

网络安全实验室CTF—基础关 writeup

原创

Senimo_ 于 2019-08-02 20:49:59 发布 2379 收藏 9

分类专栏: [网络安全实验室CTF writeup](#) 文章标签: [网络安全实验室 CTF writeup 基础关](#)

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/weixin_44037296/article/details/98237644

版权



[网络安全实验室CTF writeup](#) 专栏收录该内容

6 篇文章 0 订阅

订阅专栏

网络安全实验室CTF—基础关 writeup

key在哪里?

再加密一次你就得到key啦~

知识点: ROT13密码

猜猜这是经过了多少次加密?

据说MD5加密很安全, 真的是么?

种族歧视

HAHA浏览器

key究竟在哪里?

key又找不到了

冒充登陆用户

比较数字大小

本地的诱惑

就不让你访问

知识点: robots.txt

网络安全实验室CTF链接

key在哪里?

分值: 100

过关地址

进入页面后显示: "key就在这里中, 你能找到他吗?";

```
<body>
key就在这里中, 你能找到他吗?
  <!--key is jflsjklejflkdsjfklds-->
</body>
```

查看网页源代码，得到key。

再加密一次你就得到key啦～

分值：150

加密之后的数据为xrlvf23xfqwsxsqf

提示为加密一次，并给出了加密后的数据，尝试了大多数加密及编码失败后；

重新思考题意，给出了加密数据，再加密一次即为解密，想到回`转13位加密`，解密后的数据为：`keyis23ksdjfkfds`，即为key。

知识点：ROT13密码

ROT13（回`转13位`，rotate by 13 places，有时中间加了个连字符称作ROT-13）是一种简易的替换式密码。它是一种在英文网络论坛用作隐藏八卦（spoiler）、妙句、谜题解答以及某些脏话的工具，目的是逃过版主或管理员的匆匆一瞥。ROT13被描述成“杂志字谜上下颠倒解答的Usenet点对点体”。ROT13也是过去在古罗马开发的凯撒加密的一种变体。

猜猜这是经过了多少次加密？

分值：200

加密后的字符串为（省略了中间部分字符）：

```
Vm0wd2QyUX1VWGxwV0d4V1YwZDRWMV13WkRSV01WbDNXa1JTVjAxV2JET1hhMUpUVmpBeFYySkVUbGhoTVVwVVZtcEJlR1l5U2tWVWJVkZscmFF  
SmtNV1J6Vm0xR2FFMvdjRmxWTW5SaFlXeEtXR1ZIUmxwV1JUvkVxbFphVjFJeFNsVm1Sa1pXVmtSQk5RPT0=
```

根据结尾的等号判断为Base64编码，尝试在线Base64解码多次解码后，（解码时注意结尾的“=”）得到key：`key is jk1jdk1232jk1jdk12389`

