Web1——这 TM 是啥

首先查看源代码，可以看出是 jjencode 加密

找到倒数第三个小括号内包含的内容为运行的表达式内容。使用 console.log()将括号内的内容显示在浏览器控制台中

```
var f = "hctf{j5fuck_1z_m1233}"; alert("Hack by LoRexxar, 你的flag被我拿走了")
```

Web2——我是谁我在哪

不知道自己怎么做出来的。。。反正我就是试了一下 index.php 这个页面。然后就响应了一个 flag

Web4——不可能拿到的 flag

通过代码可以看到，需要 name 的值和 password 的值不相等，但是他们的 sha1 值要相等，首先想到特殊值，但是特殊值比较难找。所以用绕过的方式来解决，因为 sha1 传入的是字符串，所以我们通过传入数组，是其在强制转换后数值相等。最后 post:

name[]=1a&password[]=1

最后得到：Flag: hctf{o0k!!g3t_f14g_s0_ez}

Web5——PHP 真可怕我要回农村

首先通过代码审计可以发现，需要$b 的值为 0.7 以及第 11 位数为 0，并且$b 不能为数组。所以决定 post 字符串，post:: b=0.73xxxxxx0

最后获得：hctf{wochubuxiaqule_over}

Web3——神奇的数字

通过代码可以发现，要获得 flag 需要满足

number = intval(number)

intval(number) = intval(strrev(number))

并且不能是回文数，但是第二个条件又需要满足回文数，这时候由于 64 位的 integer 值长度为 9223372036854775807，但是，它的回文数明显小于 64 位系统的限制，所以我们想到前面加个 0，最后 post：number=09223372036854775807 通过 PHP 中的整数溢出可得到 hctf{go0d_job_intv4l_iz_g00d}

Pentest——lightless 的渗透教室入门篇（一）

根据题目提示，向 http://115.28.78.16:13333/pentest/01/?hacker=HelloGet post：hacker=HelloPost

获得 flag: hctf{PostAndGetIsSoEasy_comeon!}

Pentest——lightless 的渗透教室入门篇（二）

在 burpsuit 中修改 HTTP 头请求：

```
POST /pentest/02/ HTTP/1.1
Host: 115.28.78.16:13333
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 99 like Mac OS X)
AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A403
Safari/8536.25
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
remoteAddress: 127.0.0.1
x-forwarded-for: 127.0.0.1
x-real-ip: 127.0.0.1
Referer: www.google.com
```

