第16卷 第8期
2006年8月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.16 No.8
Aug. 2006

# 一种安全接入系统的设计与实现

张怡婷[1],杨 明[2]
(1.南京邮电大学 计算机科学与技术系,江苏 南京 210003;
2.东南大学 计算机科学与工程系,江苏 南京 210096)

**摘 要**:访问控制和安全审计始终是网络安全研究领域的热点问题之一,如何对物理上分布的多个服务器进行集中的安全访问控制和审计更是一项非常具有现实意义和实际应用价值的系统研发课题。文中提出了一种安全的接入系统的设计与实现,该系统采用Win32操作系统钩子与Sniffer嗅探技术实现了细粒度的访问控制与安全审计功能,同时结合Windows操作系统自有的域管理和域控制功能,可以很好地满足此需求。实际应用结果表明该系统具有很强的实用性和安全性。

**关键词**:访问控制;安全审计;钩子;嗅包器
**中图分类号**:TP393.08 **文献标识码**:A **文章编号**:1673-629X(2006)08-0103-03

# Design and Implementation of a Secure Access System

ZHANG Yi-ting[1], YANG Ming[2]
(1. Department of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)
2. Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

**Abstract**: Access control and audit are always one of the hot research problems in network security fields. Furthermore study and design on systems which can perform centralized access control and audit for distributing multi-servers is a practically valuable task. Design and implementation of a secure access system was proposed in this paper. Integrating Windows system's functions of domain management and control, it implemented fine-grained access control and audit adopting techniques of hook and sniffer, which can satisfy the needs well. Application results show that the system is practically useful and secure.

**Key words**: access control; secure audit; hook; sniffer

大型企业网往往有着这样的特点:分布在不同分支机构的服务器必须向同样分布在不同网段、甚至是移动的公司员工计算机终端提供服务,如何进行集中访问控制以及操作审计是难点问题之一。文中给出了一种分布环境下安全接入系统的设计与实现。

## 1 系统设计

### 1.1 访问控制

整个系统提供两个层面上的访问控制,分别是基于操作系统的和基于软件系统钩子(Hook)控制的。

首先是客户登录到接入的Windows终端服务上,Windows系统本身可以根据用户帐号以及域策略[1]设置对允许用户访问哪些服务器以及使用哪些软件进行控制。这个层面上的访问控制粒度较粗,依赖于Windows操作系统以及Windows域的设置。

其次,对于同一个软件,不同角色的用户可能都需要访问,但是角色不同其访问权限也不相同。以Word来说,可能允许某一类用户使用打开文件菜单操作,但不允许其使用保存/另存为菜单项。这种更细粒度的访问控制往往不被应用软件所支持,在系统中可以通过应用钩子函数过滤特定软件消息的方式来实现。

### 1.2 操作审计

用户登录Windows终端服务后,需要对其所作的操作(尤其是连接服务器所作的操作)进行日志。一方面,需要记录其操作行为;另一方面,需要记录这些操作所带来的结果。

对于用户操作,系统使用钩子函数捕获用户所有的键盘以及特定的鼠标操作信息;对于操作结果,系统使用Sniffer技术分析用户操作所引起的与服务器之间的网络交互报文。此外,Sniffer技术也用于一些无法直接捕获用户操作的应用。以FTP为例,用户如果直接以命令行方式访问服务器,那么键盘钩子可以捕获用户输入的所有命令;但是如果用户使用类似于CuteFTP之类的GUI工具,那么应用钩子程序捕获用户操作是非常困难的,可以通过Sniffer来分析FTP协议来记录用户进行了哪些操作。

收稿日期:2005-11-28
基金项目:教育部回国人员资助项目;南京市回国人员择优资助项目
作者简介:张怡婷(1978-),女,安徽合肥人,硕士,助教,研究领域为高性能网络、网络应用。

### 1.3 系统流程

图 1 所示为安全接入系统流程图。

首先,用户通过远程桌面登录接入服务器上安装的 Windows 终端服务,Windows 系统本身的帐号管理、权限分配以及 Windows 域的权限策略将对用户登录后所能进行的操作以及所能访问的外部服务器(包括 FTP、数据库服务器、Unix 服务器等)进行管理和限制。

