

# 云演【ctf实验】

原创

error0 于 2022-01-21 12:23:10 发布 203 收藏

分类专栏: [刷题+WP](#) 文章标签: [php](#) [开发语言](#) [后端](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_51295677/article/details/122329106](https://blog.csdn.net/qq_51295677/article/details/122329106)

版权



[刷题+WP 专栏收录该内容](#)

12 篇文章 1 订阅

订阅专栏

无聊随便做了几个题目

目录

[矛盾](#)

[单身一百年也没用](#)

[单身20年的手速](#)

[ereg](#)

[intval](#)

---

## 矛盾

```
<?php
$num=$_GET['num'];
if(!is_numeric($num))
{
echo $num;
if($num==1)
echo 'flag{*****}';
}
?>
```

正常给了一串代码, 正常思维下会觉得有矛盾, 矛盾点在于函数 `is_numeric()` 需要使 `num` 不为整形数字。但 `num==1`, 这是一个矛盾应该如何绕过就是需要探讨的。

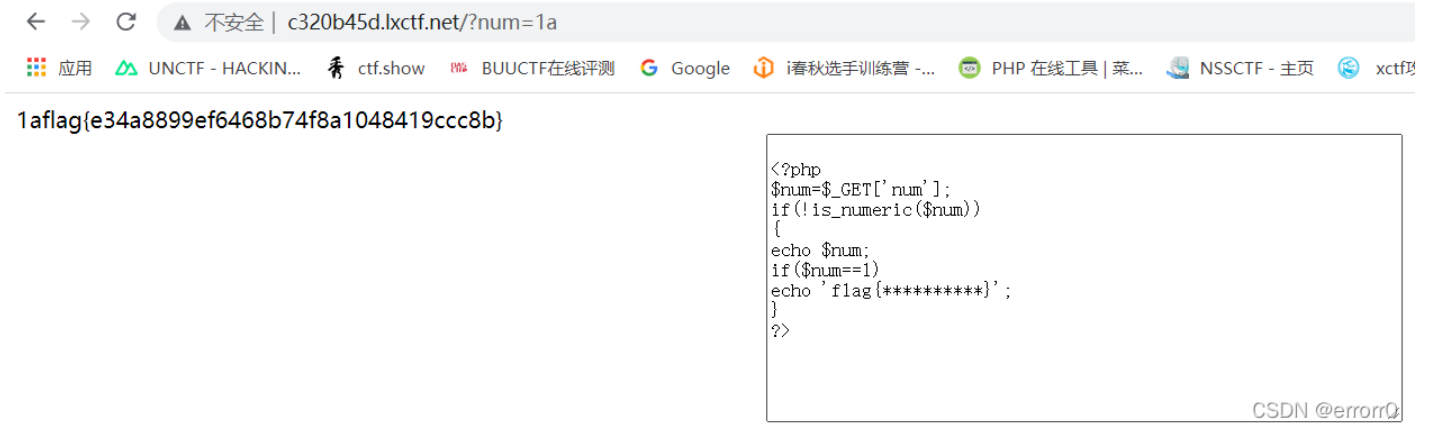
首先，这是一个num与1的弱比较，而弱比较与强比较的区别在于：

当php中数字与字符串作比较或者进行运算时，php会先将字符串转换为数字再比较。

其中php的转换规则为：如果字符串是以数字开头，则取开头的数字为转换结果。若无数字则取0。

PHP一个数字和一个字符串进行比较或者进行运算时，PHP会把字符串转换成数字再进行比较。PHP转换的规则是：若字符串以数字开头，则取开头数字作为转换结果，若无则输出0。在PHP中，== 会先进行类型转换，再进行对比，而===会先比较类型，如果类型不同直接返回false。

根据上述弱比较的转换方法构造payload为?num=1+其它的字符即可绕过。



← → ↻ 不安全 | c320b45d.lxctf.net/?num=1a

应用 UNCTF - HACKIN... ctf.show BUUCTF在线评测 Google i春秋选手训练营... PHP 在线工具 | 菜... NSSCTF - 主页 xctf

1aflag{e34a8899ef6468b74f8a1048419ccc8b}

```
<?php
$num=$_GET[' num '];
if(!is_numeric($num))
{
echo $num;
if($num==1)
echo 'flag{*****}';
}
?>
```