获得 flag: hctf{h77p_He4dEr_50_E4sy_ANd_fUn_ohhouhou}

PENTEST——lightless 的渗透教室入门篇（三）

查看响应头可以看到：一条 cookiecontent: admin=1 and isLogin=true

根据题意构造 HTTP 头：cookie:admin=1;isLogin=true，发送 post 包

之后可以看到看到 flag： hctf{hao_hao_kan_zi_liao!!!}

MISC——Explorer 的图库之一

```
ÿØÿà  JFIF
    ÿá ¦Exif  II
*  hctf{2e3e3}
```

Winhex 查看下载下来的文件。获得 flag

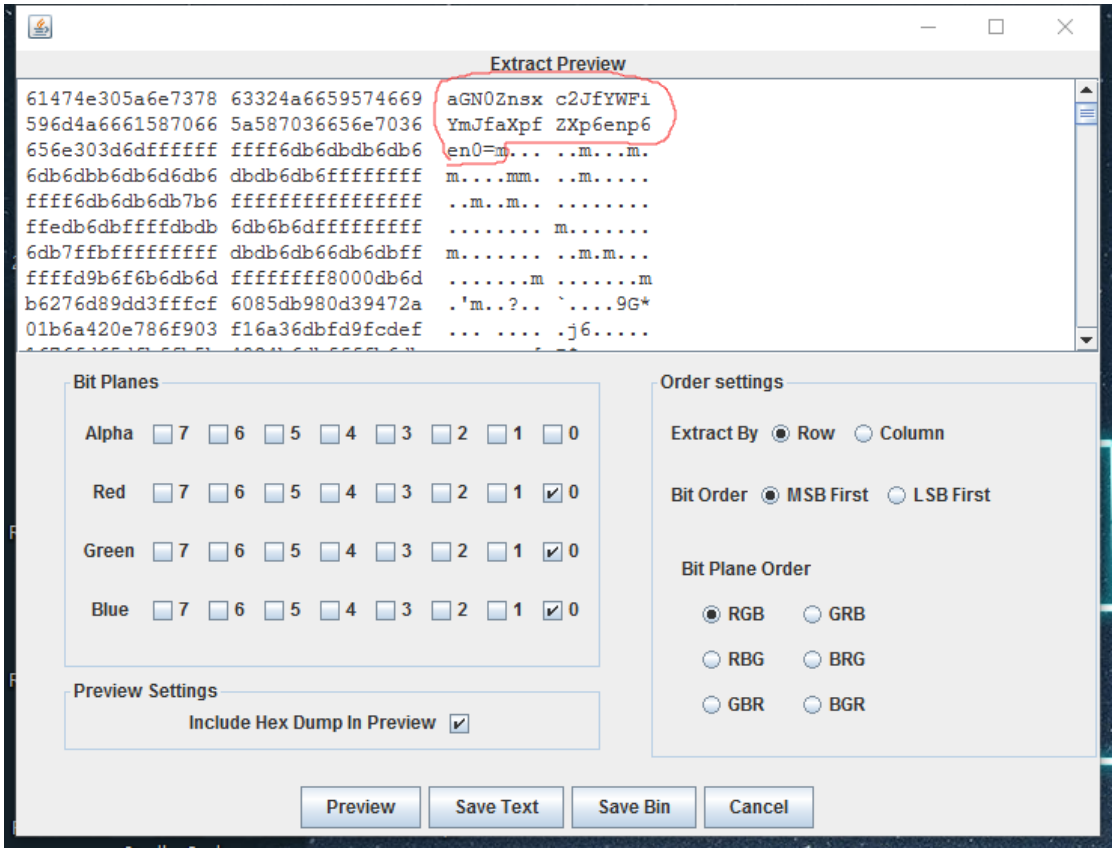
MISC——Explorer 的图库之二

使 用 binwalk 查 看 文 件

尝试分离，分离出一个 tar 文件和一个 zilb 文件，用 winhex 查看文件或在 linux 下解压 tar 文件可得 flag：hctf{nizh1dao_tuzh0ngm4}

MISC——Explorer 的图库之三

由于使用 binwalk 并没有将 png 文件提取出来，所以我们根据 binwalk 提供的数据块数在 linux 下用 dd 命令进行手动提取，获得一张 png 图，放入 StegSolve.jar 中查看 RGB 的最低 位



框选中的位置可以明显的看出 base64 的编码方式，解码可得 flag：
hctf{1sb_aabbb_iz_ezzzzzz}

RE——你看看，逆向多简单！

拿到程序，用 winhex 打开，查找文本 hctf。获得 flag：hctf{It_Is_T0o_ea5y!}

CRYPTO——密码学教室入门（二）

手撸凯撒加密，根据前面的 hgame 开头算出偏移量，最后的数字是通过无数次尝试得出的。

CRYPTO——密码学教室入门（四）

首先写了个 python 分解 n，脚本如下：

```
n=2
while (n<129207199286820063175608266250129174468363201692431902431119351352347485179222188174828106158203143126249471611051778767902474016169731274422311717901231515021876712141181312401542252372002461522452361177389611592961731751747420814920146139240541223925247193113883241051751981632135691532218310524541192160852063723013919419149 76):
    if (129207199286820063175608266250129174468363201692431902431119351352347485179222188174828106158203143126249471611051778767902474016169731274422311717901231515021876712141181312401542252372002461522452361177389611592961731751747420814920146139240541223925247193113883241051751981632135691532218310524541192160852063723013919419149 76%n == 0):
        print n,920139713/n
    n = n + 1
```

最后得到太多数了，从中抽出一对出来，算出 d=1，用大数计算器，算得 6867616D657B7273615F31735F737469316C5F653473795F6E6F77217D

转为 ascii 得到 flag: hgame{rsa_1s_sti1l_e4sy_now!}

CRYPTO——密码学教室入门（一）

直接用大数计算器通过 p,q 算出 n, 再使用私钥 d 算出 m= 6867616D657B7273615F31735F763372795F65347379217D

转成 ascii 得到 flag：hgame{rsa_1s_v3ry_e4sy!}

CRYPTO——密码学教室番外篇

正常解字母得到 hgame{dgfdyhcry19954902180//+/%}

再通过程序暴力跑出 flag：hgame{dgfdyhcry42287235413//+/%}

程序：

```c
#include "stdafx.h"
#include "string.h"

char caiser[] = "hgame{dgfdyhcry19954902180//+/%}";
int main()
{
    int k = 9;
    for(int j=0;j<10;j++)
    {
        for (int i = 0; i < strlen(caiser); i++)
        {
            if ((caiser[i] >= '0')&&(caiser[i] <= '9'))
            {
                caiser[i]++;
                if (caiser[i] > '9')
                    caiser[i] = caiser[i] - 10;
            }
        }
        puts(caiser);
    }
    system("pause");
    return 0;
}
```

PWN——pwn0

将文件放入 ida 查看，查看到一个 getflag 函数，还有一个 foo 函数可以调用 getflag 函数，但需要满足 a1==1633771873 ,而在 main 函数中传入的 a1 值为 305419896，但我们可以发先在判断 a1 之前存在一个 gets（），由于 gets()函数没有长度限制，所以需要对 gets 进行溢出，只要 gets 的字符串长度超过 0x1c 后就会覆盖后面的堆栈数据，我们只需要输入 36 个 a 即可成功 pwn。pwn 后获得 flag。