ctf web WriteUp1

原创

weixin_44219914 于 2019-10-14 20:52:36 发布 285 收藏 1

分类专栏： ctf 文章标签： ctf web

本文链接：https://blog.csdn.net/weixin_44219914/article/details/102556500

> 年纪大了，方法总结的东西不写下来一会儿就忘了

web2(bugku) & 佛说 – 白给(asuri)

网址链接：

http://120.24.86.145:8002/web2/

47.102.107.100:39002

直接按F12，隐藏在Elements里

Asuri另外一题白给2，按F12，Console里点击链接跳转

就能看到flag了

佛说 – 白给 3

网址链接：http://47.102.107.100:39004/

按F12，按F5刷新，点Network，隐藏在请求头Header里了

佛说 – 白给 4

网址前加`view-source:`看源码

计算器(bugku)

网址链接：http://123.206.87.240:8002/yanzhengma/

出来的计算题是随机的，幸运的时候答案是个位数，点验证就获得了flag，不幸运的时候答案多余1位数，但框内只能填1位数，查看源码，修改输入框的`maxlength`限制，改为>=1即可，输入多位答案，点击验证，跳出flag

web 基础 $GET & web 基础 $POST & 点击一百万次(bugku)

两种最常用的 HTTP 方法是：GET 和 POST

- **GET** - 从指定的资源请求数据。
- **POST** - 向指定的资源提交要被处理的数据

Get方法

```
/test/demo_form.asp?name1=value1&name2=value2
```

- GET 请求可被缓存
- GET 请求保留在浏览器历史记录中
- GET 请求可被收藏为书签
- GET 请求不应在处理敏感数据时使用
- GET 请求有长度限制
- GET 请求只应当用于取回数据

POST 方法

查询字符串（名称 / 值对）是在 POST 请求的 HTTP 消息主体中发送的：

```
POST /test/demo_form.asp HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2
```

- POST 请求不会被缓存
- POST 请求不会保留在浏览器历史记录中
- POST 不能被收藏为书签
- POST 请求对数据长度没有要求

web 基础 $GET：

网址链接：http://123.206.87.240:8002/get/

web 基础 $POST：

网址链接：http://123.206.87.240:8002/post/

安装火狐的插件hackbar

“点击一百万次”

源码中判断`clicks`变量是否大于等于1000000，hackbar中post data，填`clicks = 1000001`得到flag

矛盾(bugku)

网址链接：http://123.206.87.240:8002/get/index1.php

输入一个不是数字但`num==1`，`==`是弱类型比较，如果等号两边类型不同先转换成相同类型，字符串会转成数字，具体是保留字母前的数字，例如 123ab7c 会转成 123，ab7c 会转成 0（字母前没数字就是 0）。这里用Get方法改链接`?num=1a`。

web3(bugku)

网址链接：http://123.206.87.240:8002/web3/

用到Burp suite抓包软件，安装后第一次尝试，随机抓取一个地址，查看拦截的请求

根据官网指示安装CA certification，设置Https代理。

问题来了，尝试抓包`https://www.baidu.com/`

后面百度莫得界面，forward之后就

不知道怎么解决了…

这题没要求https抓包，抓包后F12看到调试器下面一行字符

粘贴进burp自带的decoder，选decoder as html，看到flag

域名解析（ bugku）

网址链接：123.206.87.240

域名解析是指把一个域名指向一个 ip，就像通讯录把姓名指向一个电话一样。用burpsuite抓包，直接把host里的ip改成域名即可。

你必须让他停下 & 头等舱(bugku)

网址链接：http://123.206.87.240:8002/web12/

123.206.87.240:9009/hd.php

用到`burp repeater`，`burp repeater`作为 burp suite 中一款验证 HTTP 消息的测试工具，通常用于**多次重放请求响应**和**手工修改请求消息的修改后对服务器端响应的消息分析**

这题先抓包，再`forward`，查看`http history`，右击捕获到的get数据包，选`sent to repeater`，在`Repeater`中点几次`Go`，找找到flag

头等舱打开什么也没有，burpsuite抓包后`sent to repeater`，`Go`一次就看到了flag。

变量(bugku)

网址链接：http://123.206.87.240:8004/index1.php

用到php的知识，打印所有的全局变量，都在$GLOBALS [*index*] 数组中，*index* 保存变量的名称。在链接后面加Get请求http://123.206.87.240:8004/index1.php?args=GLOBALS

web5（bugku）

查看源码，看到一大串符号，是 JS 代码经过 jsfuck 编码的格式，复制粘贴到控制台，回车

网站被黑 & 输入密码查看 flag(bugku)

