恶意网页防护系统的设计与实现

吴 星,陈明锐
(海南大学 信息科学技术学院,海南 海口 570228)

摘 要:恶意网页是指包含恶意脚本的网页,恶意脚本可通过修改用户计算机的注册表和加载木马等形式进行各种破坏活动,其危害性日趋严重。为防止恶意网页所带来的危害,保护用户计算机安全,设计了一种恶意网页防护系统,该系统从两个技术环节对恶意网页进行防御。首先从浏览器获得网页地址,抢先读取网页源代码,并以特征码分析法阻止恶意脚本的执行;接着监控注册表的修改行为,阻止恶意脚本修改注册表以获得控制权。两个环节相辅相成,共同构成较为完善的防护体系。实践表明,该系统对恶意网页具有较好的防护效果。

关键词:恶意网页; 浏览器监控; 注册表监控; 钩子; 线程
中图分类号:TP309.2 **文献标识码**:A **文章编号**:1673-629X(2009)05-0154-04

Design and Realization of System for Defending Malicious Web Pages

WU Xing, CHEN Ming-rui
(College of Information Science and Technology, Hainan University, Haikou 570228, China)

Abstract: Malicious web page is a page that includes malicious scripts, and malicious scripts will destroy user's computer system by means of revising registry or injecting Trojans. The harm of malicious web pages to computers is more and more serious. In order to protect user's computer system from malicious web pages, a defending system is designed and realized. The system holds back malicious web pages by two technology links. First of all, it gets the source code of a web from browser and carries on analyzing by the way of condition code. If it finds out malicious scripts in the web, it will prevent them from running. And then, the system will monitor the registry and prevent it from being revised by malicious scripts. The two links supplement each other to constitute a comparatively perfect defending system. Practice has showed that the system has good effect in defending malicious web pages.

Key words: malicious web page; browser monitoring; registry monitoring; hook; thread

0 引 言

恶意网页是指包含恶意脚本(具体有 JavaScript、VBScript、Java Applet 和 ActiveX 控件等形式)的网页,当用户浏览这些网页时,其中的恶意脚本便像幽灵一般悄悄潜入。一旦入侵成功,就会通过修改用户机器的注册表和加载木马程序等方式进行各种破坏活动。其危害性主要表现为:不断消耗用户系统资源,使用户系统无暇处理其他事务;通过木马程序非法读取用户隐私数据或者诱骗用户交出隐私数据(如用户名和密码),造成信息泄露;破坏用户系统,如格式化硬盘,使用户系统瘫痪,等等,给用户带来极大的甚至是灾难性的危害。

收稿日期:2008-09-02
基金项目:海南省教育科研项目(Hj200724)
作者简介:吴 星(1972-),男,海南文昌人,副教授,研究方向为计算机与网络安全;陈明锐,教授,研究方向为电子商务、管理信息系统。

要防止恶意网页的破坏,最直接的办法是当用户浏览到这种网页时,用一个浏览器监控程序以特征码方式识别出它的恶意性,在其中的恶意脚本被执行之前进行拦截。如果恶意脚本用变种或加密的方式蒙混过关,或者是一种新型恶意脚本被放行,接下来它们将试图修改用户机器的注册表,有的还同时往用户机器的系统文件夹中写入恶意文件,这时可以使用一个注册表监控程序监视其操作注册表的行为,向用户发出警告,并将注册表恢复回原先状态。浏览器监控模块和注册表监控模块共同构成了恶意网页防护系统,它们从特征码分析和行为监控两个技术环节实现了防护恶意网页的目的[1]。文中将以 Microsoft Windows XP 和 Microsoft Internet Explore 6.0 环境为例,具体阐述该系统的实现过程,编程工具采用 Visual Basic 6.0。

1 系统整体思路

当用户在 IE 浏览器中打开一个网页时,如果该网页的源代码中存在恶意脚本,则在瞬间恶意脚本就会

被用户机器的 WSH(Windows Script Host,Windows 脚本宿主)执行,恶意脚本入侵成功。所以,为防止恶意脚本被执行,设置一个带有键盘钩子和鼠标钩子的模块,监视 IE 浏览器中的按键与鼠标消息。一旦监视到按键或鼠标消息,立即判断是否是打开网页操作。如果是打开网页的操作,则拦截该消息,抢先通过地址栏中的网址获取该网页的源代码,并以特征码的方式分析其中是否含有恶意脚本。如果发现含有已知的恶意脚本,则发出警告,抛弃该消息,阻止网页链接;如果是正常脚本,则放行。