其次,用户在远程桌面中进行操作,部分特定的操作将被安装在接入服务器系统中的钩子程序捕获并记录至日志数据库。另一方面,这些钩子程序将按照更细粒度的权限设置将特定服务器的客户端软件系统中部分菜单功能禁用或者开启。

最后,针对菜单鼠标操作无法很好地进行记录的问题,系统通过捕捉并分析用户操作引起的客户端与服务器之间的报文交互来记录用户的操作和操作的结果。

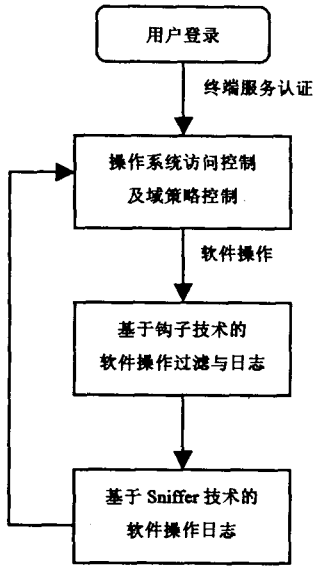


图 1 系统流程图

## 2 系统实现

### 2.1 系统结构

如图 2 所示,整个系统可以划分为信息收集模块、钩子模块、嗅包模块、日志模块以及查询和控制模块 5 部分。其中信息收集模块通过监听本地 UDP 端口,从注入至各个软件的钩子程序接收用户的各种键盘操作信息;钩子模块以动态链接库形式出现,并被自动注入用户所操作的各个客户端软件中,监视、截获或者更改用户操作消息(事件);嗅包模块底层采用 WinPcap 开发包进行网络报文的捕获,并针对特定的应用分析并记录其交互过程;日志模块从缓存读取数据并记录至日志数据库;查询和控制模块包含 3 部分功能:首先是提供日志数据库的查询报表;其次是控制(启动、停止)嗅包模块、信息收集模块以及日志

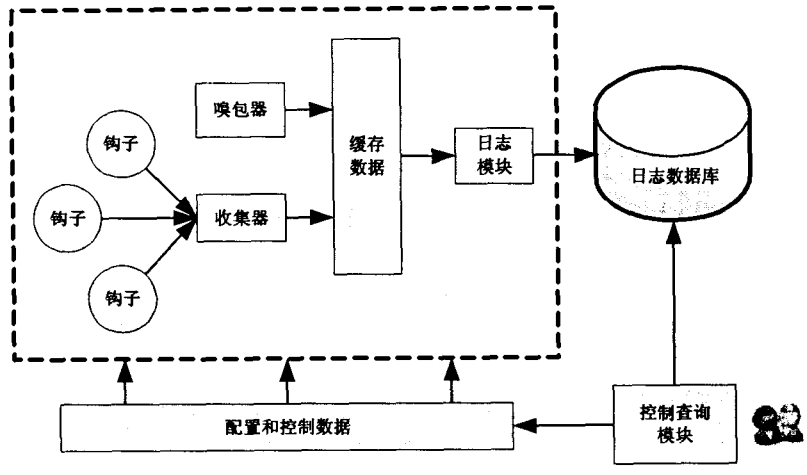


图 2 系统结构图

模块,这 3 个模块均以系统服务形式存在;最后是提供系统配置接口。

### 2.2 重要模块

#### 2.2.1 钩子模块

钩子模块主要采用 Win32 的 HOOK 技术[2]来实现。在 Windows 操作系统中,钩子是在系统事件被传递给应用之前进行截取(intercept)的一种框架机制。这些系统事件包括消息、鼠标操作、击键消息等,钩子程序可以对其进行监听、修改甚至丢弃。用户接收这些事件的函数被称为过滤函数,过滤函数被加载到系统钩子中的过程被称为钩子的安装。系统钩子包含多种类型,如 WH_CALLWNDPROC, WH_CALLWNDPROCRET, WH_CBT, WH_DEBUG, WH_FOREGROUNDIDLE, WH_GETMESSAGE, WH_JOURNALPLAYBACK, WH_JOURNALRECORD, WH_KEYBOARD, WH_MOUSE, WH_MSGFILTER, WH_SHELL, WH_SYSMSGFILTER。同一系统钩子中有多个过滤函数加载时,操作系统会自动维护一个链表,最新加载的过滤函数位于链表的头部。

