

AWFUL

题外话：这一周让我学会了。。现场学现场写,可能给的参考文章会很多，谅解下。

Web

分数 250:Random?

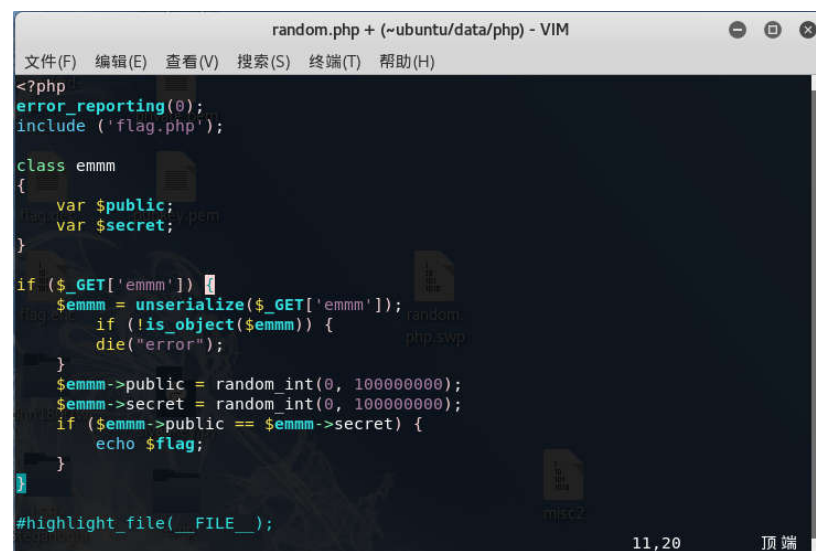
描述：多 random 几次没准就随机到一样的值呢 PS:网不好 vim 线上改代码真是致命

URL: <http://123.206.203.108:10001/random.php>

打开什么都没有，所以这里就不给出截图了。

查看源代码也没有。查看标题出现关键字 Vim，之前写过一道 Web 题它涉及到的知识点也是 Vim[1][2]（但是他没有像这道题目一样直接给出来，而是写在了源代码中，这里后面会给出相关网站），所以这里直接在输入网址

<http://123.206.203.108:10001/.random.php.swp>，下载得到文件 random.php.swp。将其放到 LINUX 系统中，终端输入“vi -r random.php.swp”[3]，得到网站源代码



```
<?php
error_reporting(0);
include ('flag.php');

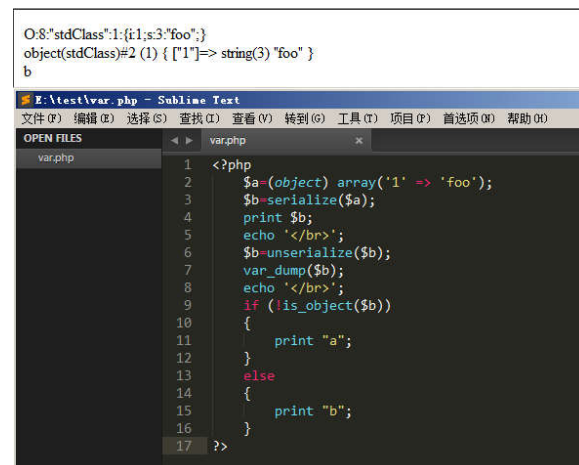
class emmm
{
    var $public;
    var $secret;
}

if ($_GET['emmm']) {
    $emmm = unserialize($_GET['emmm']);
    if (!is_object($emmm)) {
        die("error");
    }
    $emmm->public = random_int(0, 100000000);
    $emmm->secret = random_int(0, 100000000);
    if ($emmm->public == $emmm->secret) {
        echo $flag;
    }
}

#highlight_file(__FILE__);
```

其中 unserialize()函数为反序列化（类似解密方式，serialize()函数为加密），整个网站代码大致意思为：获取网站中叫做“emmm”参数的值，并对他进行反序列化操作，在这之后如果他不是对象的话，就会打印 error 并退出，如果是对象就随机抽取 2 个 0 到 100000000 的值，如果一样就跳出 flag。

那么如何创建对象呢（听起来怪怪的），经查询发现数组可以强制转换为对象，比如\$a=(object) array('1' => 'foo');[4]，unserialize()为反序列化，所以我们要输入的参数内容为经过序列化的内容，那么在本地搭建的 php 环境下测试得到他的具体内容

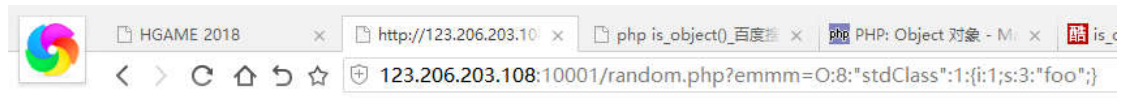


```
O:8:"stdClass":1:{i:1;s:3:"foo";}
object(stdClass)#2 1 { ["1"]=> string(3) "foo" }
b

E:\test\var.php - Sublime Text
文件(F) 编辑(E) 选择(S) 查找(F) 查看(V) 转到(G) 工具(T) 项目(P) 首选项(O) 帮助(H)
OPEN FILES
var.php
1 <?php
2 $a=(object) array('1' => 'foo');
3 $b=serialize($a);
4 print $b;
5 echo '<br>';
6 $b=unserialize($b);
7 var_dump($b);
8 echo '<br>';
9 if (is_object($b))
10 {
11     print "a";
12 }
13 else
14 {
15     print "b";
16 }
17 ?>
```

然后在网址栏输入

<http://123.206.203.108:10001/random.php?emmm=O:8:%22stdClass%22:1:{i:1;s:3:%22foo%22;}>



可以看到没有显示 error, 那么接下来就按 F5 慢慢刷新, 最终获得 flag (当初没有截屏, 因此这里没有给出最后得出结果的图片)

hgame{&_Is_wonderful!@#}

题外话: 总感觉 F5 慢慢刷新不是正确的做法, 希望知道这道题有没有更快解决的方法。

参考网站

[1]<http://www.cnblogs.com/zwfc/p/5466885.html>

[2]<http://www.jb51.net/article/110312.htm>

[3]<http://blog.csdn.net/persistvonyao/article/details/17585403>

[4]<http://php.net/manual/zh/language.types.object.php>

分数 100:草莓社区-1

描述: flag 在 ../flag.php 中

知识点: LFI

URL: <http://118.25.18.223:10011/>

打开网址, 链接两个, 每个里面一张图, 另外两个链接分别为



草莓社区

[猫片 1](#) [猫片 2](#)

http://118.25.18.223:10011/show_maopian.php?mao=1.jpg

http://118.25.18.223:10011/show_maopian.php?mao=2.jpg

经查询, 输入 http://118.25.18.223:10011/show_maopian.php?mao=../flag.php



```
<?php
$flag="hgame{#Ma0_pi4n_ha0_k4n_ma#}";
```

题外话: 虽然很简单, 但是我对它的原理不是特别理解, 所以这里没有对其进行解释

参考文章

[1]<http://www.vuln.cn/6241>

分数 100:草莓社区-2

描述: flag 在../flag.php 中

知识点: LFI

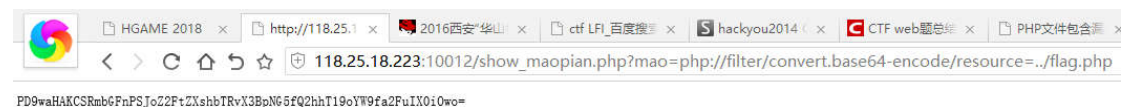
URL: <http://118.25.18.223:10012/>

打开网址,基本上跟上面的题目完全一样(除了两个图片显示不出来,但这不影响做题)

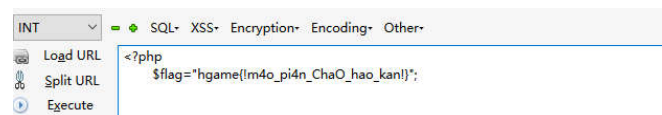
输入网址 http://118.25.18.223:10012/show_maopian.php?mao=../flag.php 发现没有反应,起初以为是网址进行了一些过滤,但是经检查并没有找到源代码(没有源代码的情况下个人感觉进行绕过是非常困难且麻烦的),后来经查询知道了另一种方式[1],

输入 http://118.25.18.223:10012/show_maopian.php?mao=php://filter/convert.base64-encode/resource=../flag.php

得到 base64 编码



经过解密得到 flag



[1] http://www.myhack58.com/Article/html/3/7/2016/79226_2.htm

分数 100:XSS-1

描述: 超简单的 xss

payload 需要在 chrome 和 firefox 上都能使用,且不能有交互。成功后带着 payload 找出题人在线陪聊

QQ1:XXXXXXX

QQ2:XXXXXXX

QQ3:XXXXXXX

知识点: XSS

URL: <http://118.25.18.223:10013/>

Try to alert(1)

```
function charge(input) {  
    input = input.replace(/script/gi, '_');  
    input = input.replace(/image/gi, '_');  
    input = input.replace(/&/g, '_');  
  
    return '<article>' + input + '</article>';  
}
```

try to input something...