CSDN @errorrQ

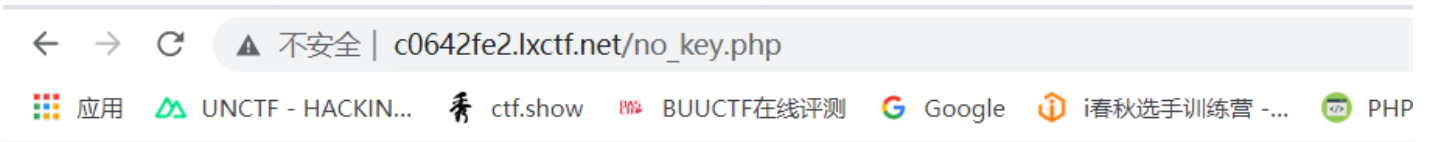
单身一百年也没用

[到这里来找key](#)

```
元素 控制台 源代码 网络 性能 内存 应用 安全 Lighthouse
<!DOCTYPE >
<html>
  <head>...</head>
  ... <body> == $0
    <a href="search_key.php">到这里来找key</a>
  </body>
</html>
```

CSDN @errorr0

可以看到点击链接会跳转到search\_key.php下，但点击后发现

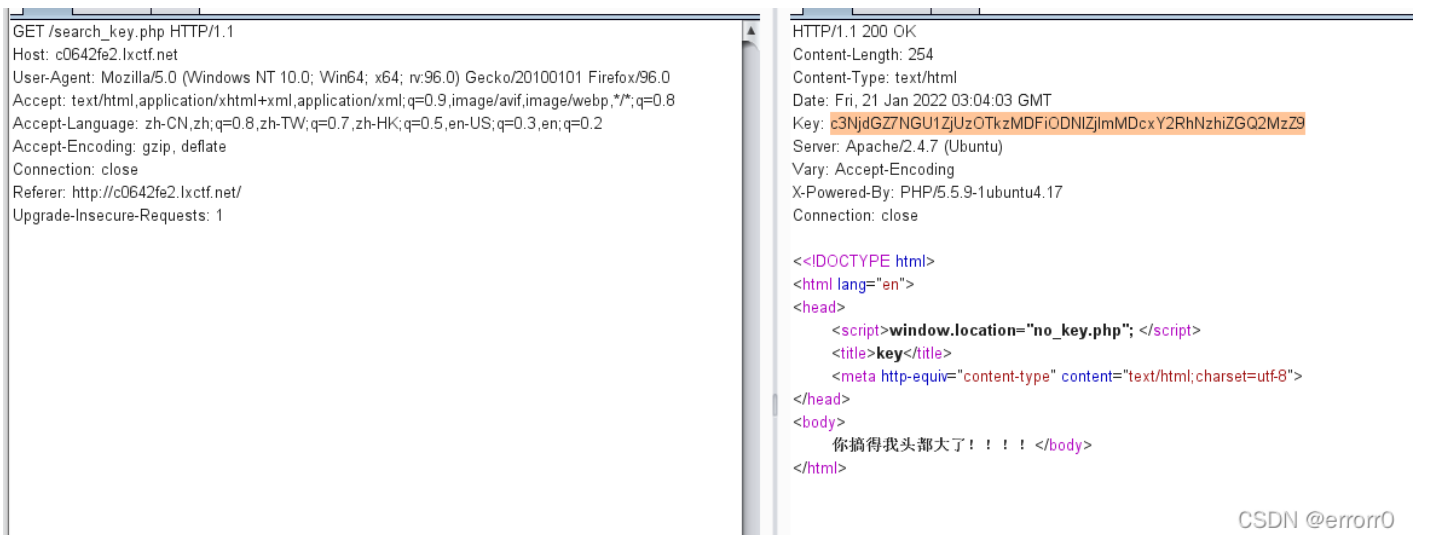


这里真的没有KEY，军爷说的，军爷从来不坑人~~~



CSDN @errorr0

跳到了no\_key.php下，说明在目标链接的内容中有跳转，所以用bp抓包阻止跳转



CSDN @errorr0

最后发现key，即目标flag

## 单身20年的手速

和上一题没什么不同，且它的flag完全没藏

```
GET /search_key.php HTTP/1.1
Host: 4b2e9f29.lxcdf.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:96.0) Gecko/20100101 Firefox/96.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://4b2e9f29.lxcdf.net/
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Content-Length: 183
Content-Type: text/html
Date: Fri, 21 Jan 2022 03:12:00 GMT
Server: Apache/2.4.7 (Ubuntu)
Vary: Accept-Encoding
X-Powered-By: PHP/5.5.9-1ubuntu4.17
Connection: close
```

```
<<!DOCTYPE html>
<html lang="en">
<head>
  <script>window.location="2.php"; </script>
  <title>key</title>
</head>
<body>
  sscff[e32f79d81ff22f0f929f94d9424ca5eb]</body>
</html>
```

