z'+'0-9'，输出的字符串一一对应nc给出的字符串，然后再逆回去位置，就得到flag了。

```
index = "abcdefghijklmnopqrstuvwxyz0123456789"
a = "aejdpcknbifhmolgquzt5s03ryvx241w6  9 8  7"  # 输入a-z对应的输出数据
b = "hetmLa+mg${UIp5j3RTe}Pi!_AmTn!Ou"  # 输入flag对应的输出数据
flag = ""
for i in index:
    t = a.find(i)
    flag += b[t]
print(flag)
```

这个脚本我bug还没调……因为a比b长，访问b的时候会越界。我也不管了，从debug窗口调出flag，看见完整的就交了233333

misc

欢迎参加HGame!

把题目给的字符串扔百度，发现是摩斯码base64加密生成的字符串。于是base64解密，然后摩斯码解密得到结果。

壁纸

直接把扩展名改为.zip尝试，尝试成功……（说实话正规方法好像是用binwalk或者winhex等16进制文件查看工具，找到是否有特殊文件的特殊字段这种方法的。不过听说过图种嘛，直接用zip尝试就出来了……），看见了flag.txt，zip备注提示密码为图片ID，右键看属性发现ID是空的。百度以后查ID，发现有个网站可以查，查了以后提交，于是得到flag

签到题proplus

根据提示

（话说这提示花了我一段时间读的……，不知道这个first和next什么含义……而且这个还是二次加密，也就是说一次解密后还不知道对不对……）

先用栅栏密码解密，先试的分成三组，试了好多次不太对，试一下每组三个的解密，解出来一段像英文句子的句子。后来再凯撒密码解密，发现是《百年孤独》的第一段话……

那么接下来就好多了。把第二部分全大写的字母以同样方式操作一下，然后打开另一个压缩包。

网上查阅得是和brainf**k一样的搞人的语言……在线解密后获得base32，再解密获得base64，然后base64解密后发现unicode编码前面有PNG几个字。查询PNGhex文件头，发现确实是PNG文件。利用在线工具base64转图片后，获得二维码，扫取获得flag。