php异或绕过,CTF中php异或绕过preg_match

转载

伍世棋 于 2021-04-03 17:34:10 发布 725 收藏
文章标签： php异或绕过

0x00:写在前面

suctf的题目和强网杯都遇到这种类型题目了，正好就当做一个笔记来记录一下。

```
$hhh= @$_GET['_'];if(!$hhh){

highlight_file(__FILE__);

}if(strlen($hhh)>18){

die('One inch long, one inch strong!');

}if ( preg_match('/[\x00- 0-9A-Za-z\'"\`~_&.,|=[\x7F]+/i', $hhh) )

die('Try something else!');

$character_type= count_chars($hhh, 3);if(strlen($character_type)>12) die("Almost there!");

eval($hhh);

?>
```

代码节选~

这里有两个需要绕过地方

1：传入字符长度可以构造$_GET[x]来绕过

2:preg_match可以通过异或来绕过

0x01:思路

异或在开发中可以用来某些场景代替if判断

```
$a = 1^ 1; //0

$a= 0^ 0; //0

$a= 1^ 0; //1$a= 0^ 1; //1
```

返回0或者1

首先看这个正则

```
/[\x00- 0-9A-Za-z\'"\`~_&.,|=[\x7F]+/i
```

\xnn 匹配ASCII代码中十六进制代码为nn的字符

[x00-x7f] 匹配ASCII值从0-127的字符

0-127表示单字节字符，也就是：数字，英文字符，半角符号，以及某些控制字符。

嗯，反正不是中文

0-127的ascii[0-127]的字符 数字 字母 一些字符等等都被过滤了

绕过原理

以制作免杀马为例:

在制作免杀马的过程，根据php的语言特性对字符进行!运算会将字符类型转为bool类型,而bool类型遇到运算符号时,true会自动转为数字1,false会自动转为数字0,如果将bool类型进行计算,并使用chr()函数转为字符,使用"."进行连接,便可以绕过preg_match匹配。

详情了解php不同于其他语言部分

但是很多的preg_match会过滤掉".",所以需要使用异或运算进行绕过,很多的免杀马都是这样制作的。php对字符进行异或运算是先将字符转换成ASCII码然后进行异或运算,并且php能直接对一串字符串进行异或运算,例如"123"^"abc"是"1"与"a"进行异或然后"2"与"b"进行异或,以此类推,在异或结束后就获得了想要的字符串。

注意点:进行异或运算时要将数字转换成字符形式,如果数字(int)和字符异或的话,结果只会是数字,例如1^"a"=1,"a"^2=2,将数字转换成字符串可以使用trim()函数。

拓展:

php特性use of undefined constant,会将没有引号的字符都自动视为字符串,ASCII码大于0x7F的都会被当作字符串,由此可知可以简化异或过程,任何字符与0xff异或都会取相反,这样就能减少运算量了。

以GET或POST传入字符绕preg_match为例:

php的eval()函数在执行时如果内部有类似"abc"^"def"的计算式，那么就先进行计算再执行，我们可以利用再创参数来实现更方便的操作,例如传入?a=$_GET[b],由于b不受限制就可以任意传值了,不过

注意1:在测试过程中发现问题，类似phpinfo();的,需要将后面的();放在第个参数的后面,例如url?a={_GET}{b}();&b=phpinfo,也就是?a=${%ff%ff%ff%ff^%a0%b8%ba%ab}{%ff}();&%ff=phpinfo,在传入后实际上为${????^????}{?}();但是到了eval()函数内部就会变成${_GET}{?}();成功执行。

注意2:测试中发现,传值时对于要计算的部分不能用括号括起来,因为括号也将被识别为传入的字符串,可以使用{}代替,原因是php的use of undefined constant特性,例如${_GET}{a}这样的语句php是不会判为错误的,因为{}使用来界定变量的,这句话就是会将_GET自动看为字符串,也就是$_GET['a']

github关于这部分的讲解

https://github.com/Samik081/ctf-writeups/blob/master/ISITDTU%20CTF%202019%20Quals/web/easyphp.md

所以综上所述，绕过长度限制可以传一个$_GET[x] 异或来绕过preg_match

0x02:payload

这里可以用fuzz脚本fuzz出来 _GET

url?_=${%fe%fe%fe%fe^%a1%b9%bb%aa}{%fe}();&%fe=phpinfo

url?_=${%ff%ff%ff%ff^%a0%b8%ba%ab}{%ff}();&%ff=phpinfo

a=${%ff%ff%ff%ff^%a0%b8%ba%ab}{%ff}();&%ff=phpinfo,在传入后实际上为${????^????}{?}();但是到了eval()函数内部就会变成${_GET}{?}();成功执行

0x03:参考

https://www.cnblogs.com/cimuhuashuimu/p/11546422.htmlhttps://www.jianshu.com/p/fbfeeb43ace2