本系统主要实现了 WH_KEYBOARD 和 WH_CALLWNDPROCRET 两种钩子过滤函数,分别用于键盘操作的记录和应用软件菜单功能的屏蔽。

* WH_KEYBOARD 钩子函数。

监听特定软件(控制台,SecureCRT 等)的用户击键信息。通过在内存中针对每个进程维持一操作数据缓存,系统将记录用户键入的击键信息直至用户键入回车键提交命令。随后,系统将完整的用户命令作为 UDP 包数据发送给收集器模块。

* WH_CALLWNDPROCRET 钩子函数。

软件菜单中的各个菜单项都有自己的消息 ID 及参数,通过拦截或者改变特定 ID 消息,本系统实现了更细粒度的用户操作访问控制。以 SecureCRT Version 4.1.4 (build 238)软件为例,应用微软的 Spy++工具,可以发现菜单项【Paste】对应的消息为:Message = WM_COM-

MAND wParam = 0000E125 lParam = 00000000;而【Paste As Quotation】对应的消息为:Message = WM_COMMAND wParam = 00008607 lParam = 00000000。下面的代码截获了【Paste As Quotation】消息并将其转向为【Paste 操作】。如果要实现禁用菜单的功能,简单更改下面的代码即可(如图 3 所示)。

```
if(Target = = Scrt){// Paste & Paste As
  if((pmsg->message = = WM-COMMAND) &&
    (LOWORD(pmsg->wParam) = = 0x8067 || LOWORD(pmsg->
wParam) = = 0xE125)){
      if(LOWORD(pmsg->wParam) = = 0x8067)//以 Paste 替换 Paste As
          pmsg->wParam = 0xE125;
      }
}
```

图 3　HOOK 代码片断

2.2.2　收集器模块

收集器模块以系统服务形式出现,监听本地的 UDP 端口。由于钩子模块是由系统注入到各个登录用户的各个用户进程中去的,因此必须有收集器这么一个集中的模块来收集并记录用户操作信息。收集器模块将钩子模块发来的用户操作数据记录至临时文件中,并每隔一固定时间段将这些临时文件转移至缓存数据文件夹。

2.2.3　嗅包器模块

嗅包器模块以系统服务形式出现,主要包含 3 部分(如图 4 所示):首先是底层采用 WinPcap[3~5] 来进行抓包;其次实现了一个 Analyzer 模块来进行协议分析并记录至临时文件;最后服务体部分每隔固定时间将临时文件移至缓存数据文件夹。

其中报文与应用进程之间的对应关系可以通过系统 iphlpapi.dll 动态链接库中的隐藏接口函数 AllocateAndGetTcpExTableFromStack 和 AllocateAndGetUdpEx TableFromStack 来查找;而进程与登陆帐号以及登陆会话之间的对应关系的查找可以通过微软终端服务软件开发包中的 WTSEnumerateProcesses 等函数来进行。

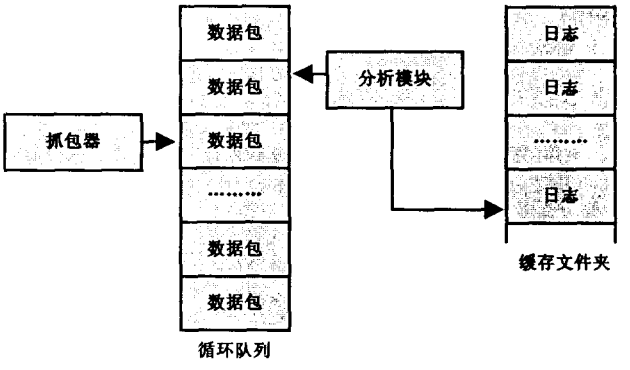


图 4　嗅包器模块

2.2.4　日志模块

日志模块以系统服务形式出现,每隔固定时间即从缓存文件夹中读取数据并记录至外置的日志数据库中。

**2.3　系统部署**

系统的部署环境如图 5 所示:客户机分布于不同局域网内,通过登录接入服务器域中的终端服务器来访问同样分布在不同局域网内的 Oracle 数据库服务器、FTP 服务器,以及 Unix 服务器。终端服务器装有供登录用户使用的 SQL Plus,CuteFTP,SecureCRT 等客户端软件,同时后台以服务形式运行着访问控制与审计系统。用户在终端服务器上的所有操作都被系统记录至日志数据库并进行审计。

