基于 WSH 的 Windows 系统监视程序设计

田 原
(荆门职业技术学院计算机系 湖北荆门 448000)

摘 要 WSH 是内嵌于 Windows 操作系统中的脚本语言工作环境,实现多类脚本文件在 Windows 界面或 Dos 命令提示符下的直接运行。FSO 组件由 Microsoft 提供,提供对计算机文件系统的访问。本文探讨采用 WSH 和 FSO 组件设计 Windows 系统监视程序。

关键词 Windows 脚本宿主,文件系统对象,Windows

The Program for Supervising Windows Based on WSH

TIAN Yuan
(Jingmen Vocational Technical College, Jingmen, Hubei448000)

Abstract Windows Script Host(WSH) is a powerful, flexible tool for automating recurring PC tasks. WSH lets you create and execute scripts to tailor your Windows system for your own needs-simply and with complete programmatic control. The FSO object model, which is contained in the Scripting type library(Scrrun. Dll), gives your applications the ability to access the file system. This paper discuses. the program for supervising Windows based on WSH and FSO.

Keywords WSH, FSO, Windows

在 DOS 环境下,借助于 DOS 命令和批处理语言可方便地辅助用户实现各类和系统有关的操作。随着 Windows 系统的推出,批处理语言已无法适应 32 位操作系统的编程要求。本文通过使用 WSH(Windows Scripting Host, Windows 脚本宿主)技术来调用 FSO(File System Object, 文件系统对象)组件和 WSH 自带的网络对象,方便地实现了对系统的监视。

1 WSH

1.1 WSH 简介

WSH 是内嵌于 Windows 操作系统中的脚本语言工作环境,实现多类脚本文件在 Windows 界面或 Dos 命令提示符下的直接运行。WSH 架构于 ActiveX 之上,通过充当 ActiveX 的脚本引擎控制器,WSH 为 Windows 用户充分利用威力强大的脚本指令语言扫清了障碍。

编写一个脚本文件(如后缀为. vbs 或. js),然后在 Windows 下双击并执行它时,系统就会自动调用 WSH 程序对它进行解释并执行。WSH 程序执行文件名为 Wscript. exe(若是在命令行下,则为 Cscript. exe)。

WSH 诞生后,微软在 Internet Information Server 4. 0、Windows Me、Windows 2000 Server,以及 Windows 2000 Professional、Windows XP 等产品中都嵌入了 WSH。

1.2 WSH 的功能

WSH 的设计,在很大程度上考虑到了"非交互性脚本(noninteractive scripting)"的需要。在这一指导思想下产生的 WSH,给脚本带来非常强大的功能,例如,可以利用它完成映射网络驱动器、检索及修改环境变量、处理注册表项等工作;管理员还可以使用 WSH 的支持功能来创建简单的登录脚本,甚至可以编写脚本来管理活动目录。这些功能的实现,均与 WSH 内置的多个对象密切相关,这些内置对象肩负着直接处理脚本指令的重任。图 1 是 WSH 的内置对象构成情况。

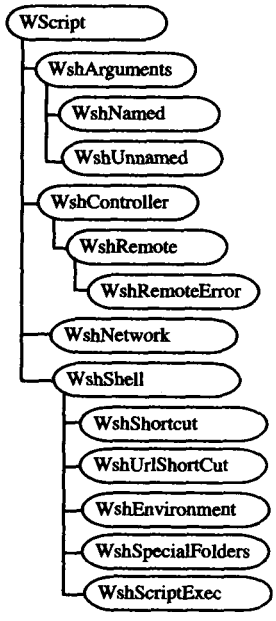

图 1 WSH 内置对象

从图 1 中可以看出,WSH 共有 14 个内置对象,它们各自有着明确分工。位于最底部的是 Wscript 对象,主要作用是提取命令行变量,确定脚本文件名,确定 WSH 执行文件名(Wscript. exe 还是 Cscript. exe),确认 host 版本信息,创建、

