南邮CG-CTF—Web writeup第二部分

原创

Senimo_ 于 2019-08-04 16:03:31 发布 955 收藏 2

分类专栏： 各CTF平台 Writeup 文章标签： 南邮CG CTF writeup web

本文链接：https://blog.csdn.net/weixin_44037296/article/details/98236220

南邮CG-CTF—Web writeup第二部分

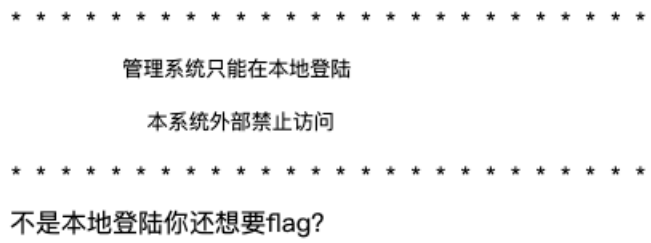
南邮CG-CTF链接

伪装者

Web 20pt

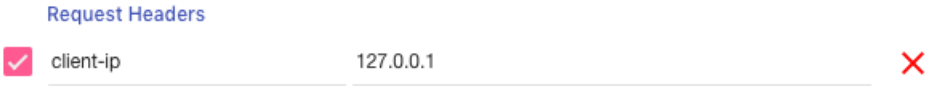
这是一个到处都有着伪装的世界

题目地址

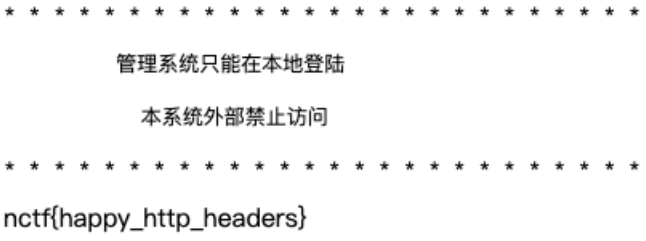


进入网页后，提示“**只能在本地登陆**”，在HTTP请求头中添加本地地址`client-ip: 127.0.0.1`

Google Chrome插件ModHeader



刷新页面后得到**flag**：



HTTP Referer是header的一部分，当浏览器向web服务器发送请求的时候，一般会带上Referer，告诉服务器该网页是从哪个页面链接过来的，服务器因此可以获得一些信息用于处理。

HTTP_CLIENT_IP 是代理服务器发送的HTTP头。

Header

Web 20pt

头啊！！头啊！！！

题目地址

题目暂时无法访问。

上传绕过

SQL注入1

Web 30pt

听说你也会注入？

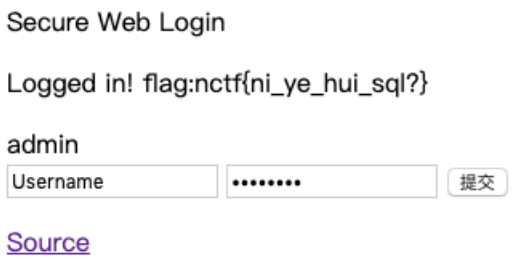
题目链接

进入页面后，用户名和密码已给出，提交后显示：`You are not admin!`，点击`Source`可以看到网页源码：

```php
//省略部分HTML代码
<?php
if ($_POST[user] && $_POST[pass]) {
    //省略部分源码
    $pass = md5(trim($_POST[pass]));
    $sql = "select user from ctf where (user='" . $user . "') and (pw='" . $pass . "')";
    echo '</br>' . $sql;
    $query = mysql_fetch_array(mysql_query($sql));
    if ($query[user] == "admin") {
        echo "<p>Logged in! flag:******************** </p>";
    }
    if ($query[user] != "admin") {
        echo("<p>You are not admin!</p>");
    }
?>
```

分析代码：当用户名（user）为“**admin**”且密码经过**md5加密**后判断相同，即为登陆成功。
源代码未经过过滤便从数据库中查询信息，尝试用**Google Chrome的插件HackBar**构造**万能密码**绕过密码判断：

