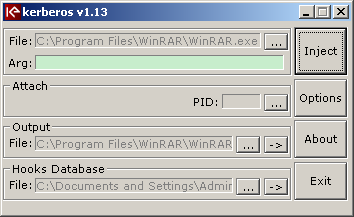
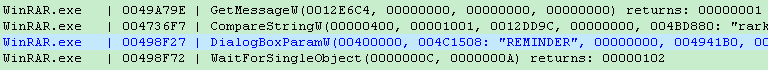
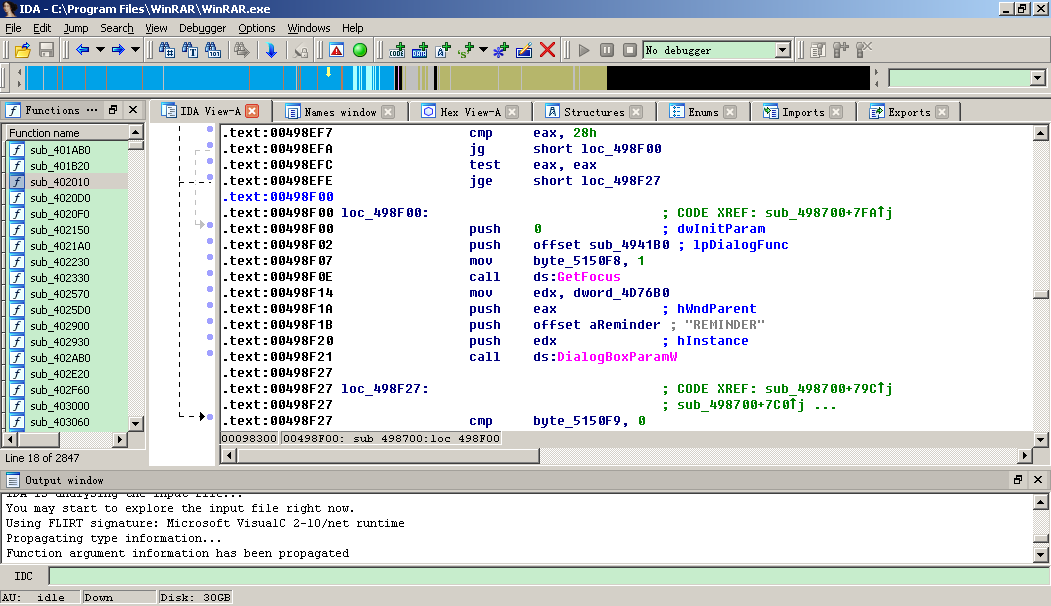
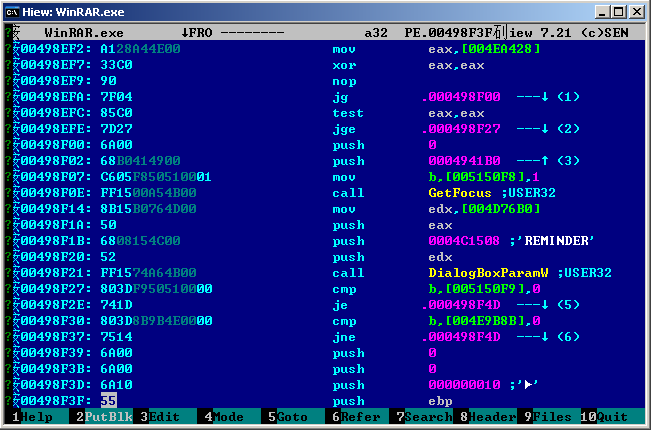
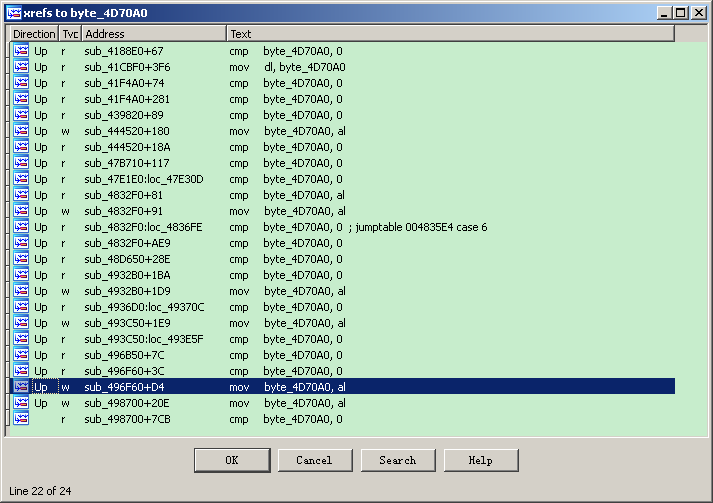
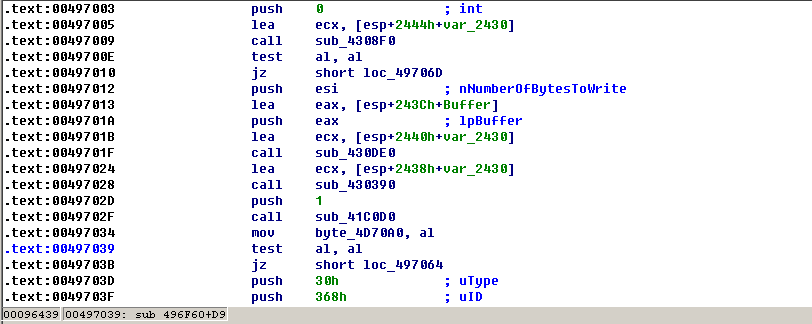
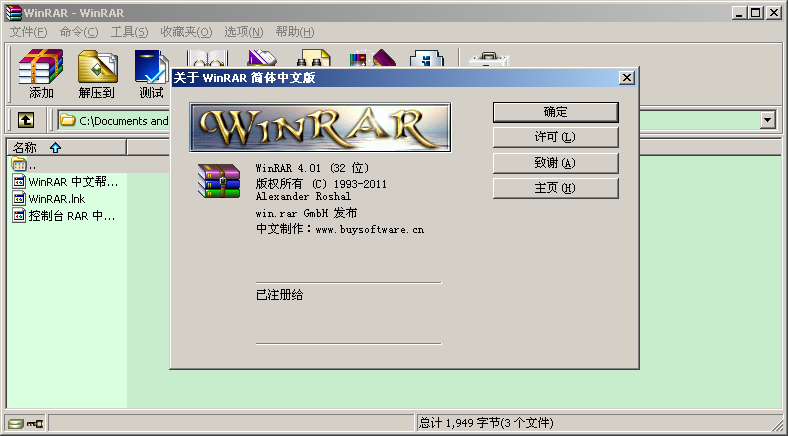
WinRAR是我们平时最常用到的软件之一，由于它并非免费软件，试用期过后，每次打开winRAR都会提示请注册的对话框，令人烦不胜烦。于是，我一般的做法是下载低版本的被破解后的winrar使用（最新版的WinRAR往往破解较慢），所以如果你想尝鲜的话，就不得不等一阵子了。  
最近看了《黑客反汇编解密（第二版）》，其中介绍了破解WinRAR的基本方法，不过书中所述版本较老，是WINRAR3.42版的。根据文中所述方法，结合自己理解，我破解了目前最新的4.01版WinRAR，不敢独享，特记录破解方法，以飨读者。  
  