田 原 副教授,主要研究方向是软件工程、信息系统。

关连及分离 COM 对象，写入事件，按程序结束一个脚本文件的运行，向默认的输出设备（如对话框、命令行）输出信息等；WshArguments 用于获取全部的命令行变量；WshNamed 负责获取指定的命令行参数集；WshUnnamed 负责获取未经指定的命令行参数集；WshNetwork 用于开放或关闭网络共享，连接或断开网络打印机，映射或取消网络中的共享，获取当前登录用户的信息；WshController 可以创建一个远程脚本对象；WshRemote 可以实现网络中对计算机系统的远程管理，也可按计划对其它程序/脚本进行处理；WshRemote Error 的作用：当一个远程脚本（WshRemote 对象）因脚本错误而终止时，获取可用的错误信息；WshShell 主要负责程序的本地运行，处理注册表项、创建快捷方式、获取系统文件夹信息，处理环境变量；WshShortcut 用于按计划创建快捷方式；WshSpecialfolders 用于获取任意一个 Windows 特殊文件夹的信息；WshURLShortcut 用于按程序要求创建进入互联网资源的快捷方式；WshEnvironment 用于获取任意的环境变量（如 WINDIR，PATH，或 PROMPT）；WshScriptExec 用于确定一个脚本文件的运行状态及错误信息。

通过这些内置对象，可以利用 WSH 充分发挥 VBScript 及 JScript 等脚本的强大威力，提高工作效率。

1.3 WSH 工作流程

脚本经常会被嵌入网页，其中包括 HTML 页面（客户机端）和 ASP 页面（服务器端）。对于嵌入 HTML 页面的脚本，其所需的解析引擎会由 IE 这样的网页浏览器载入；对于嵌入 ASP 页面的脚本，其所需的解析引擎会由 IIS（Internet Information Services）提供。对于出现在 HTML 和 ASP 页面之外的脚本（它们常以独立的文件形式存在），就需要经由 WSH 来处理了。

WSH 正常工作的前提：必须安装了 IE 3.0 或更高版本，因为 WSH 在工作时会调用 IE 中的 VBScript 和 JScript 解析引擎。

WSH 的工作流程，实际上就是脚本文件被解析并执行的过程。脚本文件经由 WSH 执行的流程图如图 2 所示。从图中能对 WSH 在脚本文件运行中所起到的作用有个理性认识。对于这个流程图，还需要补充两点：1）图中第 2、3 步，WSH 根据脚本文件后缀名，到系统注册表中查询所需的脚本引擎时，VBScript 和 JScript 两种语言的解析引擎是 Windows 系统中原有的，而其它脚本语言的解析引擎，如 PERL、TCL 等，需要用户另行定义；2）第 5 步执行脚本命令时，一些脚本指令会使用到 WSH 内置对象所提供的服务（参见 1.2 节），例如处理注册表项。这时，脚本指令就会向 WSH 提出请求，并由 WSH 完成所需任务。

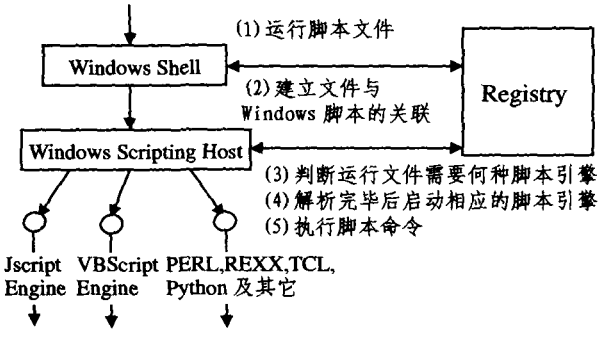

图 2 流程图

1.4 WSH 的应用

WSH 实际上是一个脚本语言的运行环境，它之所以具备强大的功能，是因为充分挖掘了脚本语言的潜力。现在给 2 个脚本文件利用 WSH 执行任务的实例。

脚本文件的编写十分方便，可以选用任意一个文字编辑软件进行编写，写完后，保存为 WSH 所支持的文件名（如. js、. vbs 文件）。例如，打开记事本，在上面写下：

WScript. Echo("走近 WSH")

将它保存为以. vbs 或. js 为后缀名的文件并退出记事本。双击执行这个文件。

再如，利用 WSH 完成一次创建十个文件夹的工作。代码如下：

```
dim objdir
set objdir=wscript. createobject("scripting. filesystemobject")
for k=1 to 10
anewfolder="c:\chapter" & k
objdir. createfolder(anewfolder)
next
```

同样，将它存为. vbs 文件并退出。运行后，可发现 C 盘根目录下增加了十个新文件夹。

2 FSO

2.1 FSO 简介

FSO 组件由 Microsoft 提供，位于脚本运行库 scrrun. dll 中，许多应用系统都会调用它，例如 Access，Word 等。FSO 对象包括：驱动器对象（Drive Object），用来存取本地盘或网络盘；文件系统对象（Fiel System Object），用来存取文件系统；文件夹对象（Folder Object），用于存取文件夹的各种属性；文本流对象（TextStream Object 简称 TS）存取文件内容。