提交数据后得到**flag**。



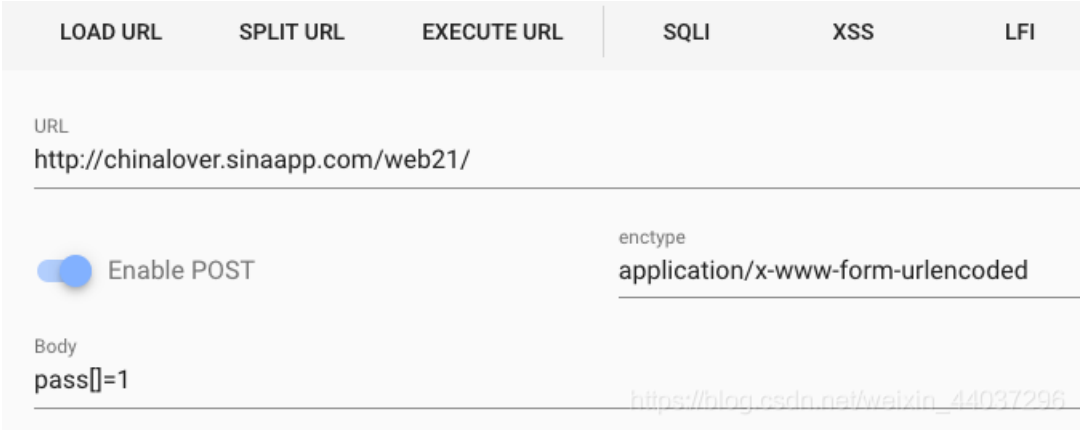
pass check

Web 30pt

题目地址

```php
//题目源码已经给出
<?php
$pass = @$_POST['pass'];
$pass1 =**********;//被隐藏起来的密码
if (isset($pass)) {
    if (@!strcmp($pass, $pass1)) {
        echo "flag:nctf{*}";
    } else {
        echo "the pass is wrong!";
    }
} else {
    echo "please input pass!";
}
?>
```

分析代码：通过**POST**方式传入变量**pass**的值，判断变量**pass**是否被设置，且**pass**要与**pass1**的值相等即输出**flag**的值。
可以利用**strcmp**函数的比较漏洞，即比较数组时，即可绕过比较但判断为**true**



在**Google Chrome插件HackBar**中构造传参：`pass[]=1`，提交数据得到flag。

flag:nctf{strcmp_is_n0t_3afe}

起名字真难

Web 30pt

题目地址

```php
//题目源码已给出
<?php
function noother_says_correct($number)
{
    $one = ord('1');
    $nine = ord('9');
    for ($i = 0; $i < strlen($number); $i++) {
        $digit = ord($number{$i});
        if (($digit >= $one) && ($digit <= $nine)) {
            return false;
        }
    }
    return $number == '54975581388';
}
$flag = '*******';
if (noother_says_correct($_GET['key']))
    echo $flag;
else
    echo 'access denied';
?>
```

分析代码：需要传入变量**key**的值，使**noother_says_correct**函数顺利执行，该函数判断长度小于**number**的长度，输入的值不能为数字“**1-9**”，函数执行成功则输出flag
因为函数限制了数字的输入，尝试转换**number**的编码以绕过函数的限制：

将**number**的值转为**16进制**时（在线进制转换），符号函数的限制，即**长度小于number**和不允许输入**数字“1-9”**：

```
http://chinalover.sinaapp.com/web12/index.php?key=0xccccccccc
```

在地址栏构造**GET**传参，得到**flag**.

