Week1-Thefth

web:

1.谁吃了我的flag:

根据vim以及diScl0Sure(泄露)的hint知道本题应该是vim泄露，于是百度搜索vim泄露——

一、vim备份文件

默认情况下使用Vim编程，在修改文件后系统会自动生成一个带~的备份文件，某些情况下可以对其下载进行查看；

eg:index.php普遍意义上的首页，输入域名不一定会显示。 它的备份文件则为index.php~

二、vim临时文件

vim中的swp即swap文件，在编辑文件时产生，它是隐藏文件，如果原文件名是submit，则它的临时文件

.submit.swp。如果文件正常退出，则此文件自动删除。

由于关机导致文件意外退出而生成了临时文件，于是输入URL:

http://118.25.111.31:10086/.index.html.swp

,果然自动下载了一个文件，打开发现flag:hgame{3eek_diScl0Sure_fRom+wEbsit@}

2.换头大作战:

打开题目点击submit得到提示:

想要flag嘛：submit

request method is error.I think POST is better

于是编辑网页源代码改method为post方式，再次点击submit得到进一步提示:

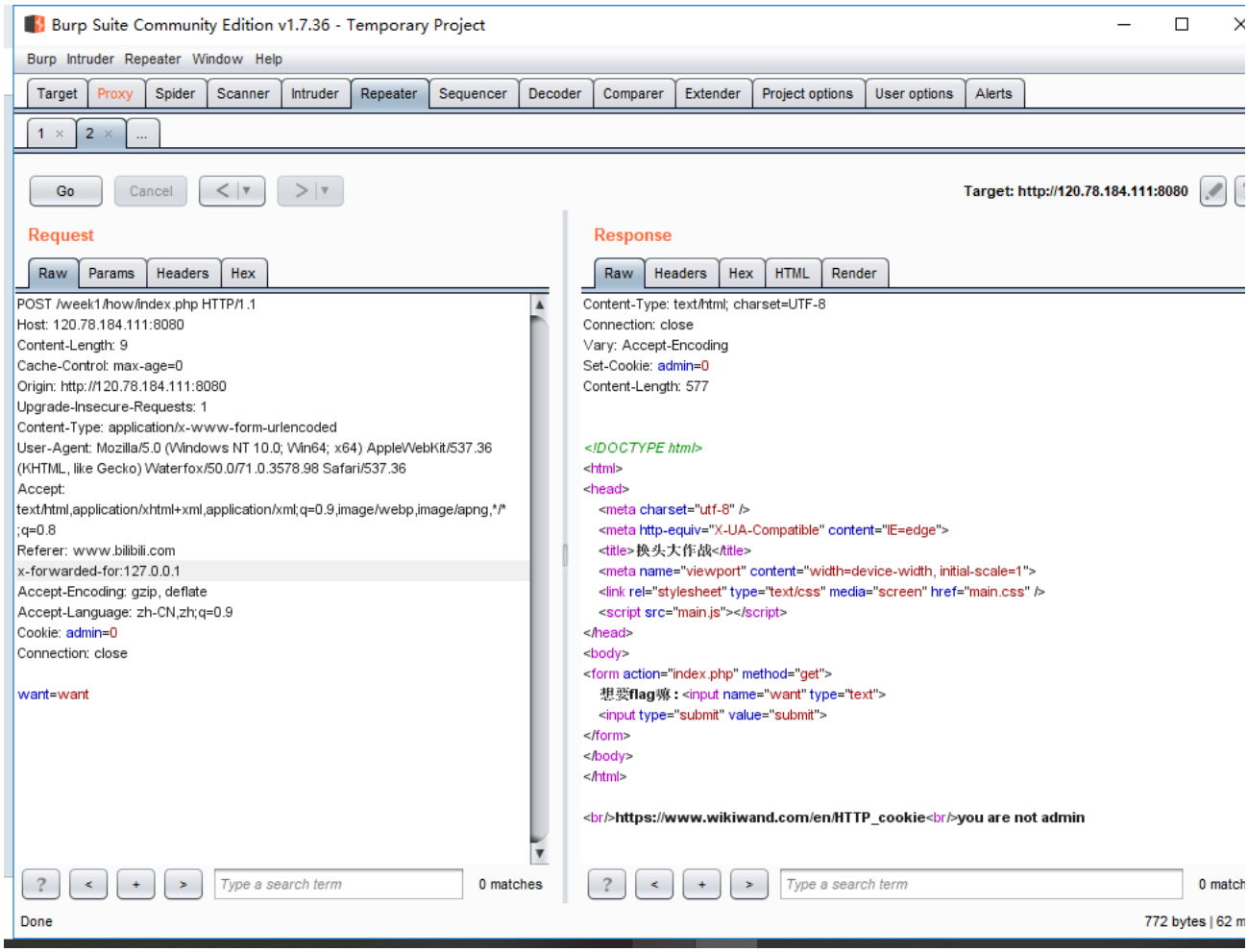
想要flag嘛：submit

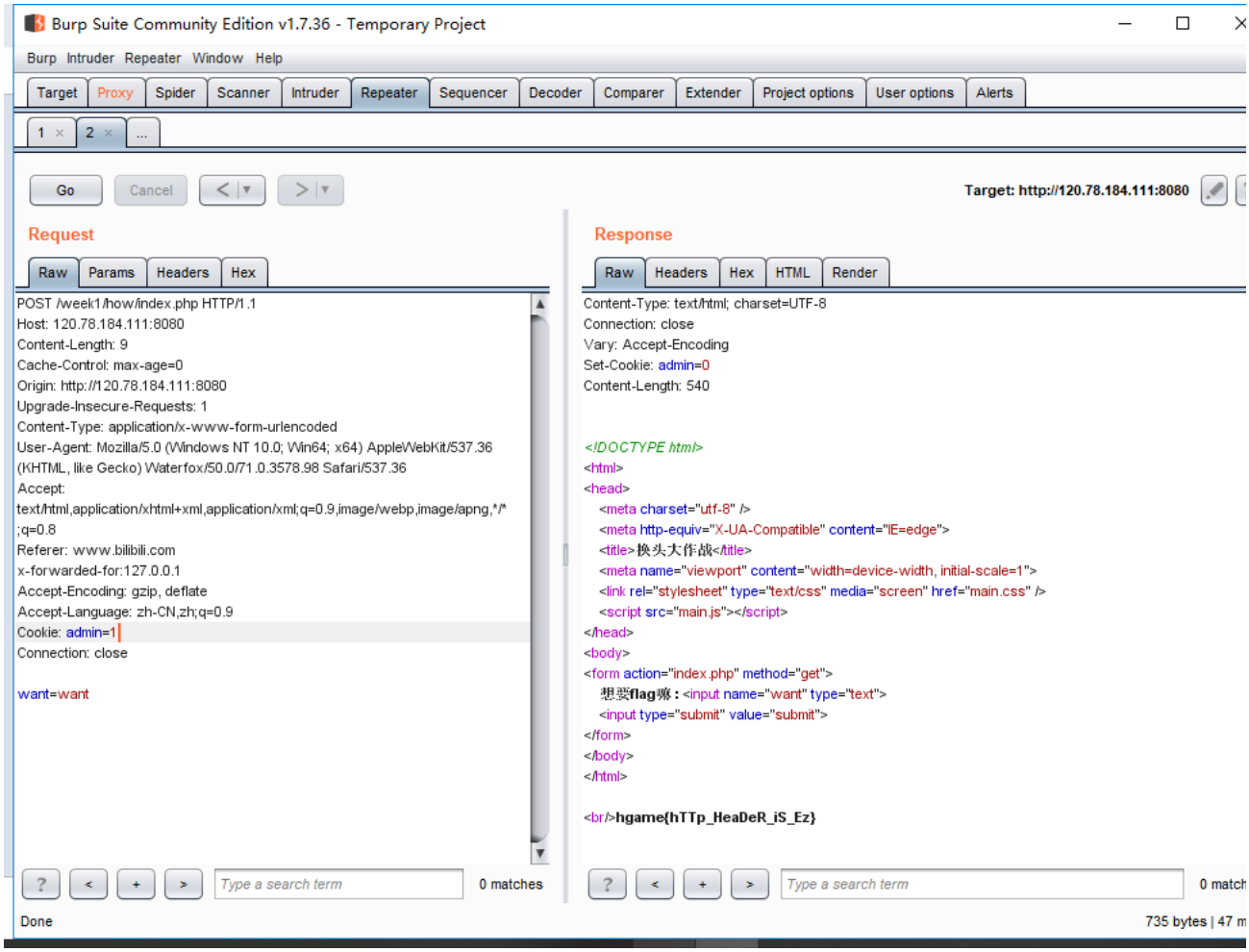
https://www.wikiwand.com/en/X-Forwarded-For
only localhost can get flag