很简单的界面,代码也告诉了过滤了什么关键字,虽然学过 html,对 XSS 漏洞有所了解,但是对绕过方面不熟练。这里贴出代码[1][2]

注意这里网站过滤了"("左括号,所以要换成(

Try to alert(1)

```
function charge(input) {  
  input = input.replace(/script/gi, '_');  
  input = input.replace(/image/gi, '_');  
  input = input.replace(/\(/, '_');  
  
  return '<article>' + input + '</article>';  
}
```


请带着payload找fantasyqt(QQ

从联系人那里获得 flag

hgame{#X5s_soo00o_e4sy#}

参考文章

[1]<http://www.freebuf.com/articles/web/20282.html>

[2]<https://zhidao.baidu.com/question/1367931273317263459.html>

分数 100:XSS-2

描述: 不能再简单惹

payload 需要在 chrome 和 firefox 上都能使用, 且不能有交互。成功后带着 payload 找出题人在线陪聊

QQ1:XXXXXXX

QQ2:XXXXXXX

QQ3:XXXXXXX

知识点: XSS

URL: <http://118.25.18.223:10014/>

Try to alert(1)

```
function charge(input) {  
  input = input.replace(/script/gi, '_');  
  input = input.replace(/img/gi, '_');  
  input = input.replace(/image/gi, '_');  
  input = input.replace(/\(/, '_');  
  input = input.replace(/\>/, '_');  
  return '<input value="' + input + '" type="text">';  
}
```

try to input something...

还是跟上一道题目一样, 但是 return 的方式变了。经查询, 找到一个相对应的绕过命令[1], 输入"type=image src onerror="alert(1)

但是这还不行, 因为他过滤了 image 和 (, 所以我们需要将这 2 个关键字进行转换, 将"image"和" ("替换为他们对应 unicode 编码[2],

最终得到"type=image src onerror="alert(1)

Try to alert(1)

```
function charge(input) {  
  input = input.replace(/script/gi, '_');  
  input = input.replace(/img/gi, '_');  
  input = input.replace(/image/gi, '_');  
  input = input.replace(/\(/, '_');  
  input = input.replace(/\>/, '_');  
  return '<input value="' + input + '" type="text">';  
}
```

"type=image src onerror="alert(1)"

请带着payload找fantasyqt(QQ



从联系人那里获得 flag

hgame{#LuCkY_y0u_a1ert_l#}

题外话：说好的在线陪聊呢？2次给完 flag 就不理人了(ノ ￣皿￣)ノ

参考文章：

[1] <https://github.com/cure53/XSSChallengeWiki/wiki/prompt.ml> Level 5

[2] <http://tool.chinaz.com/Tools/Unicode.aspx>

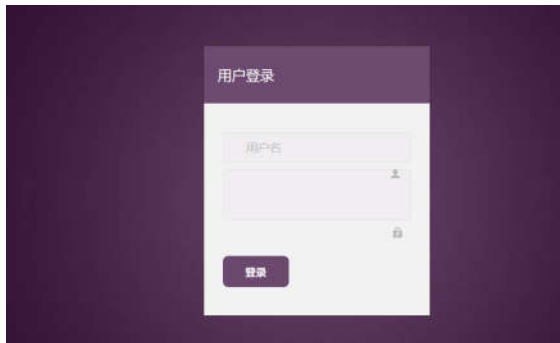
分数 50:最简单的 sql 题

描述：Only admin can get flag!!!!

sql 送分题

知识点：不想给了，太简单了

URL: <http://118.25.18.223:10015/>



登陆界面

起初以为需要是什么绕过啊，准备打开虚拟机中的 sqlmap 以防要用，在打开途中，尝试了下万能密码：账号输入 admin，密码输入 1' or '1'='1，结果直接跳出了 flag

hgame{@s0ng_fen_ti@}

Misc

分数 200：咻咻咻

描述：欢迎使用 flag 售货机，请将手机靠近声波支付感应区。

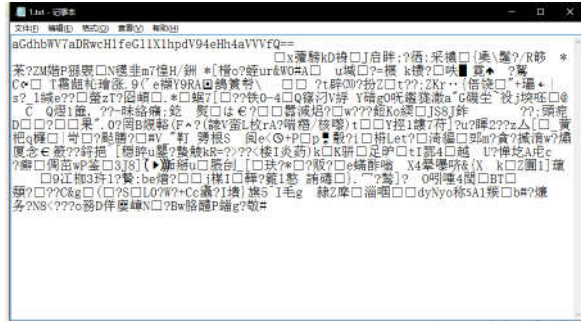
hint 1：粗心的出题人没有把锁上实就去看 ditf 了。

hint 2：音频 LSB 了解一下

URL: <http://p3pqfvzzm.bkt.clouddn.com/xiuxiuxiu.zip>

下载得到一个压缩包，结果 360 检测需要密码，但是打开的时候并不需要密码（原理没查出来），压缩得到一个“咻咻咻.wav”音

频文件，没出 hint2 之前百度看到了一道关于 WAV 的 CTF 题目，尝试用 **waveditor** 查看音频波形是否有隐藏信息，没有任何结果，而且 **binwalk** 也没有任何进展，便放弃了，在出 hint2 后百度 **wav LSB** 出来的结果都是一些很长的分析论文，没有相关的软件，看了前几页，只知道原理跟图片 **LSB** 隐写差不多。之后 **google wav LSB** 得到一个软件叫做 **WavSteg**[1]，下载解压将 **wav** 文件放入目录中，在 **cmd** 中输入 **WavSteg.py -r -s 1.wav -o 1.txt -n 1 -b 1000**，其中为了方便输入命令，将 **咻咻咻.wav** 文件改为 **1.wav**，**1.txt** 是输出文件。其他参数不需要管他。执行完命令，打开 **1.txt**，得到 **base64** 加密



经过解密得到 **flag**

hgame{h4ppy_xiu_Xiu_xxxxIU}

参考文章

[1] <https://github.com/ragibson/Steganography>

分数 150: White cosmos

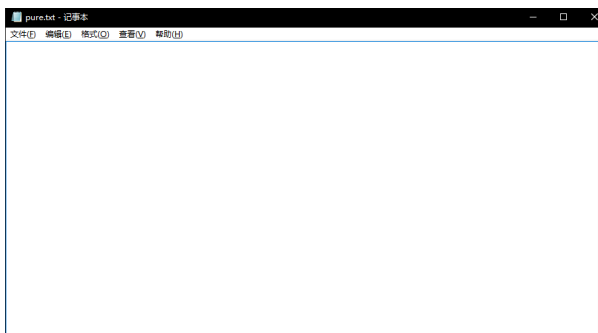
描述: 出题人留下了一张空白的纸条，他说 **flag** 就写在上面。

hint: 用牛奶、碱水之类不易被直接观察到的液体代替墨水，就可以写出“隐形的字”了。

URL: <http://p3pqfvzzm.bkt.clouddn.com/White%20cosmos.zip>

题外话: 其实解题方法说起来非常简单，但是这里仍然想说下当时自己思考的思路，如果觉得太长的话可以直接跳过看最后。

下载得到一个压缩包，里面一个 **pure.txt** 文件



可以看到什么都没有，但是在解压的时候我注意到了文本的大小

密文中间隔着 20，对第 7 个和第 b 个 20 09 09 20 20 解密得到字母 u 或者 v，但是第 c 个是 09 而不是 20，遇到了矛盾，貌似并不是正解。

