Windows 平台通用个人防火墙的分析与实现①

Analysis and Realizing of Common Personal Firewall System on Windows Platform

刘鹏远（湖北经济学院 计算机学院 湖北省武汉市 430205）

摘 要：给出一种 Windows 平台通用个人防火墙的分析与实现方法。首先定义个人防火墙系统应具备的主要功能，认为其核心技术是对网络数据包进行包过滤。然后给出 Windows 系统网络协议分层体系结构，对 OSI 参考模型和 Windows 网络体系结构对比分析的基础上给出实现包过滤的不同技术路线。最后对各技术路线进行评估，选择 SPI 作为实现方案并给出实现包过滤的工作流程。应用表明，该个人防火墙具有良好的通用性和较好的响应性能。

关键词：个人防火墙 包过滤 TDI NDIS SPI

1 引言

防火墙技术在网络安全领域是数据加密与签名外的另一个核心要点。个人防火墙位于宿主计算机和它所连接的网络之间，监视着宿主计算机上所有进入和流出的网络通信。个人防火墙系统实现所涉及的技术和企业级的网络防火墙、硬件防火墙基本相同，其中对 Windows 平台个人防火墙系统的实现还需理解 Windows 网络底层协议底层，难度很大。Windows 下的个人防火墙开发是商业领域的核心机密，也是核心的网络安全技术，对其分析实现具有理论和实践意义。

本文第 2 节分析对比国内外商业个人防火墙后提出防火墙功能定义，得出核心技术要点是网络数据包截获。第 3 节给出 Windows 网络体系结构（AWN）分析。第 4 节进行技术路线分析评估。第 5 节给出 SPI 包过滤实现流程。第 6 节做了小结展望。

解决问题的第一步是理解问题，其次要发掘出主要问题和核心要点才能找到突破的障碍。本文第 2 节对比商业个人防火墙后给出了对其功能定义，并分析得出了其核心是访问网络数据包的包过滤。第 3 节通过对 Windows 网络体系结构（AWN）的分析，提出了实现包过滤的可能的技术路线。第 4 节对这些技术路线进行了评估选择，结合项目目标选择了 SPI 包过滤技术作为技术路线来实现 Windows 平台下的通用个人防火墙。第 5 节以应用程序发出 WSPSend（或 Send）访网为例，给出了该包过滤过程中涉及的几个核心函数的工作流程说明。第 6 节对使用 SPI 包过滤实现的个人防火墙系统做了简单评价，指出了不足和未来的工作。

2 功能定义与核心技术要点定义

通过对 LOCKDOWN、NORTON、金山网镖、天网等商业个人防火墙的分析，归纳其具有以下功能行为[1]：

（1）实时监控，根据应用程序规则对进出本机的网络封包进行过滤；

（2）受到攻击时向用户报警指示；

（3）日志记录网络访问动作的详细信息；

（4）电子邮件监控，可以根据自定义的过滤规则对邮件实施过滤；

（5）根据特征库进行入侵检测；

（6）在线升级特征库；

（7）将防毒、杀毒和个人防火墙集成在一起。

防毒杀毒将病毒特征库的数据和包过滤数据进行对比，判断是否是病毒从而干预；入侵检测和各种规则

① 基金项目：湖北省教育厅重点项目（B200619001）

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

过滤(包括应用程序规则和电子邮件规则)是基于包过滤结合自定义控管规则进行分析判断是拦截还是放行,而关于特征库的定义和自学习不是防火墙技术的研究内容。因此,防火墙软件技术的核心是包过滤——对网络上流动的数据包过滤并分析,通过控管规则来决定放行或者禁止出/入。

3 Windows 网络体系结构(AWN)分析

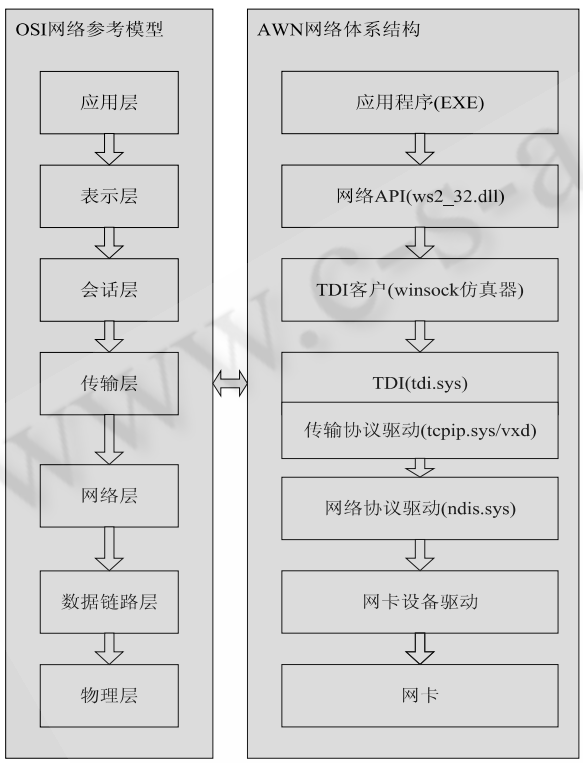

图 1　OSI 网络参考模型与 AWN 对比

图 1 给出了 OSI 参考模型[2] 与 Window 网络体系结构(AWN)[3] 的对比,因此,包过滤可以发生在网卡驱动程序所在的数据链路层及其以上直至应用层的各层中,这提供了拦截网络数据包的基本思路。