所需工具  
1. IDAPro  
  目前最好的静态反汇编工具，黑客必备，可惜是收费软件，当然也有破解的，就是版本比较低。本人用的是最新的版本的试用版，呵呵，基本功能都有，所以也就凑合用了。  
  
2. HIEW  
  著名的一款十六进制编辑工具，小巧，但是功能强大，而且支持反汇编显示文件，并可以直接使用汇编语言修改原始文件，非常方便。  
  
3. Kerberos API  
  一款API窥测工具，可以记录软件运行过程中调用过的所有API函数，使用它可以帮助我们快速定位程序关键点。  
  
  
破解方法  
  
  首先去华军软件园下载WINRAR4.01版，安装好之后，将当前系统时间先调整到40天之后。因为WinRAR安装完成之后，可以免费试用40天，试用期过后，它会弹出请购买的对话框。这个对话框就是我们破解的突破口。  
   我们希望使用Kerberos API的窥测功能定位到弹出请购买对话框的系统API，与对话框相关的API有很多，比如CreateDialog，DialogBox与MessageBox等都是这样的函数。WinRAR开发人员用的是哪一个函数呢？为了搞清楚，就需要Kerberos API探测器的帮忙了。  
   在使用Kerberos API窥测器之前，请先配置一下过滤器选项，以便让它将不能提供有用信息的API调用丢弃掉。打开ke\_spy.txt文件，然后将以下这些函数注释掉：TlsGetValue，DefWindowProcA，DispatchMessageA, GetFocus, GetMessageA, SendMessageA, SendMessageW, TranslateAcceleratorA, TranslateAcceleratorW 与TranslateMessage。在函数名之前插入分号（；），就可以注释掉一个函数。要提升过滤性能，可以单击“Option（选项）”按钮，并设置“Report only .exe calls（仅仅报告.exe调用）”复选框。这样程序仅收集从winrar.exe发出的api调用，而不收集从加载的dll中发出的调用。应该注意的是，如果没有这么做，也不会发生什么错误。不过，要是不这样做，报告文件的尺寸将会很大而难以分析。  
   
