

# Hackinglab基础关Write Up

原创

A1andNS 于 2019-09-30 12:18:40 发布 261 收藏 1

分类专栏: WP 文章标签: CTF 网络安全

版权声明: 本文为博主原创文章, 遵循CC 4.0 BY-SA 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_26139045/article/details/101761489](https://blog.csdn.net/qq_26139045/article/details/101761489)

版权



WP 专栏收录该内容

4 篇文章 0 订阅

订阅专栏

## 基础题 (AlanLee)

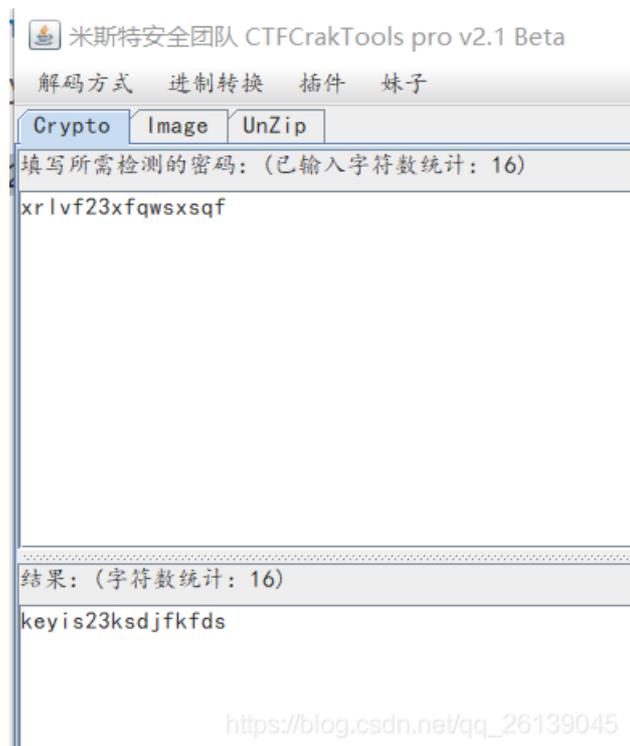
### 第一题: key在哪里?

打开目标网站, 按他的提示就在这里, 查看网页源代码就得到了key is jflsjklejflkdsjfklds

```
Elements Console Sources Network
<html>
  <head>...</head>
  <body> == $0
    "
    key就在这里中, 你能找到他吗?
    "
  <!--key is jflsjklejflkdsjfklds-->
```

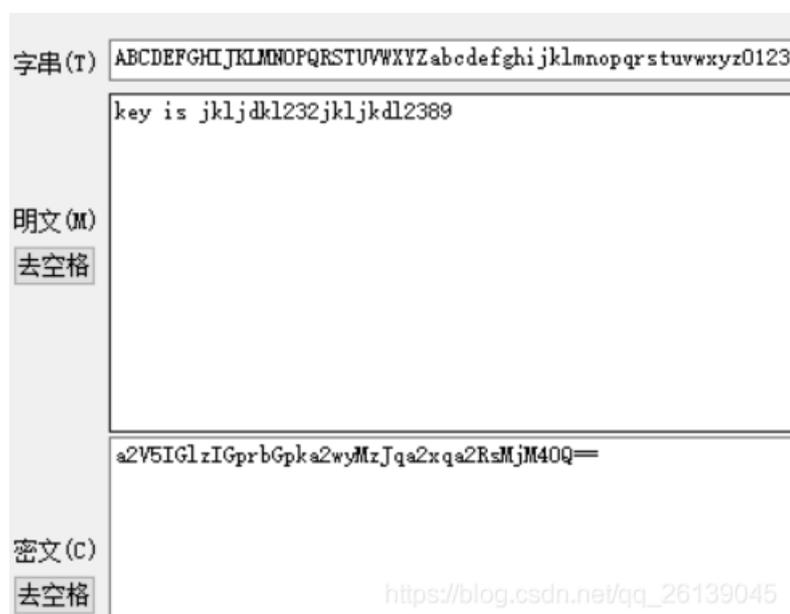
### 第二题: 再加密一次你就的到key

根据提示我们知道了明文加密一次后是密文，再加密一次又可以得到明文，故判定是rot13加密的，我使用了CTFCrakTools进行rot13解密，从而得到一个key



### 第三题：猜猜这是经过了多少次加密

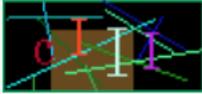
看到密文是以=结尾马上联想到base64编码，故进行base64解密，最后得到key。



### 第四题：据说MD5加密很安全，真的是么？

直接对密文进行MD5解密，得到答案。

e0960851294d7b2253978ba858e24633

输入验证码 

MD5  
解密

! e0960851294d7b2253978ba858e24633", 解密的结果为"bighp"!