再说说二进制，用二进制隐藏信息第一个想到的是 ASCII，ASCII 码有 7 位和 8 位（但是实际上第 8 位是用于确定附加的 128 个特殊符号字符、外来语字母和图形符号，所以在用表的时候第 8 位实际都是为 0），解密方式就是抽取 7 位或者 8 位出来转成十进制或者十六进制，然后参照表。但是他的困难仍然是：不知道抽 7 位还是 8 位（甚至还有可能是 4 位），中间是否存在特殊的符号对每个字母加密后的密文进行隔开（举个例子，比如加密明文“AB”，A 的 ASCII 码对应 65，二进制 0100 0001，B 的则为 0100 0010 加密后中间可一会用 09 20 隔开，0 为 09，1 为 20，密文就为 09 20 09 09 09 09 20 20 09 09 09 09 09 20）。

这里尝试了很多种可能，具体过程不说了，这里说下最后是如何确定下来的：这里假设了下 flag 格式为 hgame{XXXX}，那么最后一个单位可能为}，查看了}对应 ASCII 的二进制为 01111101 查看了下倒数 8 位为 20 09 09 09 09 20 09

是不是很相似？那么暂时假设解密方式 每次取 7 位，09 当作 1，20 当作 0，转为十进制作为 ASCII 码，再观察一下，确认中间隔着一个 20，先进行解密，如果不行则进行另一种假设

前 7 位为 09 09 20 09 20 20 20 代表二进制为 1101000，十进制为 104，对应字母为 h，然后第八位 20 作为隔开符号，第 8 位到第 e 位 09 09 20 20 09 09 09 代表二进制为 1101000，十进制为 103，对应字母为 g，以此类推。。。

结果解密进行的很顺利。得到 flag

hgame{Welc0me_2_WhItE_sp4ce}

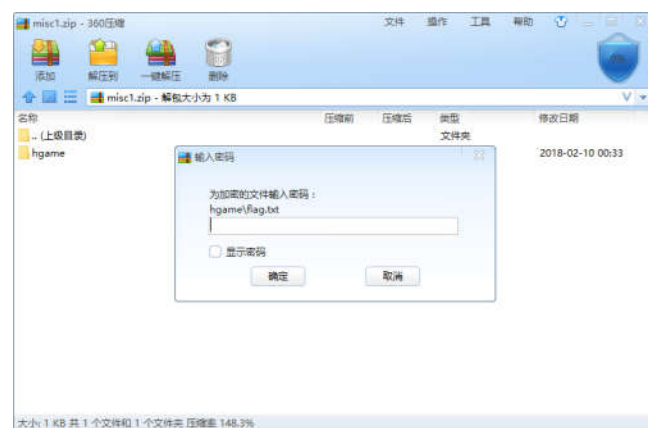
题外话：有时候进行猜测的时候各种的不顺利，却因为切入一个特别的点而找到了突破口。不知道这样写思路会不会太长，可能有点乱，所以最后还是具体举了例子说明了解密方法。

分数 150: easy password

描述：听说你们有人喜欢暴力解题，那么就来暴力一下，测测电脑性能吧。

hint:小写字母+数字

URL: <http://p1kaloi2x.bkt.clouddn.com//hgame/week2/misc1.zip>



打开下载来的压缩包，要求输入密码，看里面的文本名字，推测这应该是唯一的坎，查询百度下载了一个软件叫 Ziperello[1]，安装完后打开软件，读入压缩包点击 NEXT



点击暴力破解，点 next



根据 hint，这里只需要点击数字和小写字母，最长密码长度我这里暂时设置了 10，点击 next



点击开始即可。



最后破解得到压缩密码

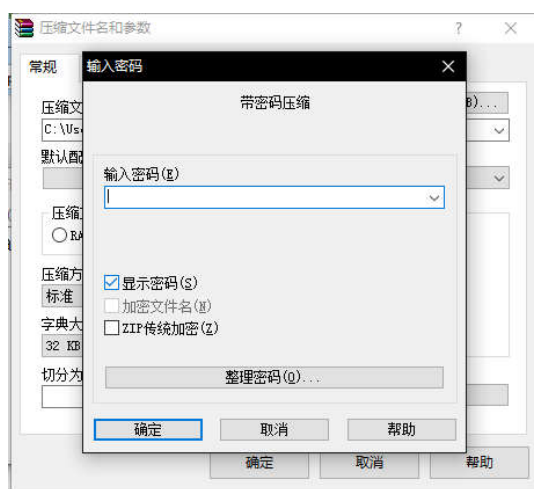


这样就成功打开 flag.txt，得到 flag



题外话：之前打比赛前曾下载了 RAR Password Unlocker 以防万一要破解压缩包密码，但是由于这个压缩包是 ZIP 文件，这个软件没法打开，这里卡了一小段时间。

自己在研究这个软件时，发现这个软件只能针对 ZIP 传统加密



可以看到在设置压缩包密码时有一个 ZIP 传统加密选项，经测试，这个软件只能对付勾选了这个项目的文件（可以理解，毕竟这软件是 2008 年的），否则破解出来密码为 0



而这个 0 的长度是根据当初设置最小长度密码而定，这里进行了测试，得出这个结论。

参考文章：

[1]<http://mydown.yesky.com/pcsoft/414246.html>

分数 200: mysterious file header

描述：

hint 1: order, in order?

hint 2: 怎么才能够运行它呢

hint 3: 四个数字排在一起可能是什么呢，Web 选手一看就知道

hint 4: $4 * 3 * 2 * 1 = 24$

URL: <http://p1kaloi2x.bkt.clouddn.com//hgame/week2/misc2>

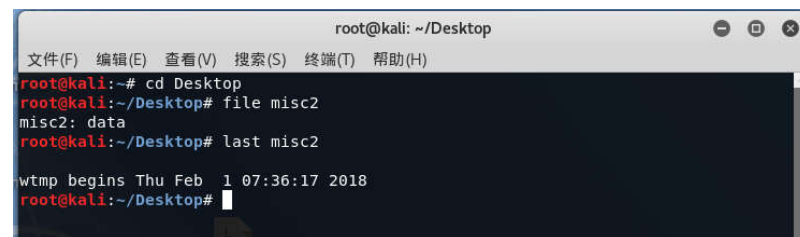
题外话：没有学过反编译，而且由于英语不好，至今不知道 hint 1 想表达什么，尝试方法我就直接写在后面了要不然太乱了

下载得到一个没有后缀名的 misc2 文件，直接扔到 linux 系统桌面，终端输入 file misc2



```
root@kali: ~/Desktop
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~# cd Desktop
root@kali:~/Desktop# file misc2
misc2: data
root@kali:~/Desktop#
```

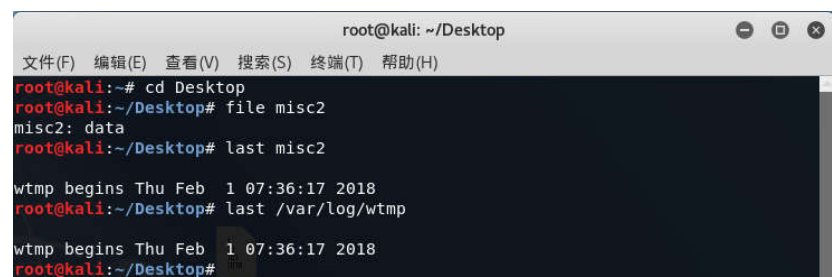
这是数据格式文件[1]，在终端输入 last misc2



```
root@kali: ~/Desktop
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~# cd Desktop
root@kali:~/Desktop# file misc2
misc2: data
root@kali:~/Desktop# last misc2

wtmp begins Thu Feb  1 07:36:17 2018
root@kali:~/Desktop#
```