The flag is:nctf{follow_your_dream}

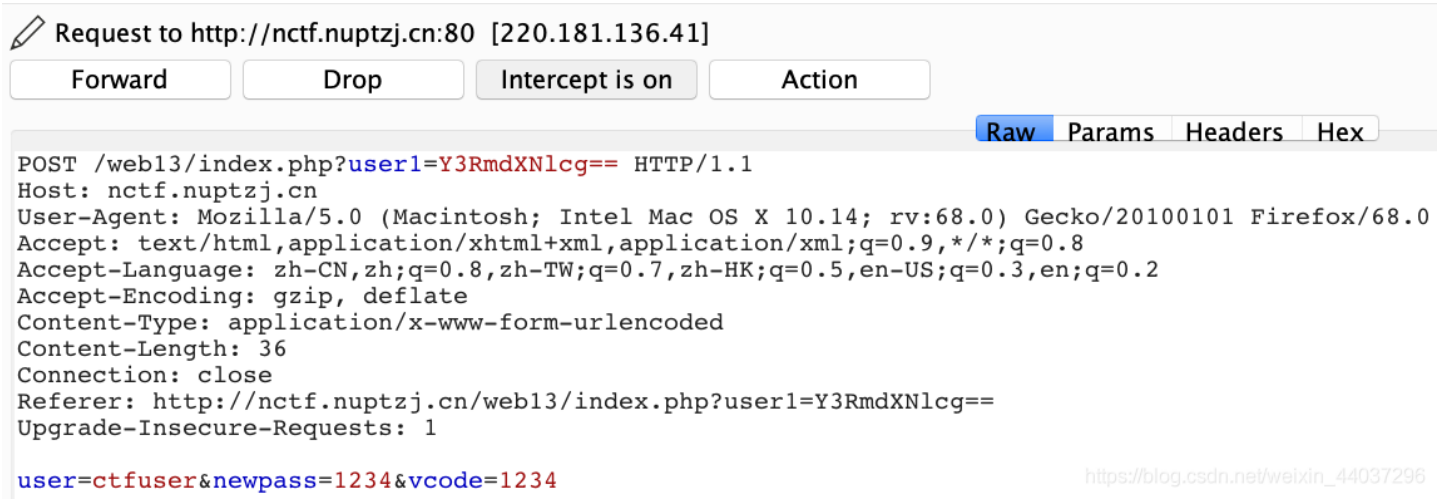
密码重置

Web 25pt
重置管理员账号：admin 的密码
你在点击忘记密码之后 你的邮箱收到了一封重置密码的邮件
题目地址

进入网页后，账号已经给出“**ctfuser**”，且不能更改，但提示为重置管理员账号密码,从网页源码中修改为“**admin**”，尝试重置密码后，提示“**error**”；

```
http://nctf.nuptzj.cn/web13/index.php?user1=Y3RmdXNlcg==
```

在地址栏发现**Base64**编码，在线Base64转码，显示为：`ctfuser`，将“**admin**”进行**Base64**编码，得到`YWRtaW4=`。



使用Burp Suite抓包修改数据，修改user1的值修改为`YWRtaW4=`及账号的user的值修改为`admin`，发送数据包后，得到flag。

php 反序列化

Web 40pt

题目地址

```php
<?php
class just4fun
{
    var $enter;
    var $secret;
}
if (isset($_GET['pass'])) {
$pass = $_GET['pass'];

if (get_magic_quotes_gpc()) {
    $pass = stripslashes($pass);
}
$o = unserialize($pass);
if ($o) {
    $o->secret = "*";
    if ($o->secret === $o->enter)
        echo "Congratulation! Here is my secret: " . $o->secret;
    else
        echo "Oh no... You can't fool me";
} else echo "are you trolling?";
?>
```

题目提示暂时无法做

SQL Injection

综合题

Web 50pt

题目地址

呀！这到底是什么玩意儿

```
[][(![]+[])[!![]+!![]+!![]]+({}+[])[+!![]]+(!![]+[])[+!![]]+(!![]+[])[+[]]][({}+[])[!![]+!![]+!![]+!![]+!![]]+({}+[])[+!!
[]]+({}[[]]+[])[+!![]]+(![]+[])[!![]+!![]+!![]]+(!![]+[])[+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+[]]+({}+[])[!![]+!![]+!![]+!!
[]+!![]]+(!![]+[])[+[]]+({}+[])[+!![]]+(!![]+[])[+!![]]](({}[[]]+[])[!![]+!![]]+({}+[])[+!![]]+({}+[])[!![]+!![]+!![]+!!
[]+!![]]+({}[[]]+[])[+[]]+[][(![]+[])[!![]+!![]+!![]]+({}+[])[+!![]]+(!![]+[])[+!![]]+(!![]+[])[+[]]][({}+[])[!![]+!![]+!!
[]+!![]+!![]]+({}+[])[+!![]]+({}[[]]+[])[+!![]]+(![]+[])[!![]+!![]+!![]]+(!![]+[])[+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+[]]+
({}+[])[!![]+!![]+!![]+!![]+!![]]+(!![]+[])[+[]]+({}+[])[+!![]]+(!![]+[])[+!![]]]((!![]+[])[+!![]]+(!![]+[])[!![]+!![]+!!
[]]+(!![]+[])[+[]]+({}[[]]+[])[+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]+!![]+!![]]+({}
[[]]+[])[+[]]+({}[[]]+[])[+!![]]+(!![]+[])[!![]+!![]+!![]]+(![]+[])[!![]+!![]+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]]+(![]+
[])[+!![]]+([]+[][(![]+[])[!![]+!![]+!![]]+({}+[])[+!![]]+(!![]+[])[+!![]]+(!![]+[])[+[]]][({}+[])[!![]+!![]+!![]+!![]+!!
[]]+({}+[])[+!![]]+({}[[]]+[])[+!![]]+(![]+[])[!![]+!![]+!![]]+(!![]+[])[+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+[]]+({}+[])[!!
[]+!![]+!![]+!![]+!![]]+(!![]+[])[+[]]+({}+[])[+!![]]+(!![]+[])[+!![]]]((!![]+[])[+!![]]+(!![]+[])[!![]+!![]+!![]]+(!![]+
[])[+[]]+({}[[]]+[])[+[]]+(!![]+[])[+!![]]+({}[[]]+[])[+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]+!![]+!![]]+(![]+[])[!![]+!!
[]]+({}+[])[+!![]]+({}+[])[!![]+!![]+!![]+!![]+!![]]+(![]+[])[+!![]]+(!![]+[])[+[]]+({}[[]]+[])[!![]+!![]+!![]+!![]+!![]]+
({}+[])[+!![]]+({}[[]]+[])[+!![]])())[!![]+!![]+!![]]+(!![]+[])[!![]+!![]+!![]])()([][(![]+[])[!![]+!![]+!![]]+({}+[])[+!![]]+
```