特征码分析法虽然简单易行,并且容易对付现有的恶意脚本,但是恶意脚本的形式是变化无穷的,特征库的更新永远滞后于恶意脚本的翻新变化。为此下一个环节采用另一种监控方法,称行为监控法,对注册表进行监控,因为恶意脚本一旦突破浏览器监控模块的防线,接下来的动作绝大多数是修改注册表的某些重要子项[2]。注册表监控模块的作用时间应限定在一定的时间域内,即脚本被放行后的某一小段时间(例如 1 秒钟内),其余时间则没有进行监控的必要。整个防护系统的思路如图 1 所示。

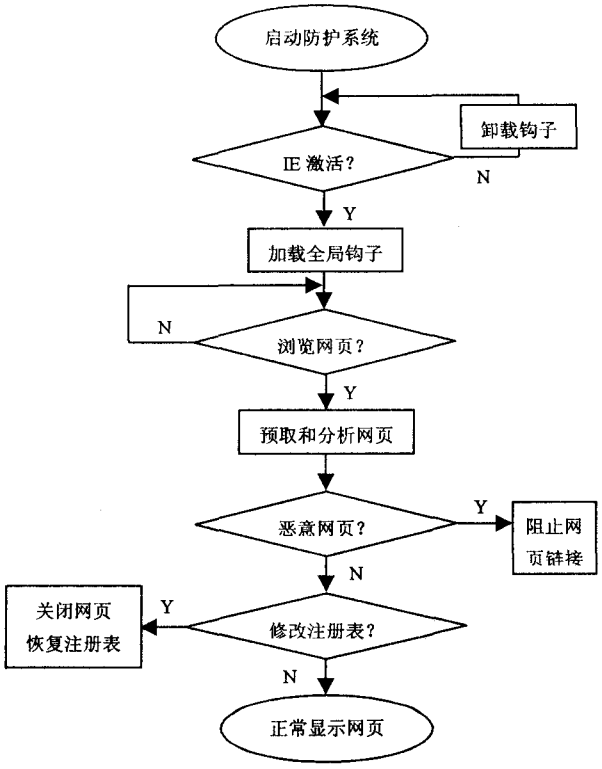

图 1　防护系统工作流程

2　钩子技术

本系统的核心技术是钩子技术,因此有必要首先对钩子技术作一个简要的介绍。

2.1　钩子的概念

在 Windows 操作系统环境下,应用程序的运行是基于消息驱动机制的。所谓消息是描述事件发生的信息。用户点击鼠标和按下按键等都会触发特定的事件,操作系统会产生对应的消息,并把消息放入应用程序的消息队列中,等待检索、分派和处理。每个应用程序都拥有自己的消息队列,所以一般情况下,应用程序只能接收到操作系统发送给自己的消息,而不能过问其他应用程序的消息,但使用钩子可以改变这一切。钩子可以理解成 Windows 消息处理过程中的一个监视点,应用程序可以利用这个监视点对来来往往的消息进行监控和处理,对任意感兴趣的消息都可以进行拦截,经处理后再转发到原目标窗口[3]。

2.2　钩子的分类

根据作用范围,钩子可以分为两类:一类是局部钩子,只能获取本进程内某个线程的消息;另一类是全局钩子,可以获取计算机系统下所有线程的消息,或者是别的进程中指定线程的消息。经笔者测试验证,局部钩子和获取所有线程消息的全局钩子可以放在 EXE 文件或者 DLL 文件中,而获取别的进程中指定线程消息的全局钩子则只能放在 DLL 文件中。根据所监视的消息类型,钩子又可分为键盘钩子、鼠标钩子、系统外壳钩子以及消息过滤钩子等具体 15 种[4]。

2.3　钩子的实现

钩子的实现分为三个步骤:安装钩子、编写钩子函数和卸载钩子。

(1)安装钩子,使用 API 函数 SetWindowsHookEx(),其原型如下:

```
Public Declare Function SetWindowsHookEx Lib "user32" Alias "SetWindowsHookExA" (ByVal idHook As Long, ByVal lpfn As Long, ByVal hmod As Long, ByVal dwThreadId As Long) As Long
```

(2)编写钩子函数,一般格式如下,而其中函数名可以随意命名:

```
Public Function MyHook (ByVal nCode As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
```

在钩子函数的最后一般应调用 CallNextHookEx() 函数来把消息传递给下一个钩子函数,因为 Windows 系统可能同时在维护多个钩子。

(3)卸载钩子,使用 API 函数 UnhookWindowsHookEx(),其原型如下:

```
Public Declare Function UnhookWindowsHookEx Lib "user32" (ByVal hHook As Long) As Long
```