## 3　总　结

文中给出了一种安全接入软件系统的设计与实现,该系统可以很好地提供集中访问控制与安全审计功能。实

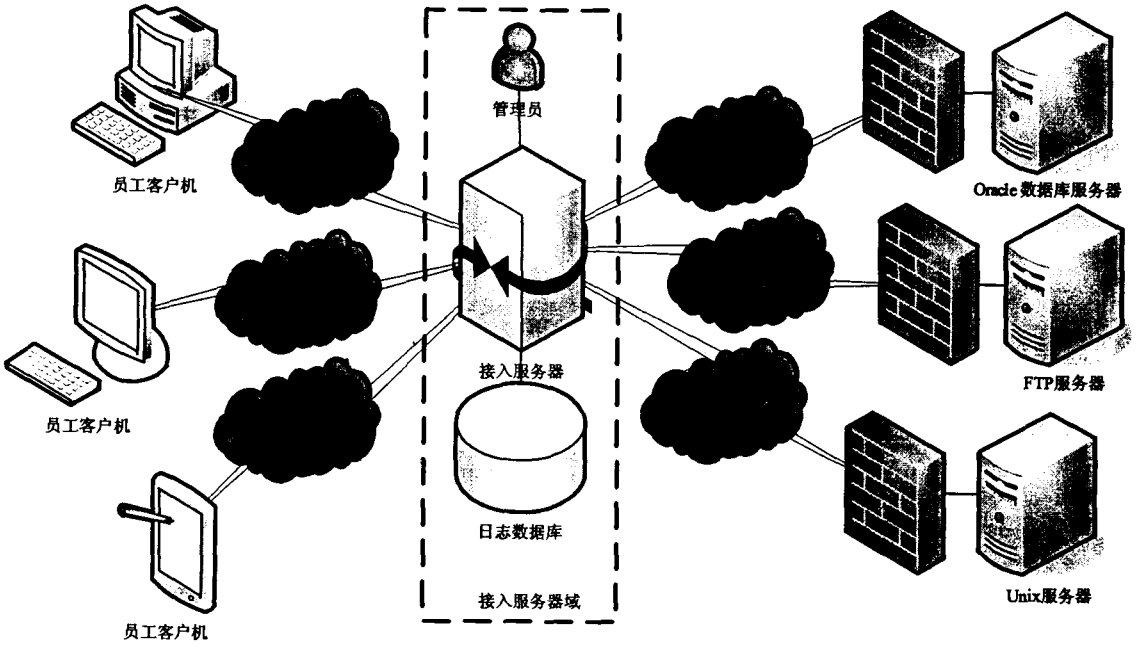


图 5　系统部署图

(下转第 109 页)

表3~5列出了C4.5,newc45在75个实验数据上分类精确度、叶子节点数目和树的尺寸大小的对比。

在这75个实验数据集合上,J48[10]的平均分类精确度为81.787037%;newc45的平均分类精确度为81.569843%;newc45与J48的平均分类精确度差为-0.217194%,非常小,可以看出newc45在绝大部分实验数据集上取得了与C4.5一样好的分类性能。在22个数据集合上,newc45的分类精确度比J48分类器的精确度高。在27个数据集合上,newc45的分类精确度比J48分类器的精确度低。在26个数据集合上,newc45的分类精确度与J48分类器的精确度一样高。