据说MD5加密很安全，真的是么？

分值：200

e0960851294d7b2253978ba858e24633

题目给出为MD5加密，尝试在线MD5解密，得到`bighp`，即为key。

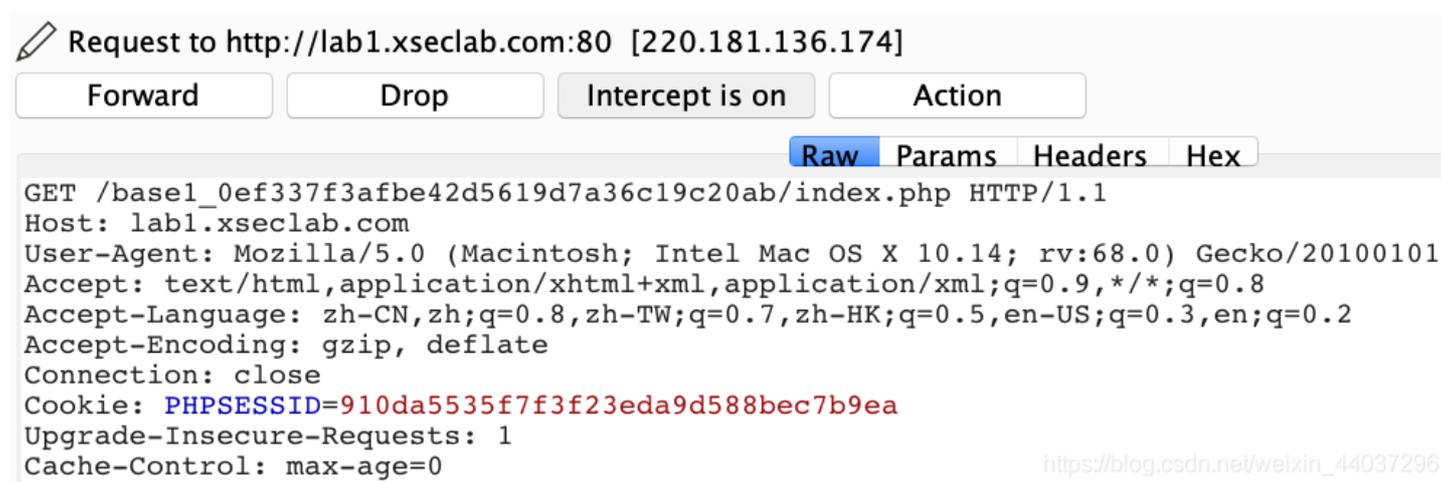
种族歧视

分值：300

小明同学今天访问了一个网站，竟然不允许中国人访问！太坑了，于是小明同学决心一定要进去一探究竟！

通关地址

进入网页后显示：“only for Foreigner”，尝试用Burp Suite抓取数据包，修改语言参数：



```
Request to http://lab1.xseclab.com:80 [220.181.136.174]
Forward Drop Intercept is on Action
Raw Params Headers Hex
GET /base1_0ef337f3afbe42d5619d7a36c19c20ab/index.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:68.0) Gecko/20100101
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: close
Cookie: PHPSESSID=910da5535f7f3f23eda9d588bec7b9ea
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
https://blog.csdn.net/weixin_44037296
```

将 `Accept-Language` 中的 `zn-CN`、`zh`、`zh-TW`、`zh-HK`，发送数据包，得到key：“key is: (TU687jksf6&”

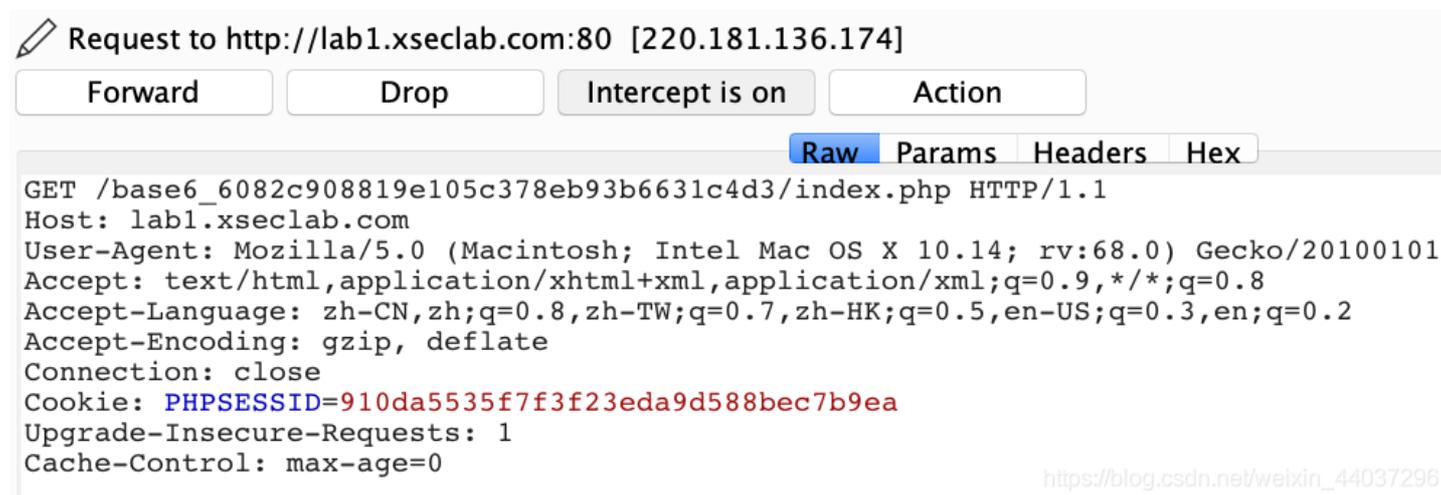
HAHA浏览器

分值：200

据说信息安全小组最近出了一款新的浏览器，叫HAHA浏览器，有些题目必须通过HAHA浏览器才能答对。小明同学坚决不要装HAHA浏览器，怕有后门，但是如何才能过这个需要安装HAHA浏览器才能过的题目呢？