钩子一旦使用完毕,必须及时卸载,因为钩子的安装要占用操作系统的空间和资源,直接影响到操作系统的运行效率[5]。

3 浏览器监控

浏览器监控模块是本防护系统的第一道防线，它的功能主要是利用特征码方式预取和分析网页的源代码。而经过大量采样已知恶意网页所形成的特征库，则作为判断当前网页是否恶意的重要依据[6]。

3.1 扫描挂钩

防护系统每隔 1 秒钟对计算机系统下的当前窗口进行扫描，判断该窗口是否为 IE 浏览器窗口，若是，则加载全局钩子，主要代码如下：

```
Private Sub Timer1_Timer() ' 每 1 秒钟扫描一次，若 IE 窗口被激活则加载钩子
    Dim hCurrent As Long, s As String * 256, i As Long, j As Long
    hCurrent = GetForegroundWindow()
    i = GetWindowTextLength(hCurrent)
    j = GetWindowText(hCurrent, s, i + 1)
    If InStr(1, s, "Microsoft Internet Explorer") <> 0 And hHook = 0 Then ' IE 窗口被激活，加载钩子
        hHook = SetWindowsHookEx(WH_KEYBOARD_LL, AddressOf MyKbHook, App.hInstance, 0)
    ElseIf InStr(1, s, "Microsoft Internet Explorer") = 0 And hHook <> 0 Then ' IE 窗口非激活，卸载钩子
        Call UnhookWindowsHookEx(hHook)
        hHook = 0
    End If
End Sub
```

3.2 钩子函数

本钩子函数的功能是当用户在 IE 地址栏中输入网页地址并按下 Enter 键，则先于 IE 浏览器获取目标网页的源代码，并用特征码方式进行分析。若发现已知恶意脚本则抛弃本按键消息，完全阻断 IE 浏览器浏览该网页。若没有发现恶意脚本，则把消息交还给 IE 浏览器，即放行该网页。在放行之前，先启动注册表监控模块。主要代码如下：

```
Public Function MyKbHook(ByVal nCode As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
    Dim hookStruct As KBDLLHOOKSTRUCT
    Dim hCurrent As Long, Loc As String * 256, strTemp As String
    Dim ThreadId As Long
    Call CopyMemory(hookStruct, ByVal lParam, Len(hookStruct))
    If wParam = 256 And hookStruct.vkCode = 13 Then ' 用户按下 Enter 键，取得地址 hCurrent = FindWindow("IEFrame", vbNullString)
    hCurrent = FindWindowEx(hCurrent, 0&, "WorkerW", vbNullString)
    hCurrent = FindWindowEx(hCurrent, 0&, "ReBarWindow32", vbNullString)
    hCurrent = FindWindowEx(hCurrent, 0&, "ComboBoxEx32", vbNullString)
    hCurrent = FindWindowEx(hCurrent, 0&, "ComboBox", vbNullString)
    hCurrent = FindWindowEx(hCurrent, 0&, "Edit", vbNullString)
    SendMessage hCurrent, WM_GETTEXT, 256, Loc
    Form1.Inet1.Execute Trim(Loc), "GET" ' 使用 Inet 控件获取网页源代码
    Do While Form1.Inet1.StillExecuting
    DoEvents
    Loop
    Open App.Path + "\Virus.dat" For Input As #1 ' 判断是否恶意网页
    Do While Not EOF(1)
    Input #1, strTemp
    If InStr(1, strSource, strTemp, 1) Then
    MsgBox "该网页存在恶意脚本，请勿随意浏览!", vbOKOnly, "警告"
    MyKbHook = 1
    Close #1
    Exit Function
    End If
    Loop
    Close #1
    Call CreateThread(0&, 0&, AddressOf RegMonitor, 0&, 0&, ThreadId) '启动注册表监控模块
    MyKbHook = 0
End Function
Private Sub Inet1_StateChanged(ByVal State As Integer) ' Inet 控件的 StateChanged 事件
    Dim strTemp As String
    Dim strLast As String
    Select Case State
    Case icError ' 错误产生
    Exit Sub
    Case icResponseCompleted ' 正常传送结束
    strTemp = Inet1.GetChunk(1024)
    Do While True
    strLast = strLast & strTemp
    strTemp = Inet1.GetChunk(1024)
    DoEvents
    If Len(strTemp) = 0 Then
    Exit Do
    End If
    Loop
    End Select
    strSource = strLast
End Sub
```