2.2 创建 FSO 对象

在 VBscript 中创建 FSO 对象的方法如下：

```
Dim fso
Set fso=CreateObject("Scripting. FileSystemObject")
```

2.3 FSO 对象的属性

FSO 对象只有一个属性 Drives，它返回一个驱动器集合，包含了本地机器上所有可用的 Drive 对象。可移动媒体的驱动器不需要插入媒体就可以出现在 Drives 集合中。

2.4 FSO 对象的方法

SO 对象的方法非常丰富，下面仅给出各种方法的语法功能。

BuildPath：在已存在路径后追加名称；CopyFile：从一个位置向另一个位置复制一个或多个文件；CopyFolder：将文件夹连同子文件夹从一个位置复制到另一个位置；CreateFolder：创建文件夹；CreateTextFile：建指定的文件名并返回一个 TextStream 对象，可使用这个对象对文件进行读写；DeleteFile：删除一个或多个指定文件；DeleteFolder：删除一个或多个指定的文件夹及其内容；DriveExists：判断驱动器是否存在；FileExists：判断文件是否存在；FolderExists：判断文件夹是否存在；GetAbsolutePathName：根据提供的路径说明返回明确完整的路径；GetBaseName：返回字符串，该字符串包含路径中最后成分中的基本名称，不包含文件扩展名；GetDrive：返回相应于指定路径中驱动器的 Drive 对象；GetDriveName：根据指定路径返回包含驱动器名称的字符串；GetExtensionName：返回包含路径中最后成分扩展名的字符串；GetFile：根据指定的路径中的文件返回相应的 File 对象；GetFileName：返回指定路径的最后成分，但指定的路径不能只是

驱动器说明；GetFolder：根据指定路径中的文件夹返回相应的 Folder 对象；GetParentFolderName：根据指定路径中的最后成分返回包含其父文件夹名称的字符串；GetSpecialFolder 方法返回指定的特殊文件夹对象，有 window、system、temp 三种；GetTempName：返回一个随机产生的临时文件或文件夹名，有助于执行那些需要临时文件或文件夹的操作；MoveFile：从一个位置向另一个位置移动一个或多个文件；MoveFolder：从一个位置向另一个位置移动一个或多个文件夹；OpenTextFile：打开指定的文件并返回一个 TextStream 对象，可以通过这个对象对文件进行读、写或追加。

3 系统功能及实现

Windows 系统监视程序主要用来获取网络信息、驱动器信息、文件夹信息、系统关键文件信息等。

为了获取网络信息，使用 WSH 自带的网络对象。具体编程代码如下：

```
Sub dispnetstat '显示网络状态的子程序
    '创建网络对象
    Set WshNetwork = Wscript. CreateObject ("Wscript. Network)
    '获取系统所有的网络驱动器
    Set oDrives=WshNetwork. EnumNetworkDrives
    '获取系统所有的网络打印机
    Set oPrinters=WshNetwork. EnumPrinterConnections
    '获取系统域名
    netinfo="域名:"+WshNetwork. UserDomain+vbcrlf
    '获取计算机名
    netinfo=netinfo+"计算机名:"+WshNetwork. ComputerName+vbcrlf
    '通过循环获取网络驱动器的名称
    netmap=""
    for i=0 to oDrives. Count()-1 step 2
        netmap=netmap+"驱动器"+oDrives. Item(i)+" ="
    +oDrives. Item(i+1)+vbcrlf
    next
    netprn=""
    '通过循环获取网络打印机的名称
    for i=0 to oPrinters. Count()-1 step2
        netprn=netprn+"端口:"+oPrinters. Item(i)+"="+
    oPrinters. Item(i+1)+vbcrlf
    next
    Set WshNetwork=nothing '释放网络对象变量
End Sub
```

为了获取驱动器信息、文件夹信息和系统关键文件信息等，必须通过使用 WSH 来调用 FSO 组件实现。其中，获取文件夹信息的具体代码如下，获取驱动器信息、系统关键文件信息等的方式与其类似，在此不具体给出。

```
Function folderatt(fo)'获取文件夹属性的函数
Dim att
On Error resume next '获取目录的名称
folderatt=folderatt+fo. name+"目录:"+vbcrlf
'获取目录的文件数
folderatt=folderatt+"该目录下的文件数:"+cStr(fo. Files. Count)
+vbcrlf
'获取目录的子目录数
folderatt=folderatt+"该目录下的子目录数:"+cStr(fo. SubFolders. Count)+vbcrlf
'获取目录的占用字节数
folderatt=folderatt+"该目录占用字节数:"+cStr(fo. Size/1024/1024)+"MB"+vbcrlf
'获取目录的最后修改时间
folderatt=folderatt+"该目录的最后修改时间:"+cStr(fo. DateLastModified)+vbcrlf
End Function
```