这是什么玩意？总感觉不对劲，找到另一个 data 类型文件进行测试（系统里面的文件，给出的网站上面有这个）

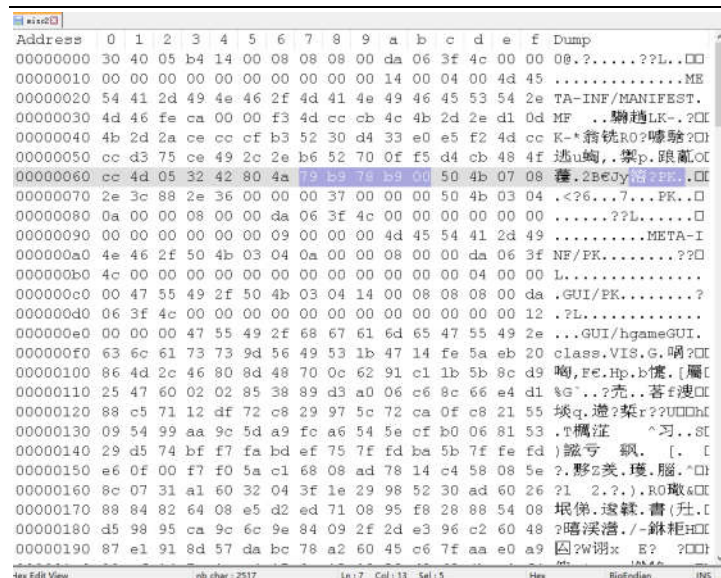


```
root@kali: ~/Desktop
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~# cd Desktop
root@kali:~/Desktop# file misc2
misc2: data
root@kali:~/Desktop# last misc2

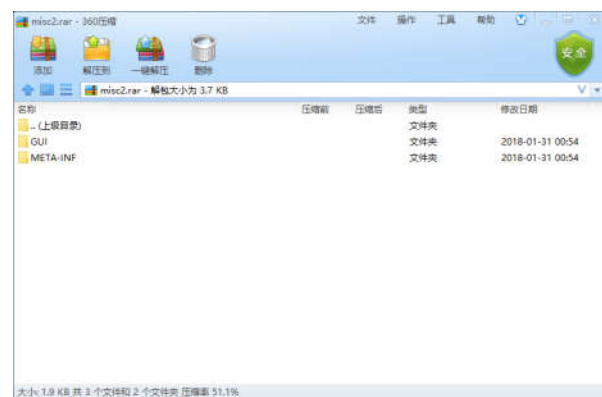
wtmp begins Thu Feb  1 07:36:17 2018
root@kali:~/Desktop# last /var/log/wtmp

wtmp begins Thu Feb  1 07:36:17 2018
root@kali:~/Desktop#
```

居然一样的！不知道发生了什么事情，只能暂时放弃这个思路。实用 notepad 打开，查看里面的十六进制内容



看到了带有 PK 的字眼，由于之前被这东西坑了好几天，马上反映到这可能是个压缩包，将后缀名 **rar** 加上



好像成功了？那么解压里面的文件，得到了 2 个文件 **hgameGUI.class** 和 **hgameGUITest.class** 来自 GUI 文件夹的（META-INF 文件夹中的 MANIFEST.MF 当时解压的时候不知道为什么出错，查询后这个文件夹不是什么关键文件，所以暂时不用理睬）
经查询 **class** 文件是 **java** 编译后的生成文件，需要反编译才能够看到代码[2]，（然而网站给出的 **Java Decompiler** 软件下载过来打开 **class** 文件的时候没有反应，进一步检查原因，提取之前编写过的 **java** 的脚本的 **class** 文件进行测试也不行，所以直接放弃这个软件），再次查询获得在线看上去很强大的反编译网站[3]，选择文件，之后等一会儿就会跳出结果，点击 **de/**，再点击里面的文件就可以得到源代码

Directory Listing For /966f50e5-6764-4cb8-b9bf-cecf87d0f948/ - Up To /		
Filename	Size	Last Modified
de/		Fri, 16 Feb 2018 03:41:27 GMT
hgameGUI.class	2.4 kb	Fri, 16 Feb 2018 03:41:26 GMT

Apache Tomcat/8.5.8

附上 **hgameGUI.class** 和 **hgameGUITest.class** 的结果

```

package GUI;

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextArea;

public class hgameGUI extends JFrame {

    private static final int DEFAULT_WIDTH = 300;
    private static final int DEFAULT_HEIGHT = 200;

    public hgameGUI() {
        super("Welcome to Hgame!");
        this.setSize(300, 200);
        JButton flag1 = new JButton("i\'m flag");
        JButton flag2 = new JButton("i\'m flag, too.");
        JButton flag3 = new JButton("RU kidding me? I\'m the true flag!");
        JButton flag4 = new JButton("UR wrong, I\'m the true flag!");
        JTextArea flagtext = new JTextArea("Want flag? Try upstairs.");
        JPanel flag = new JPanel();
        flag.setLayout(new GridLayout(5, 1));
        flag.add(flag1);
        flag.add(flag2);
        flag.add(flag3);
        flag.add(flag4);
        flag.add(flagtext);
        flag1.addActionListener(flagtext);
        flag2.addActionListener(flagtext);
        flag3.addActionListener(flagtext);
        flag4.addActionListener(flagtext);
        this.add(flag);
    }

    // $FF: synthetic method
    private static void lambda$new$3(JTextArea flagtext, ActionEvent event) {
        flagtext.setText("89");
    }

    // $FF: synthetic method
    private static void lambda$new$2(JTextArea flagtext, ActionEvent event) {
        flagtext.setText("29");
    }

    // $FF: synthetic method
    private static void lambda$new$1(JTextArea flagtext, ActionEvent event) {
        flagtext.setText("54");
    }

    // $FF: synthetic method
    private static void lambda$new$0(JTextArea flagtext, ActionEvent event) {
        flagtext.setText("118");
    }
}

package GUI;

import GUI.hgameGUI;

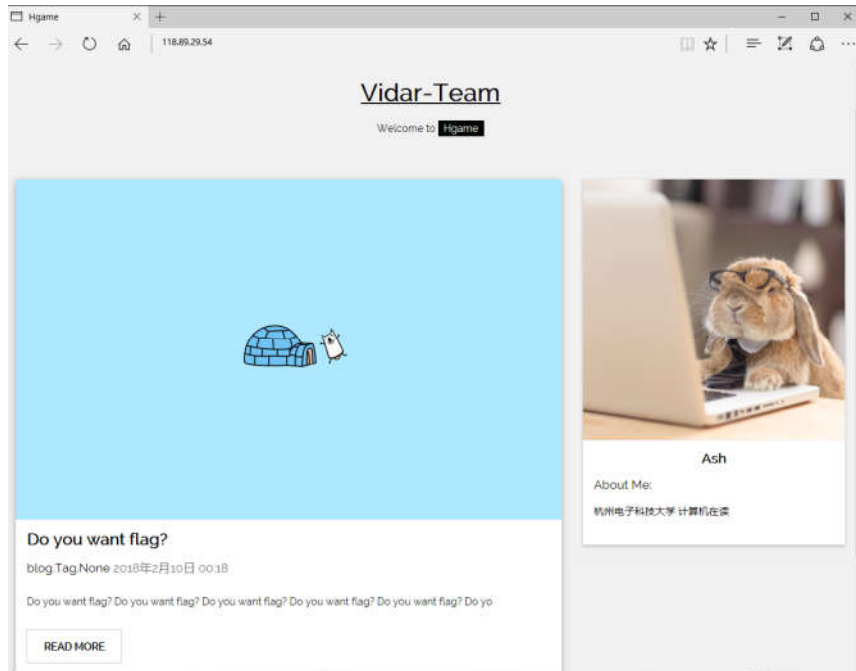
public class hgameGUITest {

    public static void main(String[] param0) {
        // $FF: Couldn't be decompiled
    }

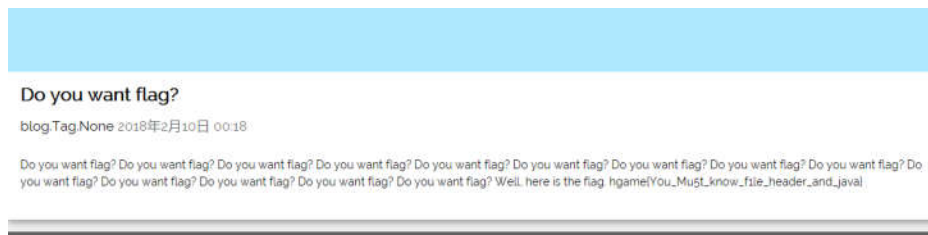
    // $FF: synthetic method
    private static void lambda$main$0() {
        hgameGUI frame = new hgameGUI();
        frame.setDefaultCloseOperation(3);
        frame.setVisible(true);
    }
}

