# bugku 代码审计 write up

转载

xuchen16 于 2018-09-26 13:15:58 发布 106 收藏
分类专栏： ctf 文章标签： bugku 代码审计 writeup bugku 代码审计 wp


转载自:https://blog.csdn.net/u011377996/article/details/79332632

## extract变量覆盖

查一下php手册

```
int extract ( array &$array [, int $flags = EXTR_OVERWRITE [, string $prefix = NULL ]] )
```



本函数用来将变量从数组中导入到当前的符号表中。
返回成功导入到符号表中的变量数目。
而且这个函数没还有任何参数，很危险，直接可以修改冲突的变量

EXTR_OVERWRITE
如果有冲突，覆盖已有的变量。

EXTR_SKIP
如果有冲突，不覆盖已有的变量。

EXTR_PREFIX_SAME
如果有冲突，在变量名前加上前缀 prefix。

EXTR_PREFIX_ALL
给所有变量名加上前缀 prefix。

EXTR_PREFIX_INVALID
仅在非法 / 数字的变量名前加上前缀 prefix。

EXTR_IF_EXISTS
仅在当前符号表中已有同名变量时，覆盖它们的值。其它的都不处理。举个例子，以下情况非常有用：定义一些有效变量，然后从 $_REQUEST 中仅导入这些已定义的变量。

EXTR_PREFIX_IF_EXISTS
仅在当前符号表中已有同名变量时，建立附加了前缀的变量名，其它的都不处理。

EXTR_REFS
截图(Alt + A)
将变量作为引用提取。这有力地表明了导入的变量仍然引用了 array 参数的值。可以单独使用这个标志或者在 flags 中用 OR 与其它任何标志结合使用。

如果没有指定 flags，则被假定为 EXTR_OVERWRITE。



关键的代码

```
$flag='xxx';
extract($_GET);
if(isset($shiyan))
{
    $content=trim(file_get_contents($flag));
    if($shiyan==$content)
    {
```



构建的payload: `?shiyan=&flag=`
因为不知道flag里面什么内容，让它变成空，然后使content变成空，然后content变成空，然后shiyan变量和$content变量的内容都会被设置成空. 满足条件便会出现flag

## strcmp比较字符串

这题关键的函数都没给出来。。。题目说用strcmp的特性。。。
那我就随便个数组进去一个?a[]...flag就出来了。。尴尬

## urldecode二次编码绕过

这个题好像在实验吧做过。。。。。
利用了两次urldecode第一次是浏览器的解码第二次是函数的解码
所以我利用了里面的其中一个字母D编码第一次编码是%44，第二次编码是%2544
payload: `?id=hacker%2544J`

## md5()函数

利用php的md5()函数有一个缺陷，这里是===，只能用数组处理，它无法处理数组返回null构造payload: `?username[]=1&password[]=2`

## 数组返回NULL绕过

首先是ereg这个正则匹配函数是处理字符串的。。。构造数组是返回null，然后的话null===false 不相等可以执行下面的else if语句。。。
strpos处理数组页数返回null,于是null!==false成立就得到了flag
payload: `?password[]=1`

## sha()函数比较绕过

关键部分还是===，只能用数组处理，利用了sha1函数处理数组返回null的特性。
payload: `?name[]=1&password[]=2`

## md5加密相等绕过

这一题里面是==，直接找一个MD5之后还是0e开头的即可，比如s878926199a
payload:

## 十六进制与数字比较

ord() 函数返回字符串的‘首个’字符的 ASCII 值。
利用这一个与题目的16进制，用16进制开头的0x去绕过即可。。。
用Python写个脚本转换一下

```
num = 3735929054
print ('%#x'%num)
```

- 1
- 2

得到`0xdeadc0de`
payload: `?password=0xdeadc0de`

## ereg正则%00截断

因为ereg存在%00漏洞，所以在第一个条件里面先构建一个截断绕过，然后就开始执行下面的else if语句，长度要小于8，数目要大于9999999，这里就用科学计数法，用1e8去绕过
下面还有一个strpos函数还要判断里面是否含有'-'符号，所以最后的payload应该是:?password=1e8%00–,出错了。。。

if (strpos ($_GET['password'], '-') !== FALSE) //strpos – 查找字符串
首次出现的位置



没办法只能找一下原题，发现人家的题目是要包含*-*，但是他题目就一条横杠。。。坑了不少人。。。
改一下payload: `?password=1e8%00*-*` ,立刻出flag
查了查手册，还有个类似的函数叫eregi，只是不判断大小写罢了

```
int eregi ( string $pattern , string $string [, array &$regs ] )
```

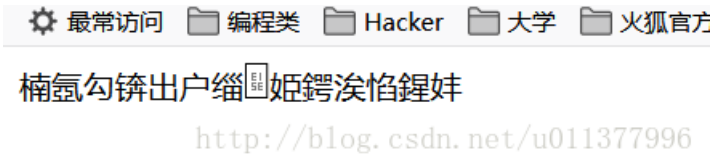
本函数和 ereg() 完全相同，只除了在匹配字母字符时忽略大小写的区别。



## strpos数组绕过

这一题跟那题数组返回NULL绕过，感觉好像是一样的。。。。。
就是传入的参数不一样了。。。。。
payload： `?ctf[]=1`

后来看了看下面的文章，发现还有一个方法2：字符串截断,利用ereg()的%00截断漏洞，绕过正则过滤 `?nctf=1%00#biubiubiu` 发现是一堆乱码，




楠氩勾铡出户缁姬鳄涘悒锃娃



这里需要特别注意， 需将#编码 `?nctf=1%00%23biubiubiu` 才能出flag。。。。

## 给一个适合新手的学习文章：

http://www.freebuf.com/articles/rookie/152209.html