BugkuCTF—代码审计 writeup

原创

Senimo_ 于 2019-08-02 19:42:32 发布 400 收藏 2

分类专栏： BugCTF writeup 文章标签： Bugku CTF writeup 代码审计 web

本文链接：https://blog.csdn.net/weixin_44037296/article/details/98229991

BugkuCTF—代码审计 writeup

- extract变量覆盖
- strcmp比较字符串
- urldecode二次编码绕过
- md5()函数
- 数组返回NULL绕过
- 弱类型整数大小比较绕过
- sha()函数比较绕过
- md5加密相等绕过
- 十六进制与数字比较
- 变量覆盖
- ereg正则%00截断
- strpos数组绕过
- 数字验证正则绕过
- 简单的waf

BugkuCTF链接

extract变量覆盖

分值：50

解题链接

```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';
    }
}
?>
```

分析代码：存在变量`flag`，使用extract()函数,通过GET方式传入的变量值导入到当前的符号表中，然后判断变量`shiyan`与变量`content`的值是否相等，相等则输出**flag**

利用**extract()函数**的变量覆盖漏洞，给变量`shiyan`与变量`content`赋空值，构造如下**payload**：`?shiyan=&content=`，通过地址栏传入参数，得到**flag**：`flag{bugku-dmsj-p2sm3N}`

strcmp比较字符串

分值50

解题链接

```php
<?php
$flag = "flag{xxxxx}";
if (isset($_GET['a'])) {
 //如果 str1 小于 str2 返回 < 0； 如果 str1大于 str2返回 > 0；如果两者相等，返回 0。
    if (strcmp($_GET['a'], $flag) == 0)
 //比较两个字符串（区分大小写）
        die('Flag: ' . $flag);
    else
        print 'No';
}
?>
```

分析代码：存在变量`flag`，通过GET方式传入变量`a`的值，如果变量`a`的值和变量`flag`的值相等便输出flag

通过**strcmp()函数**的比较漏洞，不能比较数组，且返回真，构造如下**payload**：`?a[]`，通过地址栏传参，得到**flag**：`flag{bugku_dmsj_912k}`

urldecode二次编码绕过

分值：50

解题链接

```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";
}
?>
```

分析代码：通过GET方式传递变量`id`的值，如果在变量`id`中匹配到字符串`hackerDJ`便退出程序，将传入的变量`id`的值进行**URL解码**，但变量`id`的值需要等于`hackerDJ`，才能输出flag。

因为在传入变量时浏览器会先进行一次**URL解码**，经过匹配判断后再进行一次**URL解码**，解码后的结果与`hackerDJ`相比较，所以将其中字符进行两次编码以绕过匹配，将字符`h`进行两次**URL编码**，得到如下**payload**：`?id=%2568ackerDJ`，通过地址栏传参，得到**flag**：`flag{bugku__daimasj-1t2}`

md5()函数

分值：50

解题链接

```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';
}
?>
```

分析代码：存在变量`flag`，通过GET方式传入变量`username`和变量`password`的值，两个变量的值不能相等，但经过**md5**函数加密后的值需相等,便输出**flag**

通过**md5函数**加密数组返回`NULL`，得到如下**payload**：`?username[]=1&password[]=2`，通过地址栏传入变量，得到**flag**：`flag{bugk1u-ad8-3dsa-2}`

数组返回NULL绕过

分值：50

解题链接

```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';
}
?>
```

分析代码：存在变量`flag`，通过GET方式传入变量`password`的值，经过正则表达式匹配，只允许输入`a-zA-Z0-9`，**strpos()函数匹配输入的变量中有字符串：`--`，便输出flag**

ereg()函数处理字符串判断用的是`===`，遇到数组做参数返回`NULL`，所以构造如下**payload**：`?password[]=--`，通过地址栏传递参数，得到**flag**：`flag{ctf-bugku-ad-2131212}`

弱类型整数大小比较绕过

分值：50

解题链接

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

分析代码：通过GET方式传入变量`password`的值，**is_numeric()函数**判断输入的值是否为纯数字，如果是，则返回`NULL`，如果输入的值大于`1366`，则输出**flag**

需要**绕过**is_numeric()函数**判断，并且通过弱类型比较大小时会将变量转化为相同类型这一特点，构造如下**payload**：`?password=1367a`,通过地址栏传参，得到**flag：`flag{bugku_null_numeric}`

sha()函数比较绕过

分值：60

解题链接

```
<?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!';
?>
```

分析代码：通过GET方式传入变量`name`和变量`password`的值，两个变量的值不能相等，但两个变量经过**sha1加密**后，需要相等，如果相等输出**flag**

通过**数组**的方式绕过**sha1()加密**后的判断，**sha1()加密**不能加密数组，加密数组会返回为：`NULL`，构造如下**payload**：`?name[]=1&password[]=2`，通过地址栏传参，得到**flag：`flag{bugku--daimasj-a2}`

md5加密相等绕过

分值：60

解题链接

```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";
}
?>
```

分析代码：通过GET方式传入变量`a`的值，将变量`a`的值进行md5加密后，与字符串`QNKCDZO`进行比较，两个值不能相等，但经过md5加密后的值需要相等，相等则输出flag

通过**弱类型比较**，部分字符串经过**md5加密**后会形成`0exxxxxx`的形式，在`==`弱类型比较中，会转换为`0`，所以通过构造如下**payload**：`?a=QLTHNDT`，通过地址栏传递参数，得到**flag**：`flag{bugku-dmsj-am9ls}`

十六进制与数字比较

分值：60

解题链接

```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);
?>
```

分析代码：存在变量`flag`，变量`number = 3735929054`，通过转码成ASCII码，进行判断，只能输入数字`1-9`

通过将数字串转换成十六进制，得到如下**payload**：`?password=0xdeadc0de`，通过地址栏传递参数，得到**flag**：`flag{Bugku-admin-ctfdaimash}`

变量覆盖

分值：60

解题链接

暂时无法打开。

ereg正则%00截断

分值：100

解题链接

```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';
    }
}
?>
```

分析代码：通过**GET**方式传入变量`password`的值，正则表达式限制为：`a-zA-Z0-9`，长度小于`8`，值大于`9999999`，且需要在变量`password`中匹配到`*-*`

尝试使用`%00`截断”**ereg()**“函数的正则表达式匹配，通过**科学计数法**绕过长度限制和大小比较，在地址栏构造如下**payload**：`?password=1e8%00*-*`，得到**flag**：flag{bugku-dm-sj-a12JH8}

strpos数组绕过

分值：150

解题链接

```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~';
}
?>
```

分析代码：存在变量`flag`，通过GET方式传入变量`ctf`的值，经过正则表达式匹配只能输入`1-9`，strpos()函数还需匹配到字符串`#biubiubiu`，条件成立则输出flag

使用`%00截断`绕过**ereg()函数**的正则表达式匹配，因为`#`会被**URL编码**，所以构造如下**payload**：`?ctf=1%00%23biubiubiu`，通过地址栏传递参数，得到**flag**：`flag{Bugku-D-M-S-J572}`

数字验证正则绕过

分值：150

解题链接

```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;
        $ps = array('punct', 'digit', 'upper', 'lower');
        //[[:punct:]] 任何标点符号 [[:digit:]] 任何数字 [[:upper:]] 任何大写字母 [[:lower:]] 任何小写字母
        foreach ($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;
    }
}
?>
```

转载数字验证正则绕过

简单的waf

分值：200

解题链接

暂时无法访问。