4　注册表监控

如果恶意脚本采用了变种或者加密手段,或者是最新出现的恶意脚本,它们将躲过第一道防线,顺利得到 WSH 的执行[7]。而接下来它们绝大多数会去修改注册表,这时注册表监控模块将发挥作用,该模块主要是监视注册表中的某些重要子项是否被改动,若在网页被放行后的一小段时间内(例如 1 秒钟内)被改动,判断为恶意脚本所为,则将注册表恢复为原先状态,并发出警告和关闭网页。该模块做成一个独立的线程,在钩子函数的最后被创建[8,9]。主要代码如下:

```
Public Sub RegMonitor() '注册表监控函数的定义
    Dim hKey As Long, Ret As Long, hEvent As Long, lFilter As Long
    Dim i As Long, j As Long
    Ret = RegOpenKeyEx (HKEY_LOCAL_MACHINE, "SOFTWARE", 0&, KEY_NOTIFY, hKey) ' 打开 SOFTWARE 项
    hEvent = CreateEvent(0, True, False, vbNullString)
    lFilter = REG_NOTIFY_CHANGE_NAME Or REG_NOTIFY_CHANGE_ATTRIBUTES Or REG_NOTIFY_CHANGE_LAST_SET Or REG_NOTIFY_CHANGE_SECURITY
    i = RegNotifyChangeKeyValue(hKey, True, lFilter, hEvent, True)
    j = WaitForSingleObject(hEvent, 1000) ' 等待 1 秒钟,即监控 1 秒钟
    If j = WAIT_OBJECT_0 Then
    MsgBox "注册表被改动,监控程序将进行恢复!", vbOKOnly, "警告"
    i = RegRestoreKey(hKey, App.Path & "\RegBack.reg", &0)
    End If
    RegCloseKey (hKey)
    CloseHandle (hEvent)
End Sub
```

5　结束语

阐述了一种恶意网页防护系统的设计与实现,设计上采用了按时间顺序依次监控浏览器和注册表的较为周密的方案,实现上主要采用了钩子技术。经实践表明,该系统具有较好的实时防护效果。

参考文献:

[1] 鲍欣龙,罗文坚.可用于恶意脚本识别的注册表异常行为检测技术[J].计算机工程,2005,31(8):137-139.

[2] 唐晓东,何连跃.一种恶意代码防护方法及其实现[J].计算机工程,2005,31(12):143-145.

[3] 梁　庚,李　文.应用系统钩子和内存映像实现一类进程间的通信[J].计算机工程与科学,2005,27(11):96-97.

[4] 申晓龙,许文雨.Windows 钩子技术的研究与应用[J].成都信息工程学院,2005,20(4):380-382.

[5] 余姜德,于志平.Windows 钩子技术在病毒程序中的应用[J].现代计算机,2005(2):83-86.

[6] 李永革,张维明.基于浏览器的安全问题研究[J].计算机工程,2002,28(1):55-56.

[7] Grimes R A.恶意传播代码:Windows 病毒防护[M].北京:机械工业出版社,2004.

[8] Beveridge J,Wiener R.Win32 多线程程序设计[M].侯　捷译.武汉:华中科技大学出版社,2002.

[9] Richter J.Programming Application for Windows[M].US:Microsoft Press,1999.

(上接第 8 页)

[2] Adar E,Huberman B. Free riding on Gnutella[R].[s.l.]. Xerox PARC,2000.

[3] Feldmanym, Laizk. Quantifying disincentives in peer-to-peer networks[C]//workshop on economics of peer-to-peer systems. CA: Springer-verlag,2003:117-122.

[4] Yang B, Molina H G. Micropayments for peer-to-peer systems[C]//Proceeding of the 10th ACM Conference on computer and Communications Security. Washington: ACM press,2003:300-310.

[5] Ioannidis J, Ioannidis S, Keromytis A D, et al. Fileteller: paying and getting paid for file storage[C]//In Proceedings of the Sixth Annual Conference on Financial Cryp tography. Bermuda: Springer-Verlag,2002:282-299.

[6] Adya A, Bolosky W J,Castro M,et al. Federated, available, and reliable storage for an incompletely trusted environment [C]//In Proceedings of the 5th Symposium on Operating Systems Design and Implementation. Boston: ACM Press,2002: 12-14.

[7] Cox L P, Murray C D, Noble B D. Pastiche: making backup cheap and easy[C]//In Proceedings of the 5th Symposium on Operating Systems Design and Implementation. Boston: ACM Press,2002:285-298.

[8] 宿建宗,李秉智.P2P 文件共享框架中激励机制的研究[J].重庆邮电学院学报,2006,18(1):123-125.

[9] 陈志琦,苏德富.基于博弈论框架的 P2P 激励模型[J].计算机工程,2005,31(16):118-121.

[10] 曾宇光,陈志刚. JXTA 路由改进算法的研究与实现[J].微计算机信息,2007,33:223-225.