CSDN @errorr0

## ereg

### 前置知识

ereg()函数用指定的模式搜索一个字符串中指定的字符串,如果匹配成功返回true,否则,则返回false。搜索字母的字符是大小写敏感的。

ereg函数存在NULL截断漏洞,导致了正则过滤被绕过,所以可以使用%00截断正则匹配

### 开局一片白,做题全靠扫

```
[11:18:28] Starting:
[11:18:31] 403 - 296B - /.ht_wsr.txt
[11:18:31] 403 - 299B - /.htaccess.save
[11:18:31] 403 - 299B - /.htaccess.bak1
[11:18:31] 403 - 301B - /.htaccess.sample
[11:18:31] 403 - 299B - /.htaccess.orig
[11:18:31] 403 - 299B - /.htaccess_orig
[11:18:31] 403 - 300B - /.htaccess_extra
[11:18:31] 403 - 297B - /.htaccess_sc
[11:18:31] 403 - 289B - /.htm
[11:18:31] 403 - 298B - /.htaccessOLD2
[11:18:31] 403 - 297B - /.htaccessBAK
[11:18:31] 403 - 297B - /.htaccessOLD
[11:18:31] 403 - 290B - /.html
[11:18:31] 403 - 299B - /.htpasswd_test
[11:18:31] 403 - 296B - /.httr-oauth
[11:18:31] 403 - 295B - /.htpasswd
[11:18:32] 403 - 289B - /.php
[11:18:32] 403 - 290B - /.php3
[11:18:50] 200 - 0B - /index.php
[11:18:50] 200 - 0B - /index.php/login/
[11:18:57] 200 - 440B - /robots.txt
[11:18:57] 403 - 298B - /server-status
[11:18:57] 403 - 299B - /server-status/
[11:18:59] 200 - 113B - /start.sh CSDN @errorr0
```

发现源码放在robots.txt里

```
if (isset ($_GET['password'])) {  
  
    if (ereg ("^[a-zA-Z0-9]+$", $_GET['password']) === FALSE)//如果password不为数字或者字母  
    {  
        echo '<p>You password must be alphanumeric</p>';  
  
        }  
        else if (strlen($_GET['password']) < 8 && $_GET['password'] > 9999999)//password长度小于8,key值大于9999999  
        {  
  
            if (strpos ($_GET['password'], '*-*') !== FALSE)// 判断password参数值是否出现*-*  
            {  
                die('Flag: ' . $flag);  
            }  
            else  
            {  
                echo('<p>*-* have not been found</p>');  
            }  
        }  
        else  
        {  
            echo '<p>Invalid password</p>';  
        }  
    }  
}
```

通过分析最后构造payload为?password=1e9%00\*-\*

参考: [四叶草云演-CTF03# ereg\\_weixin\\_43973521的博客-CSDN博客](#)

---

## intval

前置知识--int溢出

当前 PHP 版本支持的最大整型数字。在 32 位系统中通常为 int(2147483647)，64 位系统中为 int(9223372036854775807)。自 PHP 5.0.5 起可用。

```

<?php
if(!isset($_GET['source'])){
    highlight_file('index.php');
    die();
}
include('flag.php');
$key1 = $_GET['f'];
$key2 = $_GET['l'];
$key3 = $_GET['a'];
$key4 = $_GET['g'];
if(isset($key1)&&isset($key2)&&isset($key3)&&isset($key4))
{
    if(intval($key1) > 1 || intval($key1) < 0)
        die("key1 is error.");
    elseif(intval(intval($key1)) < 1)
    {
        if($key1 == 1){
            if($key2 < 1){
                die("key2 is error.");
            }else{
                if(intval($key2 + $key1) > 1){
                    die("key is error.");
                }else{
                    $check = is_numeric($key3) and is_numeric($key4);
                    if(!$check){
                        die("key3 or key4 is error.");
                    }elseif(!(is_numeric($key3) and is_numeric($key4))){
                        $key3 = $flag;
                        $key4 = $redpacket;
                    }
                    die("flag:".$key3."<br>".$key4);
                }
            }
        }
        else
            die("key1 is error.");
    }
    else
        die("key1 is error.");
}
?>

```

经过分析：

\*对于key1而言需要构造一个使intval(key1)<1且key1==1 -->这里主要针对intval下手

方法: 可以用16进制的0x绕过intval,0x1被intval翻译过来等于0, 但实际就是1

\*对于key2而言大于等于1的同时, 加上key1要小于等于1 -->这里会用到int最大长度溢出

当溢出后会使得二进制原本管理正负号的位数变动, 导致原本的正好成负号

方法: 利用最大int(9223372036854775807)在加1以后为负

\*而key3与key4需要在检测变量是否为数字或数字字符串的同时有一个或两个都为null -->这里是代码优先级出现了问题

方法: 利用=与and的优先级对g置空即可

最后构造payload为: ?f=0x1&l=9223372036854775807&a=1&g=&source=1