[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

## 第五题：种族歧视

打开网站发现无法访问，根据题目提示知道是accept-language的问题。网站一般通过识别请求头中的接受语言判断访问者的国籍。所以我把浏览器语言改成English问题就解决了，得到key。

request

raw headers hex

```
GET /base1_Def337f3afbe42d5619d7a36c19c20ab/index.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://hackinglab.cn/ShowQues.php?type=bases
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

response

raw headers hex html render

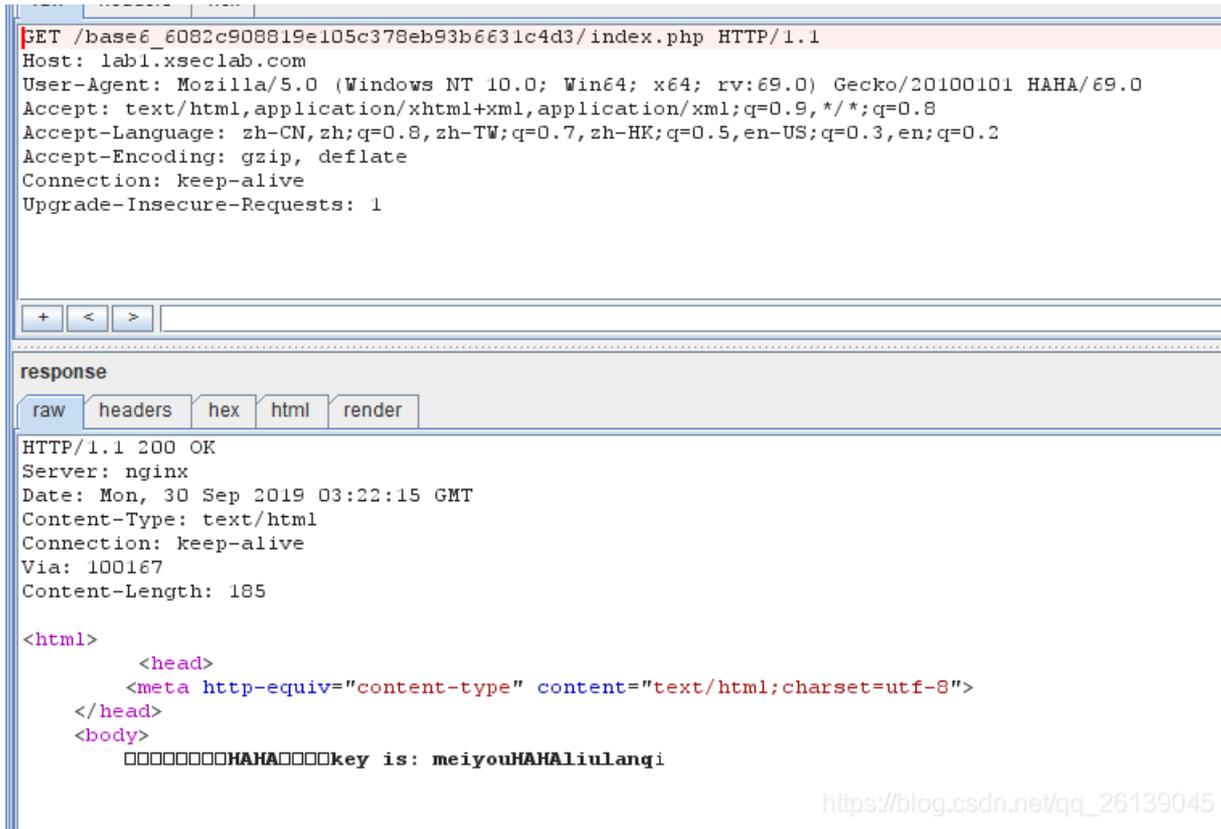
```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 30 Sep 2019 03:21:06 GMT
Content-Type: text/html
Connection: keep-alive
Via: 100167
Content-Length: 141

<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
  </head>
  <body>
    key is: *(TU687jksf6&*
```

[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

## 第六题：HAHA浏览器

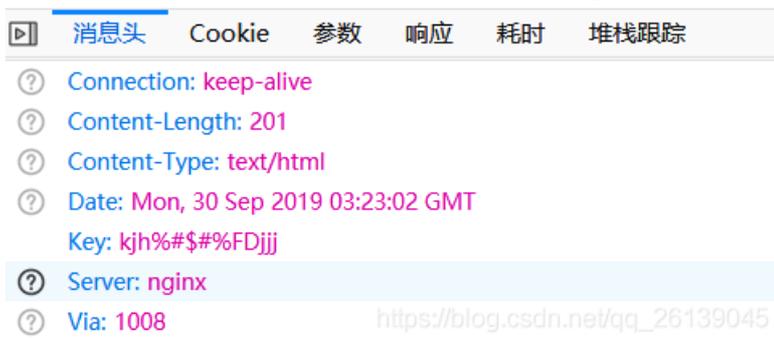
打开网站提示需要使用HAHA浏览器打开，很明显没有什么HAHA浏览器，那么我就使用burpsuit抓包，更改请求头里的UA信息，把浏览器改成HAHA，重发数据包得到key。



[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

## 第七题：key究竟在哪里

进入网页首先打开控制台，查看源代码。发现源代码里面没有需要的信息。转到网络模块查看响应头，发现key藏在里面。



[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

## 第八题：key又找不到了

打开网站发现有个链接写着到“这里找key”，于是我试着打开它看看。发现里面写着没有密码，于是尝试看看网页源码，发现也没有。找了半天都没有，结合他的密码不在这里的提示，于是我决定用burpsuit抓个包试试看，包一抓发现有个状态码为302的包。

|                      |     |                                 |     |     |      |
|----------------------|-----|---------------------------------|-----|-----|------|
| http://lab1.xsecl... | GET | /base8_0abd63aa54bef0464289d... | 200 | 347 | HTML |
| http://lab1.xsecl... | GET | /base8_0abd63aa54bef0464289d... | 302 | 475 | HTML |