于是使用burgsuite进行抓包（现学现卖设置历经艰辛... 这里注意需要输入want进行抓包传入post参数），发送到repeater中根据localhost的提示改变ip，http中加入一条**x-forwarded-for: 127.0.0.1**语句，改动ip为127.0.0.1，即本机:

看到下一步提示需用waterfox浏览器，于是改动chrome为waterfox/50.0:

下面两步如法炮制：





最后改动cookie值为非0，得到flag:hgame{hTTp_HeaDeR_iS_Ez}

3.very easy web:

打开题目，发现php代码：

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

if(strpos("vidar",$_GET['id'])!==FALSE)
    die("<p>干巴爹</p>");

$_GET['id'] = urldecode($_GET['id']);
if($_GET['id'] === "vidar")
{
    echo $flag;
}
highlight_file(__FILE__);
?>
```

分析得出：需要比较id和vidar字符串，如果不是不一样则die，所以id需要和vidar字符串不一样，同时url解码后的id需要与vidar字符串相同，百度url编码：参考博文：

https://www.cnblogs.com/jerrysion/p/5522673.html

于是构造url:

http://120.78.184.111:8080/week1/very_ez/index.php?id=%76idar

跳转后发现die且url自动将id值变成了vidar，知浏览器自动进行一次url解码，故需进行连续两次编码，即构造：

http://120.78.184.111:8080/week1/very_ez/index.php?id=%2576idar

得到flag:hgame{urlDecode_Is_GoOd}

附：url编码：编码原则：%+十六进制表示，对会引起歧义的特殊字符及汉字等进行编码，通常浏览器会自动进行一次解码，即上述%2576跳转后其实id值变为%76，Url中只允许包含英文字母（a-z/A-Z）、数字（0-9）、-_.~4个特殊字符以及所有保留字符，即上述字符经过url解码后依然是自身，这也是为什么vidar中只需要对任一字母进行编码处理，若全部进行编码处理也可以，只是没有必要。

4.can u find me?

打开题目，发现只有一句话：

> the gate has been hidden
>
> can you find it? xixixi

尝试查看网页源代码（HTML中可以隐藏信息的方法:注释、head头部、超链接等等），果然发现超链接中有隐藏信息，打开即：

yeah!you find the gate

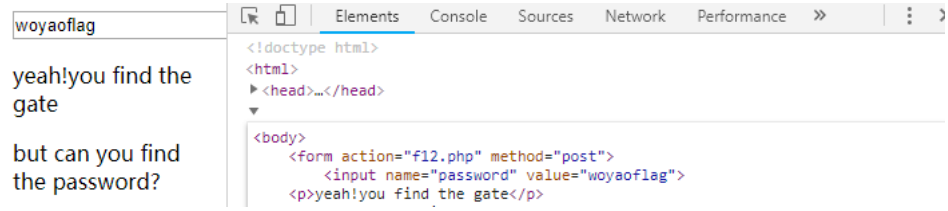
but can you find the password?

please post password to me! I will open the gate for you!

再次尝试源代码，没有发现，故尝试抓包，发送到repeater查看响应，果然找到password:

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Wed, 30 Jan 2019 08:03:46 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.2.14
password: woyaoflag
Content-Length: 242
```

题目要求post password，所以想到利用html表单，编辑网页源代码如下：



回车，得到新的网页，结果：

aoh,your speed is sososo fast,the flag must have been left in somewhere

翻译一下：跑的太快了，flag被留在了某处！结合题目中关于302跳转的资料，想到此页面应该是经过302跳转后的页面，so,回到原网页进行抓包：

```
HTTP/1.1 302 Found
Server: nginx/1.15.8
Date: Wed, 30 Jan 2019 08:16:53 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.2.14
location: toofast.php
Content-Length: 132

<html>
    <head>
        <title>can you find me?</title>
    </head>
    <body>
        <p>flag:hgame{f12_1s_aMazIng111}</p>
    </body>
</html>
```