通关地址

进入页面显示：“只允许使用HAHA浏览器，请下载HAHA浏览器访问！”，尝试通过Burp Suite修改 `User-Agent`：中的 `Mozilla` 修改为 `HAHA`；



```
Request to http://lab1.xseclab.com:80 [220.181.136.174]
Forward Drop Intercept is on Action
Raw Params Headers Hex
GET /base6_6082c908819e105c378eb93b6631c4d3/index.php HTTP/1.1
Host: lab1.xseclab.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:68.0) Gecko/20100101
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: close
Cookie: PHPSESSID=910da5535f7f3f23eda9d588bec7b9ea
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
https://blog.csdn.net/weixin_44037296
```

发送数据包后得到页面显示：“恭喜您，成功安装HAHA浏览器！key is: meiyouHAHAlulanqi”

key究竟在哪里？

分值：200

上一次小明同学轻松找到了key，感觉这么简单的题目多无聊，于是有了找key的加强版，那么key这次会藏在哪里呢？

通关地址

进入页面后显示：“**Key**就在这里，猜猜这里是哪里呢？(Web找key加强版)”

尝试用Burp Suite抓取数据包，执行Send to Repeater后发送数据包，在Response中得到key:

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 04 Aug 2019 14:35:02 GMT
Content-Type: text/html
Connection: close
Key: kjh%#$%FDjjj
Via: 10080
Content-Length: 201
```

https://blog.csdn.net/weixin_44037296

key又找不到了

分值: 350

小明这次可真找不到key去哪里了，你能帮他找到key吗？

通关地址

进入页面后显示：“到这里找key”，查看网页源代码

```
<body>
  <a href="./search_key.php">_到这里找key__</a>
</body>
```

点击链接后跳转到新的页面“key is not here!”;

观察到新页面的URL显示: `index_no_key.php`

```
http://hacklist.sinaapp.com/...../index_no_key.php
```

使用Burp Suite抓取第一次跳转的数据包，Send to Repeater后，发送数据包，在Response中得到新的提示：

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 302 Found
Server: nginx
Date: Sun, 04 Aug 2019 14:52:37 GMT
Content-Type: text/html
Connection: close
Location:
http://hacklist.sinaapp.com/base8_0abd63aa54bef0464
289d6a42465f354/index_no_key.php
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><!--都告诉了到这
里找key的啦-->
  </body>
</html>
```

https://blog.csdn.net/weixin_44037296

将Request中的请求链接 `search_key.php` 替换为 `key_is_here_now_.php`，发送数据包后，在Response中的到key：

Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 04 Aug 2019 14:56:02 GMT
Content-Type: text/html
Connection: close
Via: 10080
Content-Length: 16