  现在，单击“Browse”按钮，指定WinRAR的文件路径，然后单击“Inject(注入)”按钮。此时窗口会呈现下图所示情况。   
  
呵呵，用Kerberos注入WinRAR之后，一幅光秃秃的窗口出现在我们眼前，此时不要认为出现了什么错误，只要耐心等那么2-3秒，关闭WinRAR即可，并打开winrar.rep文件，该文件保存在WinRAR的安装目录下。  
   考察报告文件最方便的方法是从文件的结尾处入手。这是因为让用户注册的弹出窗口会在接口被初始化后才最后一个显示出来。因此，你会比较容易找到那个针对DialogBoxParamW函数进行的调用，这个函数负责显示“REMINDER”对话框。它就是创建让用户注册弹出窗口的函数。  
   下图便是Kerberos API窥测器生成的报告文件的片段  
  
   
Kerberos API窥测器甚至显示了从那个函数返回的地址--------00498F27，它可以立即将黑客引向保护代码。在反汇编器中查看这个代码。启动IDA Pro，加载winrar.exe文件，然后按“G”（跳转到某个地址），指定返回地址（00498F27）并按“Enter”键。如下图所示：  
  
   
你可以显而易见地看到DialogBoxParamW函数的调用，在它之上是如下所示的反汇编代码：  
.text:00498E95                 cmp     dword\_4EA434, 0 ; jumptable 00498E21 case 2  
.text:00498E9C                 jnz     loc\_498F27  
.text:00498EA2                 push    400h  
.text:00498EA7                 lea     eax, [esp+0BB0h+WideCharStr]  
.text:00498EAE                 push    eax  
.text:00498EAF                 mov     ecx, offset unk\_4F0DD0  
.text:00498EB4                 call    sub\_4130A0  
.text:00498EB9                 cmp     byte\_5150F8, 0  
.text:00498EC0                 jnz     short loc\_498F27  
.text:00498EC2                 cmp     byte\_4D446F, 0  
.text:00498EC9                 jnz     short loc\_498F27  
.text:00498ECB                 cmp     byte\_4D70A0, 0  
.text:00498ED2                 jnz     short loc\_498F27  
.text:00498ED4                 push    6               ; int  
.text:00498ED6                 push    offset aRarkey  ; "rarkey"  
.text:00498EDB                 lea     ecx, [esp+0BB4h+WideCharStr]  
.text:00498EE2                 push    ecx  
.text:00498EE3                 call    sub\_451C20  
.text:00498EE8                 push    eax             ; lpString1  
.text:00498EE9                 call    sub\_473650  
.text:00498EEE                 test    eax, eax  
.text:00498EF0                 jz      short loc\_498F27  
.text:00498EF2                 mov     eax, dword\_4EA428  
.text:00498EF7                 cmp     eax, 28h  
.text:00498EFA                 jg      short loc\_498F00  
.text:00498EFC                 test    eax, eax  
.text:00498EFE                 jge     short loc\_498F27  
.text:00498F00  
.text:00498F00 loc\_498F00:                             ; CODE XREF: sub\_498700+7FA j  
.text:00498F00                 push    0               ; dwInitParam  
.text:00498F02                 push    offset sub\_4941B0 ; lpDialogFunc  
.text:00498F07                 mov     byte\_5150F8, 1  
.text:00498F0E                 call    ds:GetFocus  
.text:00498F14                 mov     edx, dword\_4D76B0  
.text:00498F1A                 push    eax             ; hWndParent  
.text:00498F1B                 push    offset aReminder ; "REMINDER"  
.text:00498F20                 push    edx             ; hInstance  
.text:00498F21                 call    ds:DialogBoxParamW  
  可以看到，只是在执行到如下两条语句时进行跳转。  
.text:00498EF7                 cmp     eax, 28h  
.text:00498EFA                 jg      short loc\_498F00  
如果eax>28h，就跳转，跳转到DialogBoxParamW函数处。28h对应的十进制数是40.这个值指定的是试用期的持续天数。到此为止，变量dword\_4EA428的物理含义已经十分清楚了：自从程序安装以来所过去的天数。  
   保护机制已经找到了。接下来做点什么呢？例如，要阻止“提示注册”窗口的出现，可以用xor eax,eax/nop(33 C0/90)替换cmp eax, 28h(83 F8 28)。这样一来，不管当前的日期是什么，eax寄存器的值总是零。加入nop指令是为了弥补指令长度（由于cmp指令占据3个字节，而xor指令只有两个字节长，指令替换后长度变短了）。  
   启动HIEW，加载winrar.exe，初始界面给人的感觉像是乱码，不过按两次“Enter”键之后便可以切换到汇编模式。然后，按“F5”键（到……去），指定cmp指令的地址：. 498EF7。注意，开头的句点（.）不能少，它用来告诉HIEW，这里给定的值是个地址，而不是文件之中的偏移量。按“F3”键切换到汇编编辑模式，然后按“Enter”键以指定要执行的汇编指令。接下来，会显示一个对话框。输入xor eax,eax，<回车>，nop，<回车>，按F9键将结果保存在文件之中，如下图所示，然后退出。  
   