果然，响应302，抓到了跳转前的包，得到flag:hgame{f12_1s_aMazIng111}

Re:

1.brainfxxker:

打开题目，分析代码

```
void Parser::execute(const std::string &buf) {
  for (auto i = buf.cbegin(); i != buf.cend(); ++i) {
    switch (*i) {
      case '>':
        ++ptr;
        break;
      case '<':
        --ptr;
        break;
      case '+':
        ++data[ptr];
        break;
      case '-':
        --data[ptr];
        break;
      case '.':
        putchar(data[ptr]);
        break;
      case ',':
        while ((data[ptr] = getchar()) == '\n') ;
        break;
      case '[':
        if (!data[ptr]) {
          while (*i++ != ']') continue;
          --i;
        }
        break;
      case ']':
        if (data[ptr]) {
          while (*(i-1) != '[') --i;
          --i;
        }
        break;
      default:
        break;
    }
  }
}
```

（结合Wiki的解释看懂后），执行：

```
  parser.execute(",>++++++++++[<---------->-]<++[+.],>+++++++++[<--------
->-]<-[+.],>+++++++[<------->-]<---[+.],>++++++[<------>-]<+++
[+.],>++++++++[<---------->-]<++[+.],>++++++++++[<---------->-]<--
[+.],>++++++++++[<--------->-]<-----[+.],>++++++++++[<---------->-]<+
[+.],>+++++++++[<--------->-]<---[+.]");
}"
```

根据notice2:知道规则是不能执行[+.]这一部分，所以分析如下：经过“，”没有变化，经过“>”ptr变成1，随后data[1]变成10，由于[]是当data[1]不为零时疯狂循环的意思，所以data[0]会经过10遍减10的操作，从而变成-100，接着data[0]变成-98，由于正确的输入不能执行[+.]这个部分，所以data[0]此时应该为0，所以一开始的输入应该是98，联想到ASCII码值进行查表操作，得b，后面部分如法炮制，得flag:hgame{bR4!NfUcK}

2.HelloRe:

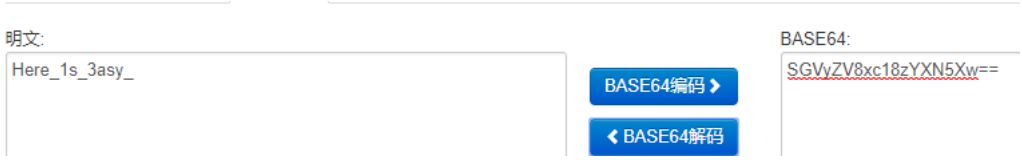
点击链接，自动下载了一个文件，尝试用文本编辑器打开，运气很好的找到了flag:hgame{Welc0m3_t0_R3_World!}

3.Pro的Python教室（一）

打开题目，发现python代码，分析得输入的第一部分即：hgame{，第二部分找到关键

```
secend = raw_input()
secend = base64.b64encode(secend)
```

于是百度搜索：base64，发现在线编解码，对enc2进行解码：



得到第二部分，第三部分进行编解码操作都尝试失败，n次失败后想到句子的意思可能是“这是简单的python”，输入后成功，即flag:hgame{Here_1s_3asy_Pyth0n}

Misc:

1.Hidden image in LSB

根据题目送分的提示：下载神奇的神器stegsolve（下载过程一言难尽：下载stegsolve-下载jdk-设置java环境配置变量-编写.bat文件在stegsolve的目录下即可打开stegsolve），利用它打开题目中的图像：当调至Red plane模式时(疯狂点击左右箭头)，得到flag:hgame{LSB_is_easy_for_u}

2.打字机

打开题目，自动下载压缩包，打开发现图片两张，一张奇形怪状的打字机，一张flag，看来就是破解flag图片上的意思了，根据格式知道前五个字母是：hgame,从而破解第五个字符是e，带入后面的相同字符，根据提示用谷歌进行以图搜图，结果居然是...日漫紫罗兰与永恒花园中的打字机
emmmm，好吧...根据原型：