```

可以看到第一个文件中有 4 个数字，那么这 4 个数字是干什么的呢？根据 hint3 和 hint4，推测可能为网址和但是网址具体不知道是什么，需要 4 个数字随机排序得到。经过测试得到能访问的网站 **118.89.29.54**



访问网页，得到 flag



题外话：这玩意卡了 1 天半左右，主要问题就在那 4 个数字，其实根据 hint 我们需要让程序运行起来。然而这个网页编译出来的代码无法运行。

```

15
16 public HgameGUI() {
17     super("Welcome to Hgame!");
18     this.setSize(300, 200);
19     JButton flag1 = new JButton("i\'m flag");
20     JButton flag2 = new JButton("i\'m flag, too.");
21     JButton flag3 = new JButton("RU kidding me? I\'m the true f");
22     JButton flag4 = new JButton("UR wrong. I\'m the true flag!");
23     JTextArea flagtext = new JTextArea("Want flag? Try upstairs.");
24     JPanel flag = new JPanel();
25     flag.setLayout(new GridLayout(5, 1));
26     flag.add(flag1);
27     flag.add(flag2);
28     flag.add(flag3);
29     flag.add(flag4);
30     flag.add(flagtext);
31     flag1.addActionListener(flagtext);
32     flag2.addActionListener(flagtext);
33     flag3.addActionListener(flagtext);
34     flag4.addActionListener(flagtext);
35     this.add(flag);
36 }
37

```

可以看到程序时报错的，而且编译器提示进行修复运行起来还是错的

```
private static final int DEFAULT_WIDTH = 300;
private static final int DEFAULT_HEIGHT = 200;

public HgameGUI() {
    super("Welcome to Hgame!");
    this.setSize(300, 200);
    JButton flag1 = new JButton("I\'m flag");
    JButton flag2 = new JButton("I\'m flag, too.");
    JButton flag3 = new JButton("RU kidding me? I\'m the true flag!");
    JButton flag4 = new JButton("UR wrong, I\'m the true flag!");
    JTextArea flagtext = new JTextArea("Want flag? Try upstairs.");
    JPanel flag = new JPanel();
    flag.setLayout(new GridLayout(5, 1));
    flag.add(flag1);
    flag.add(flag2);
    flag.add(flag3);
    flag.add(flag4);
    flag.add(flagtext);
    flag1.addActionListener((ActionListener) flagtext);
    flag2.addActionListener((ActionListener) flagtext);
    flag3.addActionListener((ActionListener) flagtext);
    flag4.addActionListener((ActionListener) flagtext);
    this.add(flag);
}

// $FF: synthetic method
UI.hgameGUI > hgameGUI >
GUI (run) x
run:
Exception in thread "main" java.lang.ClassCastException: javax.swing.JTextArea cannot be cast to java.awt.event.ActionListener
    at GUI.hgameGUI.<init> (HgameGUI.java:32)
    at GUI.hgameUITest.main (HgameUITest.java:5)
C:\Users\kikipp\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1
构建失败 (总时间: 1 秒)
```

经过查询得出将代码进行以下更改[4]

```
JPanel flag = new JPanel();
flag.setLayout(new GridLayout(5, 1));
flag.add(flag1);
flag.add(flag2);
flag.add(flag3);
flag.add(flag4);
flag.add(flagtext);
flag1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        lambda$new$0(flagtext,e);
    }
});
flag2.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        lambda$new$1(flagtext,e);
    }
});
flag3.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        lambda$new$2(flagtext,e);
    }
});
flag4.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        lambda$new$3(flagtext,e);
    }
});
//flag1.addActionListener((ActionListener) flagtext);
//flag2.addActionListener((ActionListener) flagtext);
//flag3.addActionListener((ActionListener) flagtext);
//flag4.addActionListener((ActionListener) flagtext);
```

改动还是比较大的虽然这样无法确定每个按钮按下所触发的事件是否跟原来一样，但是在当时只知道不完整的代码的情况下，只能这样了。

转到 hgameUITest


```
package GUI;

import GUI.hgameGUI;

public class hgameGUITest {

    public static void main(String[] param0) {
        // $FF: Couldn't be decompiled
    }

    // $FF: synthetic method
    private static void lambda$main$0() {
        hgameGUI frame = new hgameGUI();
        frame.setDefaultCloseOperation(3);
        frame.setVisible(true);
    }
}
```

学过 java 的应该可以看出，这里主程序中没有执行命令，这里我们主动加上,并运行

```
package GUI;

import GUI.hgameGUI;

public class hgameGUITest {

    public static void main(String[] param0) {
        // $FF: Couldn't be decompiled
        lambda$main$0();
    }

    // $FF: synthetic method
    private static void lambda$main$0() {
        hgameGUI frame = new hgameGUI();
        frame.setDefaultCloseOperation(3);
        frame.setVisible(true);
    }
}
```



可以看到程序可以成功运行（在实际操作中，我曾经尝试调试找到错误但是失败了）点击 4 个按钮可以分别得到 4 个数字 89, 29, 54, 118。

再次查看代码，确认 flag 不在源代码中。转会题目看 hint

hint 3: 四个数字排在一起可能是什么呢，Web 选手一看就知道

hint 4: $4 * 3 * 2 * 1 = 24$

不知道 hint4 什么意思，直接百度


```

public class hgameGUI extends JFrame
{
    private static final int DEFAULT_WIDTH = 300;
    private static final int DEFAULT_HEIGHT = 200;

    public hgameGUI() {
        super("Welcome to Hgame!");
        this.setSize(300, 200);
        final JButton flag1 = new JButton("i'm flag");
        final JButton flag2 = new JButton("i'm flag, too.");
        final JButton flag3 = new JButton("RU kidding me? I'm the true flag!");
        final JButton flag4 = new JButton("UR wrong, I'm the true flag!");
        final JTextArea flagtext = new JTextArea("Want flag? Try upstairs.");
        final JPanel flag5 = new JPanel();
        flag5.setLayout(new GridLayout(5, 1));
        flag5.add(flag1);
        flag5.add(flag2);
        flag5.add(flag3);
        flag5.add(flag4);
        flag5.add(flagtext);
        flag1.addActionListener(event -> flagtext.setText("118"));
        flag2.addActionListener(event -> flagtext.setText("54"));
        flag3.addActionListener(event -> flagtext.setText("29"));
        flag4.addActionListener(event -> flagtext.setText("89"));
        this.add(flag5);
    }
}

```

这个工具叫 `procyon-decompiler`[5] (然而当时网站下过来的是 `jar` 文件, 双击没有反应, 但是自己有些其他 `jar` 文件都可以运行, 查询网上暂时没有解决方案, 所以在命令符里面运行, 输入 `java -jar procyon-decompiler-0.5.30.jar hgameGUI.class` 即可) 可以看到源代码没有什么 `lambdanew0` 这些东西。

所以最后的灵感来源来自哪里呢? 放图吧



不得不说百度真是神奇的地方。。两个灵感来源都是它

参考文章:

- [1]<https://zhidao.baidu.com/question/456469364229536165.html>
- [2]<https://jinqyan.baidu.com/article/d169e18677e87b436611d81f.html>
- [3]<http://javare.cn/>
- [4]<http://blog.csdn.net/hbjhappy/article/details/46882743>
- [5]<http://download.csdn.net/download/ueu2715/10176201>