newc45在75个实验数据集上生成的决策树的平均节点数目为149,平均叶子节点数目为82;在J48上生成的决策树的平均节点数目为149.800000,平均叶子节点数目为82.293333。虽然它们的平均节点数目和平均叶子节点相差不大,但newc45在20个实验数据集上生成的决策树比C4.5生成的要小,在46个实验数据集上生成的决策树比C4.5生成的要大,而C4.5仅仅在9个实验数据集上生成的决策树比newc45生成的小。综上所述:从统计意义上讲,newc45在75个实验数据集上生成的决策树比C4.5生成的小。对于数据集合aetrain,aetest,vehicle,soybean-large,sign,satellite,mfeat-mor,german,audio,newc45的分类精确度比J48的高,而且newc45生成的决策树比C4.5生成的小。对于newc45的分类精确度比J48低的27个数据集合中,10个数据集合cleveland,glass7,hepatitis,ionosphere,letter-recog,pendigits,sbl,segment,soybean,splice-c4.5,newc45生成的决策树比C4.5生成的小。

## 7 结　论

综合信息增益和增益比率、Gini索引、基于Goodman-Kruskal关联索引这三种选择分裂属性的标准的特点,通过竞争机制,用投票的方式多数胜少数的方法选择最佳分裂属性,保留了经典决策树分类器C4.5一样好的分类精确度。实验结果表明它在大部分实验数据集上,可以生成更小的决策树。

关于是否还有其他更好的分裂标准来选择最佳的分裂属性,还有待下一步研究。

**参考文献:**


[1] Han Jiawei, Kamber M. Data mining concepts and techniques [M]. San Francisco: Morgan Kaufmann Publishers, 2001. 185-219.

[2] Mitchell T M. Machine learning[M]. New York, America: McGraw-Hill Companies, Inc, 1997.112-140.

[3] Simovici D A, Szymon J. A metric approach to building decision trees based on Goodman-Kruskal association index[A]. The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining[C]. Sydney, Australia:[s.n.],2004.181-190.

[4] Witten I H, Frank E. Data mining: practical machine learning tools and techniques with java implementations[M]. Seattle: Morgan Kaufmann, 2000.

[5] Quinlan R. C4.5: Pragrams for machine learning[M]. California: Morgan Kaufmann Publishers, Inc, 1993.

[6] Quinlan J R. Induction of decision trees[J]. Machine Learning, 1986, 1 (1):81-106.

[7] Kohavi R. Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree Hybrid[A]. In: Simoudis E, Han J, Fayyad U M, eds. Proc of the 2nd Int'l Conf on Knowledge Discovery and Data mining[C]. Menlo Park: AAAI Press, 1996. 202-207.

[8] Blake C L, Merz C J. UCI Repository of machine learning databases[Z]. Irvine, CA: Department of Information and Computer Science, University of California, 1998.

[9] 黄厚宽,石洪波,王志海,等.一种限定性的双层贝叶斯分类模型[J].软件学报,2004,15(2):193-199.

[10] Friedman J H, Kohavi R, Yun Y. Lazy decision trees[A]. Thirteenth National Conference on Artificial Intelligence[C]. Menlo Park: AAAI Press,1996.717-724.


(上接第105页)

际应用表明,该系统具有很强的实用性和安全性。下一步的工作主要集中在进一步拓展系统支持的应用类型。

**参考文献:**


[1] Danseglio M, Dillard K, Maldonado J, et al. Windows Server 2003 Security Guide[EB/OL]. Microsoft Solutions for Security and Compliance group (MSSC). http://www.microsoft.com/technet/security/prodtech/windowsserver2003/w2003hg/sgch00.mspx,2005-12.

[2] Marsh K. Win32 Hooks[EB/OL]. Microsoft Developer Network Technology Group. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwui/html/msdnhooks32.asp,1994-02.

[3] TheWinPcap Team. WinPcap Documentation[EB/OL]. http://www.winpcap.org/docs/docs31/html/main.html,2005-12.

[4] Degioanni L. Development of an Architecture for Packet Capture and Network Traffic Analysis[D]. Turin, Italy: Politecnico Di Torino,2000.

[5] Risso F, Degioanni L. An Architecture for High Performance Network Analysis[A]. Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC 2001)[C]. Hammamet, Tunisia:[s.n.],2001.