个人防火墙系统的设计与实现

严鹏 曾文方 于小红 (成都 四川大学计算机学院 610065)

摘要：本文根据现在市场上常见的防火墙存在的“只防外不防内”的缺点，提出了一种新型的防火墙设计方案来保护桌面操作系统，以弥补常见防火墙存在的上述缺点，使企业内部局域网的主机间网络通信更加安全、可靠。

关键词：TCP/IP 防火墙 Ndis Vxd

1 实现个人防火墙系统的必要性

随着计算机网络技术的迅猛发展和Internet在全世界范围内的日益普及，以及许多新的网络服务的出现，计算机网络安全问题变得越来越严重。为此，对于网络安全方面的研究也越来越多，主要包括数据加密和防火墙技术的研究。其中，防火墙产品的研制和实现是一个热点。但市场上大多数的防火墙产品仅仅是网关型的，如check_point公司的Firewall_1和Sun microsystem公司的SunScreen，这些产品主要解决企业内部与Internet互连方面的安全问题。虽然它们的功能相当强大，但由于它们基于下述假设：内部网是安全可靠的，所有的威胁都来自网外。因此，它们只“防外不防内”，难以实现对企业内部局域网内的主机之间的安全通信，也不能很好的解决每个拨号上网用户所在主机的安全问题。

目前，对主机的攻击越来越多。一般都是利用操作系统设计的安全漏洞或者通信协议的安全漏洞来实现攻击。如假冒IP包对通信双方进行欺骗；对主机大量发送IP数据包来对主机进行轰炸攻击，以达到使主机崩溃的目的；还有经常出现的蓝屏攻击等。

因此，为了保护主机的安全通信，在主机上安装个人防火墙系统很有必要。

2 个人防火墙系统的功能要求

个人防火墙系统主要目的是保护主机的安全通信。因此，个人防火墙系统应该能对进出主机的数据包依据数据包的源IP地址、目的IP地址、源端口号、目的端口号等，根据用户定义的过滤规则对数据包进行过滤，还可以对数据进行加密，完整性检查，身份认证。在必要时，还可以对发生的事件进行日志记录，以便用户能根据记录调整自己的过滤规则，更好的对主机通信进行控制。

3 个人防火墙系统的设计与实现

3.1 利用NDIS技术和虚拟设备驱动(VxD)技术实现包过滤

NDIS（Network Device Interface Specification）是一个网络接口规范。它位于Nic和数据链路层之间。防火墙通过Ndis API函数来访问Nic，在Nic和网络层协议之间起到桥梁作用。其网络接口如图1所示。

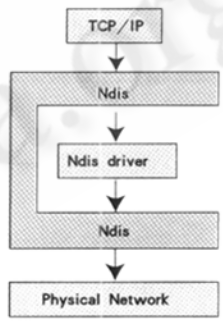

图 1

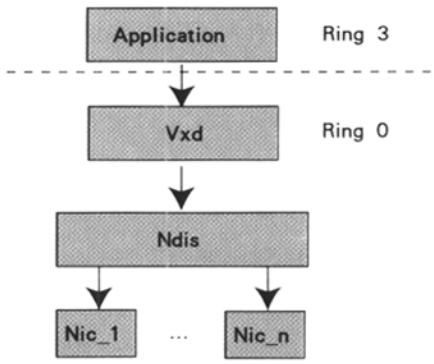

图 2

Vxd （Virtual Device Driver）是用来扩展 Windows 操作系统功能的一类程序。它以 DLL 的形式链入 Windows 操作系统的核心层（Ring 0）。Vxd 主要解决不能被 Ring 3 层应用程序处理的一系列问题。由于 Vxd 运行于操作系统的核心层，所以它具有最高特权级，它能完成任何功能，因此对于上层应用程序，利用 Vxd 技术实现包过滤功能都是透明的，它支持各种Ndis兼容网卡。它与上面应用程序和下面 Nic 的关系如图 2 所示。

在应用程序与Vxd之间，防火墙需要将过滤规则传给 Vxd。对于过滤规则，可用图形化界面来生成和修改。其规则采用文件形式进行存放。当应用程序将规则传给Vxd时，将指向该文件的指针(如 *fp)通过函数 CreateFile("＼＼＼＼.＼＼firewall.vxd", PACKET_MUD,fp,sizeof(fp), NULL,NULL,NULL,NULL)传给 Vxd。其中过滤规则格式如下表所示：

过滤规则号	允许/拒绝	源 IP 地址	目的 IP 地址	源端口号	目的端口号	协议
1	允许	202.115.*.*	202.115.60.207	*	*	TCP
2	拒绝	202.116.*.*	202.115.60.207	*	*	TCP
3	允许	202.115.60.207	*	*	*	TCP