Crypto

分数 150:easy rsa

描述: 真的很简单的啦

URL: <https://pastebin.com/yB5SQdhn>

```

p = random_prime(2**1024)
q = random_prime(2**1024)
N = p * q
e = 65537

flag = "xxxxxxxxxxxxxxxxxxxx..."
m = int(flag.encode('hex'), 16)
c = pow(m, e, N)
print "N: " + str(N)
print "e: " + str(e)
print "c: " + str(c)
print "h: " + str(p+q)

...
output:
N:
1038511285350354528353459449801400216330281919254288135962901617865181459339453822393733674125477453748418677846543570433509186453439897628509042367641638605796280506469598857872127102
1836244935120824154200938246665792571840648519258635324070387081531738138451636079303880672328523875536502775513804305125108594627576700137327744464365102621228492597080893934812645457
11565234024195713041049572386007243341480416299554565488918506092454861627134347488019688384580087306252753880774307836121161612450376309844794007213153187554046570932068258835721493934
81806067157147431981573823960963614146686202457034323040706001
e: 65537
c:
437197606589433890314975885075127128451240983808800709698046359245834252204150660135884882257934880338033907956567188535876921776874898534795022472667719240357498052992696025272720367
88769904108885493823764984982805025952459173246366939243972669582338728034363614943062106220697944193226897767645789368465460202024200438535770983989035642434091720020123447189714932941
2039532014211438168566024105162077029048069034351631913482778674758139857656850331738272019703969084393602184095626927532572350848935484496658484866819312588553293845344222453379024867
1083002562017871712806386748477524316776702973435067495735891
h:
211473031829143387075248428327012971987132927708382843078496747812049686092488080611907415709990988195782979354578429516721486464408046484700638962800675832747784587010153523205480959
518942953437786700176784903631911934300110277162348447359625868267531918956816603020094562890253995876322745344347924616750
...

```

打开为一段 python 代码，写着加密方式，去查询具体了解 RSA 算法[1]得知 RSA 算法中一共需要几个参数：（简单为主，有些可能被忽略）

p,q 两个质数

$N=p*q$

c 为密文，m 为明文

如何加密：密文=明文的 e 指数 mod N （即 $c=pow(m,e,N)$ ），e 为公钥

如何解密：明文=密文的 d 指数 mod N （即 $m=pow(c,d,N)$ ），d 为密钥

按照目前已知的情况下，根据一般的题目涉及到的 RSA 思路，都是知道 N 通过某种软件求出 p，q，然后求出 $(p-1)*(q-1)$ ，根据 e 的值以此推出密钥 d，最后进行解密。

那么先按照这个思路进行，尝试用 yafu,RSATool（用 RSATool 解，闪退了 2 次），在线网站解[2]，然而 N 的数字过大，导致了运算过大无法求出 p，q，会去观察，题目有给出 $p+q$ 的值，在当时，知道 $p+q=h$ 和 $p*q=N$ 的情况下怎么求 p 和 q，第一个想法是一元二次方程， $q=N/p$ ，将两个式子化简可以得到 $p^2-hp+N=0$ ，只要求出 Δ 就行了，然而实际运行下来 delta 长度也很大（610 多位），下来了大整数计算器过来然而里面并没有求根的功能。尝试了挺长时间最终放弃。

最后再次换思路，我们求 p 和 q 最终是为了什么？为了求 d。那 d 跟什么值有关？ $(p-1)*(q-1)$ 和 e 有关，那么可以尝试解出 $(p-1)*(q-1)$ 的值， $(p-1)*(q-1)=pq-p-q+1=N-h+1$ ，正好在不用知道 p 和 q 的情况知道，通过 python 完成解出明文

```

1 import gmpy2
2 N=103851128535035452835345944980140021633028191925428813596290161786518145933945382239373367
3 h=211473031829143387075248424832701297198713292770838284307849674781204968609248808096119074
4 e=65537
5 c=4371976065894338903149758850751271284512409838088007096980463592458342522041506601358848
6
7 phi_n = gmpy2.mpcz(N-h+1)#(p - 1) * (q - 1)
8 d = gmpy2.invert(e, phi_n)
9 print(d)
10 print pow(c, d, N)

```

```

C:\Users\...\Desktop\java脚本\python>python RSA.py
579732599516955619265625737689919082570812884567772883320914226919109261790041016407273992504814978783566378273365879616
567639022840683361672037650518286385390019259851977164373181210664796835520605970929618585799512950126003650773925747796
071701691844094144567664977239189220919335030060071700599242115186861227403801618222416661349566608811844346080574368566
4888592095059142543921001091238282866716907939297599192658117601167227975593187938209839565673572680673266449386670743
045135561957718165113121618784685301355459056235890694432262507852084420154456564100269904702429174191721491495059858475
7498477800664133
733148629355484306968869284663023756047697324501468694392866623711149044556651368755667115954822150035813344725483906213
073228455811142727060402997974268832835225447146787463755604044325171122894595339673436483775922787877842772415340354382
39445829403403933010376834233726997228461728071629720620824279384064
C:\Users\...>python>

```

可以看到结果倒数第三行开始就是明文了。但是这里还没有结束

```

p = random_prime(2**1024)
q = random_prime(2**1024)
N = p * q
e = 65537

flag = "xxxxxxxxxxxxxxxxxxxx..."
m = int(flag.encode('hex'), 16)
c = pow(m, e, N)
print "N: " + str(N)
print "e: " + str(e)
print "c: " + str(c)
print "h: " + str(p+q)

```

Big Integer Calculator v1.13

CLEAR ALL

X 686761D657B706869SF6973SF696D706F7274616E74SF746F6F217D0000000000000000
CL

Y 0
CL

Z 0
CL

A 0
CL

B 0
CL

LCM 0
CL

Rem 0
CL

GCD 0
CL

Ans 0
CL

To X

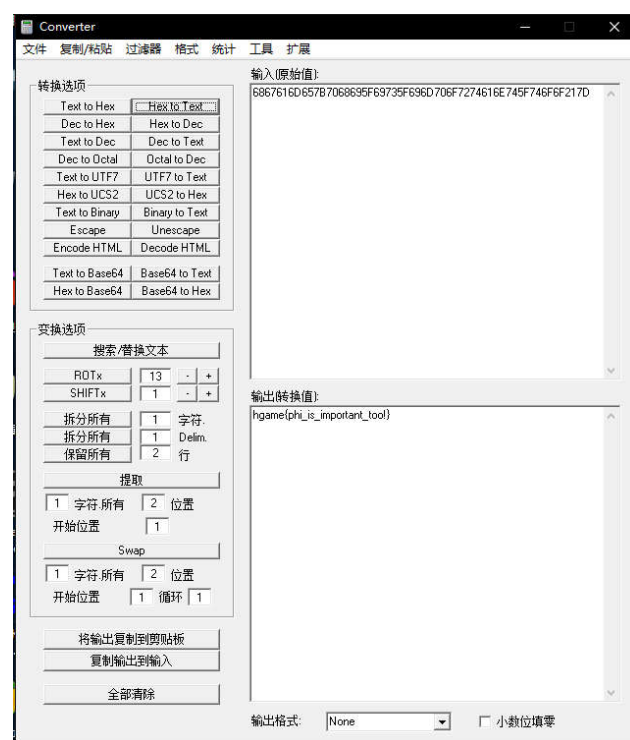
X-Y X+Y X*Y X/Y A*X+B*Y X*A+Y*B+Z X^Y MOD Z Ans =Y/X MOD Z

X / Prime(X) X^n X^(1/n) GCD(X,Y) X*Y*Z*A*B X^A*Y^B MOD Z

Base 2 8 10 16 36 60 64 256 About Exit

Bits: x- 1023 y- 0 z- 0 a- 0 b- 0 ans- 0

用 converter 转换为明文得到 flag



[1] <http://www.freebuf.com/articles/rookie/154183.html>

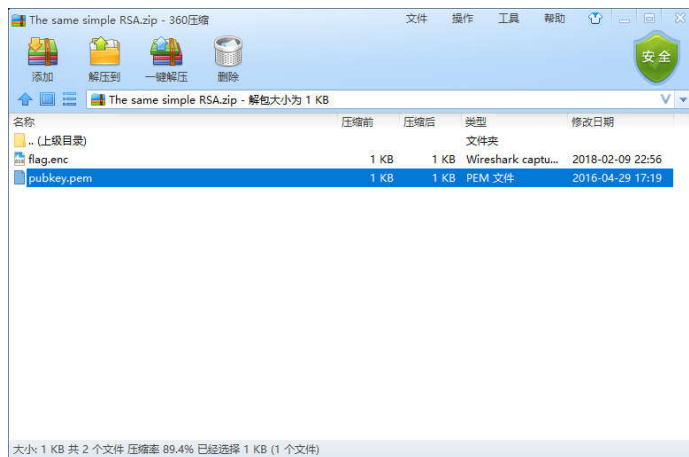
[3] <https://www.52pojie.cn/forum.php?mod=viewthread&tid=490769>

链接在文章结尾

分数 150:The same simple RSA

描述: do you know openssl?