继续破解出：M，0，P，w，R，r (猜测字母v,i,L)

由此根据猜测知：hgame{M *_ vi0Le *_ ** PewRi *er}，然后根据我不咋地的英语直觉好像是紫罗兰的英文，搜索得到violet，果断破解t，变成：hgame{M _vi0Let_t PewRiter}，百度搜索了一下后半部分，搜索结果typewriter打字机，瞬间感觉正解，带入发现不是很对啊，然后n次实验发现应该是小写l，故得到flag:hgame{My_vi0let_tyPewRiter}

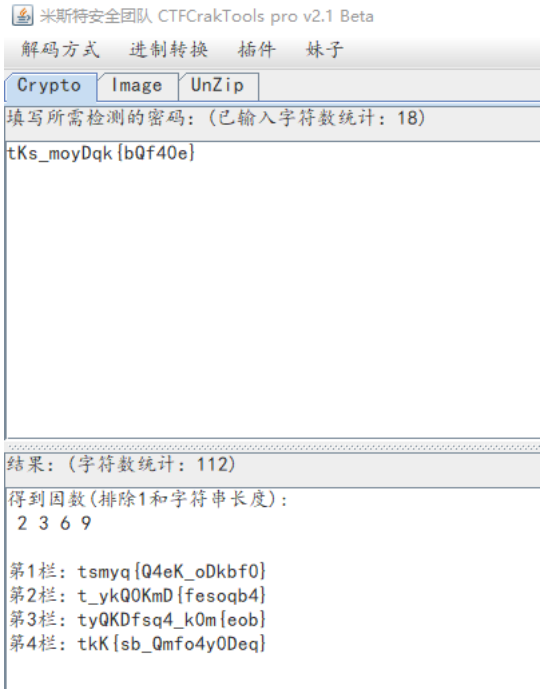
Crypto:

1.Mix:

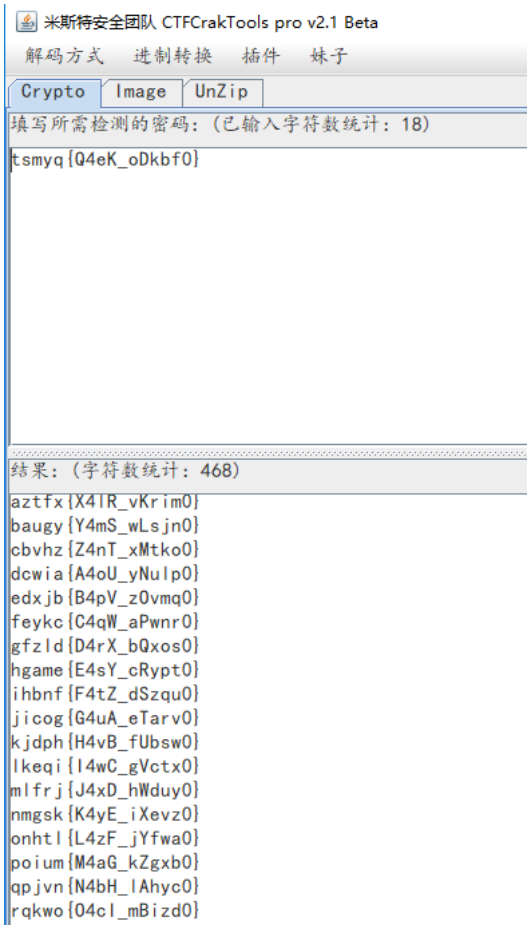
根据description中的

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

联想到摩斯密码进行破解得：744B735F6D6F7944716B7B6251663430657D，反复反复反复（重要的事情说三遍，再观察不出就#%X，一般混合密码得线索可通过观察字符串特点，题目提示得出）观察想到十六进制：尝试两位两位破解，得：tKs_moyDqk{bQf40e}，此时就比较开心了，根据{}的位置，考虑栅栏解码，利用神器得：



根据flag格式选择第一栏，进行凯撒解码如下：



看见hgame复制flag:hgame{E4sY_cRypt0}

参考学习资料（十六进制解码）：

https://zh.wikipedia.org/w/index.php?title=ASCII