key: ohHTTP302dd
```

https://blog.csdn.net/weixin_44037296

冒充登陆用户

分值：200

小明来到一个网站，还是想要key，但是却怎么逗登陆不了，你能帮他登陆吗？

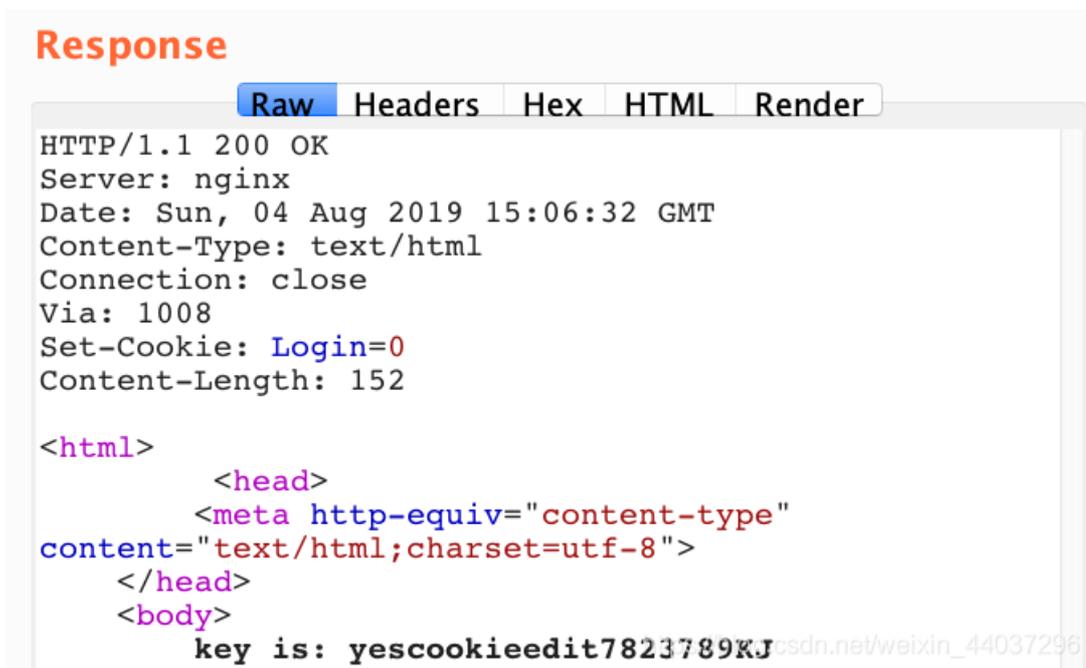
通关地址

进入网页后显示：“您还没有登陆呢！”

使用Burp Suite抓取数据包，Send to Repeater后，发送数据包，在Response中得到新的提示：



在Request中将原来Cookie的值替换为 Login=1，发送数据后，在Response中的到key:



比较数字大小

分值: 100

只要比服务器上的数字大就可以了!

通关地址

进入页面后只有一个输入框，提示需要输入大于服务器上的数字，但在HTML代码对输入框输入的长度做了限制：

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

将“<input>标签”中的“maxlength”属性的长度限制删除或为空，输入 999999999，提交后便得到key:

key is 768HKyu678567&*&K

本地的诱惑

分值: 200

小明扫描了他心爱的小红的电脑，发现开放了一个80端口，但是当小明去访问的时候却发现只允许从本地访问，可他心爱的小红不敢让这个诡异的小明触碰她的电脑，可小明真的想知道小红电脑的80端口到底隐藏着什么秘密(key)?

通关地址

进入页面后显示：“必须从本地访问！”，查看网页源码：

```
<?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! ";
}
?>
<?php
//SAE 服务调整, 该题目无法继续...可尝试自行搭建环境测试.
echo file_get_contents(__FILE__);
```

因为服务调整，所以key以给出，可尝试本地搭建环境测试，在Google Chrome插件ModHeader中添加HTTP本地请求头 X-Forwarded-For: 127.0.0.1 即可得到key:

就不让你访问

分值: 150

小明设计了一个网站，因为总是遭受黑客攻击后台，所以这次他把后台放到了一个无论是什么人都找不到的地方...可最后还是被黑客找到了，并被放置了一个黑页，写到:find you ,no more than 3 secs!

通关地址

进入网页后显示：“I am index.php ,I am not the admin page ,key is in admin page.”，尝试访问 admin.php，但显示“Not Found”，查看网页是否存在 robots.txt :

```
User-agent: *
Disallow: /
Crawl-delay: 120
Disallow: /9fb97531fe95594603aff7e794ab2f5f/
Sitemap: http://www.hackinlab.sinaapp.com/sitemap.xml
```

显示有Disallow: 不允许爬取的页面，将原网页 index.php 替换为 9fb97531fe95594603aff7e794ab2f5f/ 访问该链接得到新的提示：“you find me,but I am not the login page. keep search.”

提示为login页面，在URL后添加 login.php ,即得到key: “right! key is UIJ%%!OOqweqwdsf”

知识点: robots.txt

robots协议是网站跟爬虫间的协议，用简单直接的txt格式文本方式告诉对应的爬虫被允许的权限，也就是说robots.txt是搜索引擎中访问网站的时候要查看的第一个文件。当一个搜索蜘蛛访问一个站点时，它会首先检查该站点根目录下是否存在robots.txt，如果存在，搜索机器人就会按照该文件中的内容来确定访问的范围；如果该文件不存在，所有的搜索蜘蛛将能够访问网站上所有没有被口令保护的页面。