```

HTTP/1.1 302 Found
Server: nginx
Date: Mon, 30 Sep 2019 03:24:01 GMT
Content-Type: text/html
Connection: keep-alive
Location: http://hacklist.sinaapp.com/base8_0abd63aa54bef0464289d6a42465f354/
Via: 10080
Content-Length: 224

```

```

<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
  </head>
  <body>
    <a href="./key_is_here_now_.php">__</a><!--00000000key00-->
  </body>
</html>

```

[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

于是进去看看，发现响应包里面有货，在返回的HTML里有一个key\_is\_here\_now\_.php网页，copy一下到url试一下，得到key: ohHTTP302dd



key: ohHTTP302dd

## 第九题：冒充登陆用户

进入网站发现需要登陆，但是没有登陆框，也是打开控制台查看网络看请求头和回应头，发现login=0，于是我们通过burpsuit进行抓包，修改数据包的头部把login=0改成login=1释放数据包，得到key。

The screenshot shows a network request and response in Burp Suite. The request is a GET to /base9\_ab629d778e3a29540dfd60f2e548a5eb/index.php with a Cookie: Login=1. The response is an HTTP 200 OK from nginx with a Set-Cookie: Login=0 and a body containing 'key is: yescookieedit7823789KJ'.

```
raw params headers hex
GET /base9_ab629d778e3a29540dfd60f2e548a5eb/index.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: Login=1
Upgrade-Insecure-Requests: 1

response
raw headers hex html render
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 30 Sep 2019 03:27:42 GMT
Content-Type: text/html
Connection: keep-alive
Via: 1008
Set-Cookie: Login=0
Content-Length: 152

<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
  </head>
  <body>
    key is: yescookieedit7823789KJ
```

[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

## 第十题：比较数字大小

首先我先试着提交一下发现，只能输入3位数，这说明输入框的长度被限制了。我查看源代码，发现确实如此，我把长度改成比3大的数字，然后再输入数据，（例如我把长度改成了6位后，输入111111提交）得到key。

```
<head>...</head>
<body>
  <form action="" method="post">
    <input type="text" maxlength="3" name="v">
    <input type="submit" value="提交">
  </form>
```

key is 768HKyu678567&\*&K

[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

## 第十一题：本地的诱惑

打开网页提示需要从本地访问，打开查看源代码，发现隐藏了php语句，发现key

```
<!--
?php //print_r($_SERVER); $arr=explode(',',$_SERVER['HTTP_X_FORWARDED_FOR']);
if($arr[0]=='127.0.0.1'){ //key echo "key is ^&*(UIHKJjkadshf"; }else{ echo "必须从本地访问!"; } ?
-->
```

## 第十二题：就不让你访问



I am index.php , I am not the admin page ,key is in admin page.

[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

日常看看源代码有没有秘密，发现也没有。

看看响应头发现什么都没有。

burpsuit抓个包也没有。。。。。

看来就是需要找到后台所在，考察后台查找

直接先试一下login.php和/admin/login.php以及manager.php之类的常见的后台发现都不行。

于是我们查看网站信息用蜘蛛协议，robots.txt



发现disallow，把disallow加到url的末尾访问



you find me,but I am not the login page. keep search.

结果。。。。

还不是叫我继续。。。。。

那就直接在这个目录下面试一试index.php

发现还是这个网页

那我再试一下login.php

发现成功了得到了key

[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)



right! key is UIJ%%!OOqweqwdsf

[https://blog.csdn.net/qq\\_26139045](https://blog.csdn.net/qq_26139045)

这里解释一下蜘蛛协议的内容，User-agent: \* 这里的代表所有的搜索引擎种类，\*是一个通配符。

disallow: /xxx/ 表示禁止搜索引擎访问的目录

这里解释一下蜘蛛协议的内容，User-agent: \* 这里的代表所有的搜索引擎种类，\*是一个通配符。

disallow: /xxx/ 表示禁止搜索引擎访问的目录