都用到爆破

网址链接：http://123.206.87.240:8002/webshell/

http://123.206.87.240:8002/baopo/

根据提示，虽然没什么用，但是经常遇到，`webshell`，猜测这个网站存在`webshell`

webshell就是以asp、php、jsp或者cgi等网页文件形式存在的一种命令执行环境，也可以将其称做为一种网页后门。黑客在入侵了一个网站后，通常会将asp或php后门文件与网站服务器WEB目录下正常的网页文件混在一起，然后就可以使用浏览器来访问asp或者php后门，得到一个命令执行环境，以达到控制网站服务器的目的。

猜测：http://123.206.87.240:8002/webshell/shell.php

要填pass，用burpsuite抓包，随便填一个如`admin`，forward放行，这时burpsuite

Raw空白处右击，`sent to Intruder`。

在Intruder中Payload Options的`Add from list`处选`Password`，右上角`Intruder`–>`start attact`，等待结果

将结果按`Length`升序排序，第一个是`hack`，选中，下方Response中找到flag。

下一题“输入密码查看 flag”

抓包，放行，随意输入5位如12345，放行，`sent to intruder`，在 Positions 中点击 clear 清除 burp 认为需要猜测的密码，然后选中 12345 (也就是我们刚才输入的密码，点击 add)，改Payloads中的设置，

Options中设置Number of threads为100，等待后结果为13579，得到flag。

管理员系统(bugku)

网址链接：http://123.206.31.85:1003/

随便输入一个用户名和密码，提示`请联系本地管理员登录`。

直接F12有一串字符`dGVzdDEyMw==`，base64解码得到test123，猜想是密码，用户名随便填一个admin，用burpsuite抓包，填用户名和密码后伪装成本地访问，改包：Headers 中增添一对键值对：`X-Forwarded-For : 127.0.0.1`，X-Forwarded-For 是一个 HTTP 扩展头部，主要是为了让 Web 服务器获取访问用户的真实 IP 地址，但是这个 IP 却未必是真实的。一些开发者为了获取客户 IP，经常会使用 request.remote_ip 来获得用户 IP。但是很多用户都是通过代理来访问服务器的，如果使用 remote_ip 这个全局变量来获取 IP，开发者往往获得的是代理服务器的 IP，并不是用户真正的 IP。

放行，得到flag。

wab4(bugku)

提示要unescape

在线unescape这一大段，得到一串字符，填入框内submit得到flag。

flag 在 index 里（bugku）

网址链接：http://123.206.87.240:8005/post/index.php?file=show.php

这里用到一个读取 php 文件源码的方法，是文件包含漏洞（本地文件包含（Local File Include），简称 LFI）。构造URL：http://123.206.87.240:8005/post/index.php?file=php://filter/read/convert.base64-encode/resource=index.php

然后来解释`php://filter/read/convert.base64-encode/resource=index.php`

首先这是一个 file 关键字的 get 参数传递，`php://`是一种协议名称，`php://filter/`是一种访问本地文件的协议，`/read=convert.base64-encode /`表示读取的方式是 base64 编码后，`resource=index.php`表示目标文件为`index.php`。输入这个链接，抓包，再放行，这样就能读取文件源码并以 base64 的 方式输出。将 base64 码解码就拿到 flag 了。

备份是个好习惯(bugku)

备份文件一般情况是在后缀名后加的 .swp，.bak，于是将 URL 改成`123.206.87.240:8002/web16/index.php.bak`，回车，下载到一个 php 文件，

```php
<?php
/**
 * Created by PhpStorm.
 * User: Norse
 * Date: 2017/8/6
 * Time: 20:22
*/

include_once "flag.php";
ini_set("display_errors", 0);
$str = strstr($_SERVER['REQUEST_URI'], '?');
$str = substr($str,1);
$str = str_replace('key','',$str);
parse_str($str);
echo md5($key1);

echo md5($key2);
if(md5($key1) == md5($key2) && $key1 !== $key2){
    echo $flag."取得flag";
}
?>
```

网上说双写key绕过`str_replace`，这个函数就是把`$str`里的`key`替换成空，双写成`kkeyey`或者`kekeyy`都行，key1和key2不同但md5加密值要相同

构造两个md5值都为0e开头的

```
http://123.206.87.240:8002/web16/index.php?kekeyy1=s878926199a&kekeyy2=s155964671a
```

或者利用无法`hash`的数组，返回空来绕过

```
http://123.206.87.240:8002/web16/index.php?kkeyey1[]=1&kkeyey2[]=2
```

都能得到flag。