结束语 通过 WSH 可以充分利用脚本来实现计算机工作的自动化，但不可否认，也正是它的这一特点，使系统又有了新的安全隐患。许多计算机病毒制造者正在热衷于用脚本语言来编制病毒，并利用 WSH 的支持功能，让这些隐藏着病毒的脚本在网络中广为传播。因此，对于来历不明、尤其是邮件附件里的一些脚本文件还是应该保持戒备。

参 考 文 献

1 张志民，王强，武港山. 基于 WSH 的脚本开发. 计算机应用研究，2000，(7)

2 哈节棍. 访问注册表之 WSH 篇. 中文信息：程序春秋，2002(9)

3 范青山. WSH，批处理技术的新武器. 软件，2002(6)

4 邵方，刘宗田. 脚本语言发展研究. 计算机科学，2000，27(1)

5 孙康生. 应用 FSO 对象模型实现文件查找的探讨. 开封大学学报，2004(4)

6 王军号，等. 基于 Web 的文件搜索引擎的设计与实现. 科技情报开发与经济，2005(4)

(上接第 284 页)

结束语 在 Unix 系统中，通过对进程正常运行时的执行轨迹进行分析来刻画进程的正常运行状态，是一种重要的异常检测技术。在利用进程运行时产生的系统调用短序列建立入侵检测模型方面，有枚举法、基于频率的方法、基于数据挖掘的方法、基于有限状态机的方法和结合调用参数的系统调用短序列异常检测等方法。本文提出了一种基于粗糙集约简的系统调用序列异常检测方法，它根据进程系统调用的前 $k-1$ 个位置，利用粗糙集约简来对第 k 个位置进行预测。Rough 集理论的约简方法确保了能得到一组预测第 k 个系统调用位置的最小规则集，从而可以利用这组规则来对实际进程进行异常检测。基于合成的 UNM sendmail 系统调用数据的实验结果表明，本文所提出的异常检测算法性能好于 Forrest 等人的 tide 方法，与 Wenke Lee 等人的数据挖掘算法检测精度相当。但在选择较大的阈值时，漏报率更低。

参 考 文 献

1 Forrest S, Hofmeyr S A, Somayaji A, et al. A sense of self for Unix process [J]. In: Proceedings of 1996 IEEE Symposium on Computer Security and Privacy, 1996. 120～128

2 Lee Wenke, Stolfo S, Chan Phil. Learning Patterns from Unix Process Execution Traces for Intrusion Detection [A]. AAAI Workshop: AI Approaches to Fraud Detection and Risk Management, July 1997

3 Warrender C, Forrest S, Pearlmutt B. Detecting Intrusion Detection Using System Calls: Alternative Data Model [J]. In: Proceedings of 1999 IEEE Symposium on Computer Security and Privacy, 1999. 133～145

4 Wespi A, Dacier M, Debar H. Intrusion Detection using variable-length audit trail patterns [J]. RAID, 2000. 110～129

5 Tandon G, Chan P. Learning Useful System Call Attributes for Anomaly Detection [A]. In: Proc 18th Intl FLAIRS Conf, 2005. 405～410

6 Liao Y H, Vemuri V R. Use of K-Nearest Neighbor classifier for intrusion detection [J]. Computers & Security, 2002, 21(5): 439～448

7 Lee Wenke, Stolfo S J, Mok K W. A Data Mining Framework for Building Intrusion Detection Models [A]. 1999 IEEE Symposium on Security and Privacy, Oakland, California, May. 1999

8 王国胤. Rough 集理论与知识获取[M]. 西安：西安交通大学出版社，2001

9 蔡忠闽，管晓宏，邵萍，等. 基于粗糙集理论的入侵检测新方法. 计算机学报[J]. 2003，26(3)：361～366

10 Mahoney M, Chan P. Learning Rules for Anomaly Detection of Hostile Network Traffic [A]. In: Proc. Third IEEE Intl Conf on Data Mining(ICDM), 2003. 601～604

11 UNM Sequence-based Intrusion Detection data set [EB/OL]. http://www.cs.unm.edu/～immsec/data/. Cited 2005

12 Rosetta [EB/OL]. Knowledge Systems Group, Dept of Computer and Info Science, Norwegian University of Science and Technology, Trondheim, Norway and Group of Logic, Inst of Mathematics, University of Warsaw, Poland. http://rosetta.lcb.uu.se/general/