现在，启动WinRAR看看会出现什么情况。这一次，恼人的提示注册的窗口没有再次出现。整个破解过程用了不到10分钟。  
  
强行注册  
虽然提示注册的窗口成功阻止了，但是软件并没有注册。窗口标题会诚实地提示你，这是一个评估版本软件。如果从“Help”命令菜单选取“About”命令，软件会提示这是一个只有40天的试用版软件。虽然试用版在功能上没有受到限制，但是心里难免感觉有些别扭。  
WinRAR软件的注册是通过一个含有数字签名的密钥文件来实现的，这种数字签名通过加密方式生成，以确保篡改的密钥无效。所以，伪造密钥文件几乎是不可能的，不过我们真正想要做的事情是设置注册标志。如何找到这个标志，不妨回到cmp eax,28h这条关键指令的上面去看看。  
我们重点关注cmp eax,28h这条指令之上的几条指令的参数。你也许会问，我怎么会知道要去看哪个参数。实际上，我也不知道哪个参数才是破解的关键点，这需要很多猜测和尝试。当然也不是胡乱猜测，我们可以借助IDA Pro的交叉参考功能，理解参数的作用。所谓交叉参考，就是查看代码段中那些指令调用了本条指令或者是参数。  
在IDA中，将光标移到变量名处，打开上下文菜单，选择“jump to xref(跳到交叉引用)”或者简单地按下“X”键，就会出现如下图所示界面。  
  
整个程序中散布着许许多多的执行读（r）写（w）操作的交叉引用。选中交叉引用列表中的某一行，然后双击鼠标左键，即可跳转到代码中相应位置。  
这里，我就不列举我尝试错误的参数了，直接告诉大家关键参数在哪里吧，实际上就是这句  
.text:00498ECB                 cmp     byte\_4D70A0, 0  
其中byte\_4D70A0参量的交叉引用，就是上图所示的情形，而在倒数第三行的写操作那里双击鼠标，来到相关代码段：  
   
其中00497034地址处的指令便是一处byte\_4D70A0参量的引用地点，看到如下几条指令：  
.text:0049702F                 call    sub\_41C0D0  
.text:00497034                 mov     byte\_4D70A0, al  
.text:00497039                 test    al, al  
.text:0049703B                 jz      short loc\_497064    ；继续注册  
首先调用sub\_41C0D0函数，然后比较返回值，等于零时跳转，实际上sub\_41C0D0便是进行注册认证的函数所在了，有兴趣的读者可以再跟进去看看它是怎么认证的，我这里就不演示了。  
现在，我们只要让sub\_41C0D0函数非零值，就可以成功注册了。  
启动HIEW，按两次回车键，切换到反汇编模式。按F5键，给出.41C0D0值，这个值就是认证注册函数的首地址，然后按F3键切换到编辑模式。输入如下指令：<回车>xor eax,eax <回车>inc eax<回车>retn<Esc>（先将eax寄存器重置为0，然后加1再退出函数）。按F9键将结果保存在文件中，然后退出HIEW。  
启动WinRAR看看情况如何。可以看到，评估版本字符串已经从窗口的标题中消失了。关于对话框中会显示“注册到”字符串，如下图所示：  
   
至此，完美破解WinRAR4.01。  
  
  
参考文献：  
《黑客反汇编揭秘（第二版）》