URL: <http://p3xihyup6.bkt.clouddn.com/The%20same%20simple%20RSA.zip>



两个文件，拉到 kali linux 桌面，终端输入 `openssl rsa -pubin -text -modulus -in pubkey.pem`

```
root@kali: ~/Desktop
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~# cd Desktop
root@kali:~/Desktop# openssl rsa -pubin -text -modulus -in pubkey.pem
Public-Key: (256 bit)
Modulus:
    00:c2:63:6a:e5:c3:d8:e4:3f:fb:97:ab:09:02:8f:
    1a:ac:6c:0b:f6:cd:3d:70:eb:ca:28:1b:ff:e9:7f:
    be:30:dd
Exponent: 65537 (0x10001)
Modulus=C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD20Q/+5erCQKP6qxsC/bNPXDr
yigb/+L/vjDdAgMBAAE=
-----END PUBLIC KEY-----
root@kali:~/Desktop#
```

其中 Modulus 的值就是 n 的值，Exponent 的值就是 e 的值。

下载 msieve，分解 n 的值接触 p，q[1]，cmd 输入

`msieve153 -v 0xC2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD`（注意 openssl 分析出来的 N 为十六进制所以数字前面要加 0x）

```
E:\CTF\RSA\msieve153>msieve153 -v 0xC2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD

Msieve v. 1.53 (SVN 1005)
Fri Feb 16 16:08:59 2018
random seeds: 992f9e7c 7a7594f7
factoring 87924348264132406875276140514499937145050893665602592992418171647042491658461 (77 digits)
searching for 15-digit factors
commencing quadratic sieve (77-digit input)
using multiplier of 1
using generic 32kb sieve core
sieve interval: 12 blocks of size 32768
processing polynomials in batches of 17
using a sieve bound of 924097 (36332 primes)
using large prime bound of 92409700 (26 bits)
using trial factoring cutoff of 26 bits
polynomial 'A' values have 10 factors
restarting with 18293 full and 198086 partial relations

36446 relations (18293 full + 18153 combined from 198086 partial), need 36428
sieving complete, commencing postprocessing
begin with 216379 relations
reduce to 52409 relations in 2 passes
attempting to read 52409 relations
recovered 52409 relations
recovered 43101 polynomials
attempting to build 36446 cycles
found 36446 cycles in 1 passes
distribution of cycle lengths:
  length 1 : 18293
  length 2 : 18153
largest cycle: 2 relations
matrix is 36332 x 36446 (5.2 MB) with weight 1077931 (29.58/col)
sparse part has weight 1077931 (29.58/col)
filtering completed in 3 passes
matrix is 26413 x 26477 (4.1 MB) with weight 869029 (32.82/col)
sparse part has weight 869029 (32.82/col)
saving the first 48 matrix rows for later
matrix includes 64 packed rows
matrix is 26365 x 26477 (2.8 MB) with weight 655945 (24.77/col)
sparse part has weight 475543 (17.96/col)
commencing Lanczos iteration
memory use: 2.8 MB
lanczos halted after 419 iterations (dim = 26361)
recovered 13 nontrivial dependencies
p39 factor: 275127860351348928173285174381581152299
p39 factor: 319576316814478949870590164193048041239
elapsed time 00:00:06
```

可以看到最后面两行 p39 factor 就是 p 和 q

用 python 脚本生成 private.pem 并将其放入 linux 桌面中

```
import math
import sys
from Crypto.PublicKey import RSA

keypair = RSA.generate(1024)

keypair.p = 275127860351348928173285174381581152299
keypair.q = 319576316814478949870590164193048041239
keypair.e = 65537

keypair.n = keypair.p * keypair.q
Qn = Long((keypair.p-1) * (keypair.q-1))

i = 1
while (True):
    x = (Qn * i) + 1
    if (x % keypair.e == 0):
        keypair.d = x / keypair.e
        break
    i += 1

private = open('private.pem', 'w')
private.write(keypair.exportKey())
private.close()
```

终端输入 openssl rsautl -decrypt -in flag.enc -inkey private.pem -out flag.dec

执行完后生成 flag.dec，打开它就可以拿到 flag



参考文章:

[1] <https://sourceforge.net/projects/msieve/>

分数 200:Caesar&&Caesar

描述: https://www.wikiwand.com/en/Vigen%C3%A8re_cipher

flag 是书的名字 空格部分用_替换 不要忘记 hgame{}

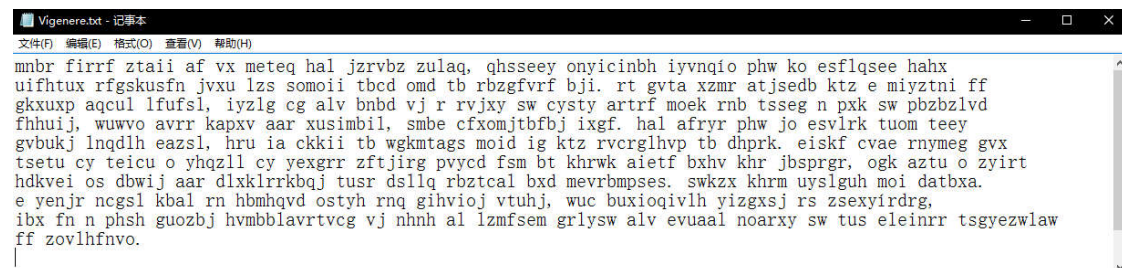
PS:校内的同学提交 writeup 的时候请给出简要的分析过程

URL: <http://p3xlhyup6.bkt.clouddn.com/Vigenere>

下载过来为一个无后缀的文件，放到 kali linux 查看文件类型

```
root@kali:~/Desktop# file Vigenere
Vigenere: UTF-8 Unicode text, with CRLF line terminators
```

为 txt 文本，那么直接在文件名后面加 txt 后缀并打开



可以看到密文，根据文件名字可以推出这是维吉尼亚密码，需要密钥才可以解密，然而我们现在并没有密钥

经过查询有一种方法可以在不知道密钥情况下破解，代码如下网址上有，但是还需要另外写一些内容，这里说明下在 `public static void main(String[] args) {}` 括号中写入：

String

```
b="MNBFRIRRFZTAIIAFVXMETEQHALJZRVBZZULAQQHSSEYONYICINBHIYVNQIOPHWKOESFLQSEEHAXUIFHTUXRFGSKUS
FNJXULZSSOMOIITBCDOMDTBRBZGFVRFBJIRTGVTAZMRATJSEDBKTZEMIZTNIFFGKXUXPAQCULLFUFSLIYZLGCALVBNB
DVJRRVJXYSWCYSTYARTRFMOEKRNBTSSEGNPKXSWPBZBLVDHFHUIJWUWVOAVRRKAPXVAARXUSIMBILSBECFXOMJTBFB
JIXGFHALAFRYRPHWJOESVLRKTUOMTEEYGVBUKJLNQDLHEAZSLHRUIACKIITBWGKMTAGSMOIIDIGKTZRVCRLHVPTBDHPR
KEISKFCVAERNYMEGGVXTSETUCYTEICUOYHQZLLCYEXGRRZFTJIRGPVYCDFSMBTKHRWKAITFBXHVKHRJBSPRGROGKAZT
UOZYIRTHDKVEIOSDBWIJAARDLXKLRR"; //这是密文
```

```
int c=Friedman(b);
```