打开网页显示为**JSFuck编码**，**JSFuck**是一种基于**JavaScript**的编码方式，它只使用六个不同的字符来编写和执行代码。因为是基于JavaScript，所以可以在浏览器F12控制台直接执行：

执行后得到`1bc29b36f623ba82aaf6724fd3b16718.php`，尝试访问该地址，显示：“**哈哈哈哈哈哈你上当啦，这里什么都没有，TIP在我脑袋里**”，提示TIP在头文件里，查看HTTP请求头文件：

GET http://teamxlc.sinaapp.com/web3/b0b0ad119f425408fc3d45253137d33d/1bc29b36f62:	
Status: HTTP/1.1 200 OK	
Request Headers	
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Geckc
Response Headers	
Connection	keep-alive
Content-Encoding	gzip
Content-Type	text/html
Date	Sun, 04 Aug 2019 07:47:56 GMT
Server	nginx
tip	history of bash
Transfer-Encoding	chunked
Via	100142

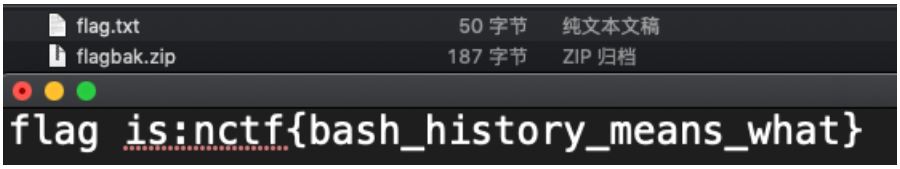
发现`tip: history of bash`，**Bash (GNU Bourne-Again Shell)** 是许多**Linux**发行版的默认**Shell**，所以提示为历史命令，在**Linux**中查询命令行历史命令的语法为：`vi.bash_hitsory`，尝试在地址栏输入此命令：

```
http://teamxlc.sinaapp.com/web3/b0b0ad119f425408fc3d45253137d33d/.bash_history
```

查询到历史命令：`zip -r flagbak.zip ./*`，下载flagbak.zip：

```
http://teamxlc.sinaapp.com/web3/b0b0ad119f425408fc3d45253137d33d/flagbak.zip
```

解压后的到flag。



system

SQL注入2

综合题2

密码重制2

file_get_contents

变量覆盖

Web 40pt

变量覆盖，代码审计类题目

题目地址

进入网页后为空白，查看网页源码得到提示的注释：

```
<!--foreach($_GET as $key => $value){
        $$key = $value;
}
if($name == "meizijiu233"){
    echo $flag;
}-->
```

分析代码：可变变量key获取了一个普通变量value的值作为这个可变变量的变量名。使用foreach来遍历数组中的值，再将获取到的数组键名作为变量，数组中的键值作为变量的值。因此就产生了变量覆盖漏洞。

```
http://chinalover.sinaapp.com/web24/?name=meizijiu233
```

通过地址栏进行GET传参，变量覆盖便形成了`$$key = $name`，`$value = meizijiu233`，构成了`$name == "meizijiu233"`，所以得到**flag**。

nctf{AD3FBD8D5928693CA499347C91570AE6}

知识点：变量覆盖

经常导致变量覆盖漏洞场景有：$$使用不当，extract()函数使用不当，parse_str()函数使用不当，import_request_variables()使用不当，开启了全局变量注册等。

注意！！

Web 1pt
再次重申，请不要未经同意便盗用我们的题目，如果有使用的需要，请和我们联系，联系方式已经在notice已经给出.
flag{zhaowomen}
强调一下版权问题，未经允许不要盗用题目，flag已经给出。

HateIT

Anonymous

Web 80pt
PHP是最好的语言，不是吗？
// SUCTF 2018，出题人：梅子酒
题目地址