其过滤实现流程如图 3 所示。

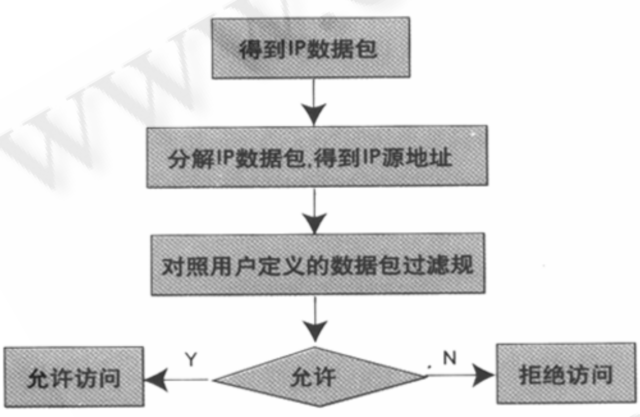

图 3

3.2 利用 IPSec 对数据进行加密处理

在数据加密时，为了使安全功能模块能兼容 IPv4 和 Ipv6,在防火墙设计中引入了IPSec协议。IPSEC是由IETF正式定制的开放性 IP 安全标准，是虚拟专网的基础，已经相当成熟可靠，同时IPSEC的厂家支持广泛，Microsoft 的NT5、Cisco PIX防火墙、Ascent Secure Access Control 防火墙都支持 IPSEC 标准。IPsec 协议主要包括 AH(authentication Header)和 ESP(Encapsultating Security Payload)两个子协议。ESP 的基本思想是对整个 IP 包或更高层协议的数据进行封装，并对ESP的数据进行加密，它有 Tunnel 方式和Transport 方式两种模式。Tunnel 方式将整个IP包加密后作为新的IP包的数据段，并生成一个新的IP包；Transports 方式只对原IP包中的数据进行加密并发送出去。本防火墙采用 Transports 方式。

在Vxd的编写中，对于将某些网络功能（如发送数据包等）进行挂起是至关重要的。在该防火墙中，用一些用户定义的函数来代替原来的某些系统函数，先对接收到的数据包进行过滤、加密等处理，然后再转到原来的系统函数上对其进行进一步的处理。例如在下面的代码中，NdisSend是系统函数，NSHIM_Send是用户定义函数，其中完成了对进出数据包的加密处理。

这是 HOOK 的代码：

```
BOOL OnSysDynamicDeviceInit()
{
///// 将系统函数 NdisSend 挂起，而用用户定义函数 NSHIM_Send 来实现数据加密功能，并得到系统函数 NdisSend 的地址。
realNdisSend=(void (NDIS_API *) (PNDIS_STATUS, NDIS_HANDLE, PNDIS_PACKET))Hook_Device_Service_C(__NdisSend,myNdisSend,&NdisSendThunk);
return TRUE;
}
VOID _stdcall myNdisSend(
        OUT PNDIS_STATUS Status,
        IN  NDIS_HANDLE  NdisBindingHandle,
        IN  PNDIS_PACKET Packet
        )
{
    .............
```

```
        switch( Querytable(SendIPHeader,QUERY_SEN-
D_PACKET,&key) ){
      .....
     case PACKET_MUD :// 完成数据包的加密
          CurSendPktLen = DES_BCB_Encrypt(
   (PUCHAR)SendIPHeader+IP_HEADER_LEN,
CurSendPktLen-MAC_HEADER_LEN-IP_HEA-
DER_LEN,sndkeyno);
              CurSendPktLen += (MAC_HEADER_LEN
              +IP_HEADER_LEN);
             Old_cksum = SendIPHeader->HeaderCRC; //
             保存以前的校验和
             SendIPHeader->HeaderCRC = 0;
    LengthL   =((count*8+IP_HEADER_LEN+AH_TOTA-
LLEN+4+4+8)&0xff00)>>8;
          LengthU=(count*8+IP_HEADER_LEN+AH_
          TOTALLEN+4+4+8)&0xff;
          SendIPHeader->Protocol=AH_PROTOCOL;
         SendIPHeader->Length=(LengthU<<8)+LengthL;
         New_cksum = checksum((PUCHAR)SendIPHe-
ader,20); // 得到新校验和
           SendIPHeader->HeaderCRC = New_cksum;
           break;
          }
          .....
realNdisSend( Status, NdisBindingHandle, SendPktList
[SendPktListHead]); // 调用系统函数 //NdisSend
    }
.....
}
VOID NDIS_API NSHIM_RegisterProtocol(
    OUT PNDIS_STATUS Status,
    OUT PNDIS_HANDLE NdisProtocolHandle,
    IN PNDIS_PROTOCOL_CHARACTERISTICS
    ProtocolCharacteristics,
    IN UINT CharacteristicsLength
    )
{
     .....
     realNdisReceive = ProtocolCharacteristics-
>ReceiveHandler;
     ProtocolCharacteristics->ReceiveHandler =
myNdisReceive;
     realNdisSendComplete = ProtocolCharacteristics-
>SendCompleteHandler;
     ProtocolCharacteristics->SendCompleteHandler =
myNdisSendComplete;
    realNdisOpenAdapterComplete=ProtocolCharacteristics-
>OpenAdapterCompleteHandler;
    ProtocolCharacteristics->OpenAdapterCompleteHa-
ndler= myNdisOpenAdapterComplete;
    realNdisRegisterProtocol(Status,NdisProtocolHandle,
ProtocolCharacteristics,CharacteristicsLength   );
    .....
    return;
}
```

4 结束语

本防火墙完成了在IP层对数据加密，在一定程度上解决了主机网络通信方面的安全问题，加强了对桌面操作系统的保护。但对于防火墙关于网络病毒的防范、基于内容的过滤、网络访问控制、安全网络协议的研究以及相应标准的制定，将成为今后几年防火墙研究的热点，防火墙技术必将朝着智能化的方向发展。■

参考文献

1 周明天，汪文勇。TCP/IP网络原理与技术，清华大学出版社。

2 杨强，李堂秋编著。Win9x 虚拟设备驱动程序编程指南，机械工业出版社。

3 刘渊、乐红兵等编著。因特网防火墙技术，机械工业出版社。

4 Microsoft Inc 。Microsoft DDK for Win98