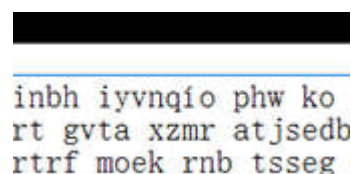
```
decryptCipher(c,b);
```

使用方面注意几点：

- 1.必须大写
- 2.空格。逗号和句号全部去掉
- 3.非字母全部去掉或者改掉

这是我做了几个小时得出的经验，因为之前一直在报错。

为什么我提到了第三点，如果仔细观察文本的话，会注意到一下内容



位于第一行，这个单词的倒数第二位不是字母。后来才发现的，将其改为 I，文中还有多处出现这种情况。

密文全部处理后，运行起来。



可以看到下面的明文，看上去很乱，将他复制出来进行手动分开单词，这是当时解出来时的部分明文

42

vnodmbnafmfhbejfijtxpnqumscvavolgnhjgdtzlanybzfybrnotprrwqvawastoremember that distant afternoon when his father took him to discover Ice at that time Macondo was a village of twenty adobe houses built on the bank of a river of clear water that ran along a bed of polished stones which were white and enormous like prehistoric egg sthe world was so recent that many things lacked names and in order to indicate them it was necessary to point every year during the month of march a family of fraged gypsies would set up their tents near the village and with a great uproar of pipes and kettled.

中间有一段很顺的明文，查询得到是《百年孤独》英文名为 One Hundred Years of Solitude，根据题目提示加上 hgame{}和_，得到 flag

hgame{One_Hundred_Years_of_Solitude}

题外话：如果那天运气不好搜到的书名全是小写不知道该怎么办呢。。。。

参考文章

[1] http://blog.csdn.net/white_idiot/article/details/61201864

分数 200:violence

描述：头铁一点其实也挺好的

URL: <http://p3xlhyup6.bkt.clouddn.com/Affine.py>


```

a = ?
b = ?
m = ?

flag = "hgame{" + m + "}"

cipher = ''
for i in m:
    if 96 < ord(i) < 123:
        cipher += chr(a * (ord(i) + b - 97) % 26)
    else:
        cipher += i

print cipher.encode('hex')

# https://www.wikiwand.com/en/Affine_cipher flag是一个有意义的句子
# cipher = 1917090506070905195f07065f06031505195f035f0a07065f170c5f1407170205101105

```

打开后为 python 代码

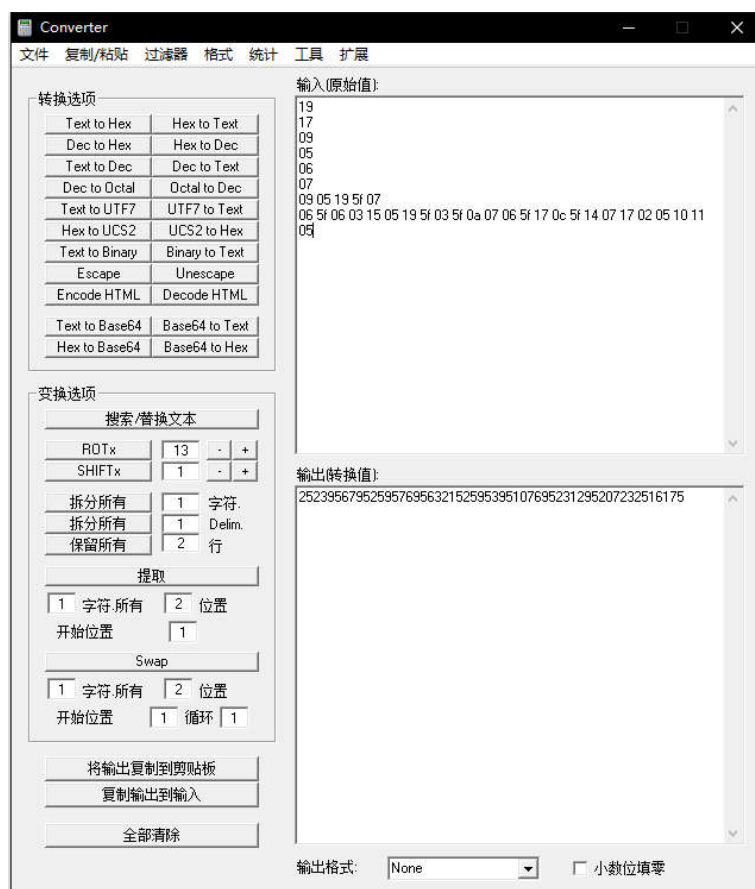
题目提示为仿射密码，仿射密码需要一个密钥 (a, b)，加密方式为把字母转换为数字 n (a 为 0, b 为 1, 以此类推)，

然后每个数字经过以下处理

密文=(n*a+b)mod26

然后连在一起就可以了

然而这里我们并不知道密钥多少，经过查询得到 python 脚本[1]，观察题目，发现密文又经过 ASCII 转码编程十六进制才是我们看到的 cipher 密文，因此在运行脚本前，我们需要将其恢复成原来的只经过仿射密码加工的密文,将密文每两个进行隔开，放入 converter 转换为十进制



将结果每两个数字隔开，并转换为字母（之前说过 0 为 a, b 为 1, c 为 2, 以此类推），推出密文为

zxjfhjfhzhgddvfdzdnkhgxmuhxcfrfnk

接下来写入脚本中并运行

```

# coding:utf8

# 密文
Cs = r"zxjfgghjfhzggdvfzdnkhgxmuhxcfrfnk"

charTable = [ chr(c) for c in range(97,123)]
frequencyTable = [4, 19, 14, 0, 13, 8, 17, 18, 7, 3, 11, 2, 20, 12, 15, 24, 22, 6, 1, 21, 10]

# 删除预留的标点
def del_point(c):
    if c in [' ', ',', '.', ';', '\'', '?', '!']:
        return False
    return True

def get_int_by_char(c):
    return charTable.index(c)

def get_char_by_int(i):
    return charTable[i]

# 最大公约数
def gcd(a, b):
    if a < b:
        a, b = b, a
    while b != 0:
        temp = a % b
        a = b
        b = temp
    return a

# 排序
def sort_by_value(d):
    items = d.items()
    backitems = [ [v[1],v[0]] for v in items ]
    backitems = sorted(backitems, reverse = True)
    return [ backitems[i][1] for i in range(0,len(backitems))]

# 获取k3
def get_k3(k1, k2):
    for k3 in range(0,26):
        if k3 * k1 % 26 == 1:
            return k3

# 判断一个数是否是整数
def is_int(n):
    int_n = int(n)
    return n * n == int_n * int_n

# 仿射解密过程
def FsJM(c, k1, k2):
    k3 = get_k3(k1, k2)

```

```

C:\WINDOWS\system32\cmd.exe

C:\Users\user\Desktop\java脚本\python>python 1.py
(7, 3, 'sometimesittakesaubitofviolenceub')
(19, 9, 'uyaitaaiuettmciumsletyhreybizkisl')
(23, 19, 'yqmwneemwyennoiwyocdenqlreqxwbswcd')
(1, 19, 'geqmnoqmgonnkcmgkuronetboejmxymur')

```

可以看到结果中有一个有意义的句子，

sometimesittakesaubitofviolenceub

将其以 hgame{}形式提交，发现不对，试着将其分开，却发现句子貌似有一点不通

sometimes it takes a ubit of violenceub

总感觉 ubit 的 u 和 violenceub 的 ub 不该存在，将其删除，单词之间用_隔开，试着提交上去，结果成功了，最后的 flag 为

hgame{sometimes_it_takes_a_bit_of_violence}

参考文章：

[1] <https://www.jianshu.com/p/128b203dfdd8>