在用户态下进行网络数据包拦截就是指会话层和表示层的包过滤,利用 Winsock SPI(Service Provider Interface,服务提供程序接口)或直接替换系统自带的 Winsock 动态链接库来过滤包。TDI 层及以下层的操作须借助它提供的一些接口和开发规范进行,操作系统工作在系统态下。涉及 Windows DDK(windows device developing kit)、TDI(Transport Layer Device Interface Filter Driver)和 NDIS 接口规范(Network Driver Interface Specification)[4]。

在系统态下首先看最底层,从 AWN 看过滤可发生在网卡驱动程序所在的数据链路层。但防火墙需要从数据包过滤中得到 IP 地址、协议服务类型和应用程序信息,在网络层进行过滤已足够可获得这些信息,没必要对网卡 MAC 帧进行过滤分析。而且在网卡上进行包过滤还要适应不同的网卡硬件环境,相当于额外还需开发通用网卡驱动程序,这与项目目标不吻合。从 AWN 来看,系统态下的包过滤还可使用 TDI 层过滤驱动或网络层上的 NDIS 中间层驱动,前者是 TDI 接口规范后者是 NDIS 接口规范。

4 技术路线分析评估

个人防火墙的工作是监控网络进出的数据流,对用户认为危险或者有害的数据流向进行禁止或者监控,其核心功能是网络数据包的监控于分析过滤。从底层看,NDIS 的中间驱动由于是在网卡驱动程序和传输驱动程序之间插入了一层,所以可以过滤较为底层的封包,可以完成更为低级(最底层的是在网卡驱动程序层过滤,但前面阐述过在网卡驱动程序层进行包过滤没有实现价值)的操作,不会有网络数据包从这里旁路,因此其最大的优点是安全系数高,但需要指出的是,NDIS 层次上的网络操作不采用标准的 I/O 模式(IRP)形式,因此不能确定某个网络操作是由哪个进程引起的。个人用户看不到网络数据是源于哪个进程,就不容易让用户自定义过滤包规则,这是个非常大的遗憾和缺陷。当然,越靠近底层的驱动程序编写可移植性和健壮性越难保证,编码复杂也是其一个缺点。

对于 TDI 过滤驱动程序,由于采用标准的 Windows I/O 请求(IRP),它没有 NDIS 中间层驱动的不能得到进程信息的缺点,但由于它工作在传输驱动程序 Tcpip.sys 之上,由 Tcpip.sys 接收后直接处理的数据包是不会传递到上层 TDI 过滤驱动程序的,如 ICMP 协议的应答包。Ping 和 Tracert 就是利用 ICMP 来探测网络的可达性和跟踪路由。

NDIS 中间驱动和 TDI 过滤驱动,都是 32 位 Windows 平台上才提供的方法。整个 NDIS 规范和 TDI 的概念,是在 Windows NT 平台上提出和得到发展的,以后的 Windows2000,Windows XP 都支持,但以前的

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

Windows98 和 Windows Me 都不支持。因此如果要开发一个通用的 Windows 平台下的个人防火墙软件,这两种方法是不适合的。

用户态下的 SPI 包过滤使用 DLL 监控使用 Winsock 调用进行网络通信的网络数据包。它工作在 TDI 客户之上,所有的用户进程之下,因此对于用户进程交给它的网络请求和意图非常清楚——在经过底层的分段(IP 分段)之前,对用户进程的行为,目的可以更直观的了解,非常适合做内容过滤[5]。个人防火墙系统应能详细记录各种进程的访问网络信息,而在应用层进行包过滤能得到最完备的数据包信息,因而能实现记录最丰富网络访问信息的个人防火墙系统。并且其过滤的所有 Winsock 调用广泛地被所有 Windows 平台支持。综上所述,选择 SPI 包过滤作为技术路线。

5 SPI 包过滤实现流程

个人防火墙系统分为前台界面进程和后台监视进程两部分。后台负责对宿主机上一切访网调用的监视和包截获过滤,前台从共享数据区取得处理结果展现到用户界面。本节以应用程序发出 WSASend(或 Send)访问网络调用被个人防火墙系统的后台截获为例,给出其进行包过滤实现流程。定义了几个核心函数(限于篇幅不给出这些函数的详细处理流程),图 2 对此给出了整体流程描述,其中上下层的箭头表示嵌套调用,同层中从左到右表示被上层调用的先后顺序。

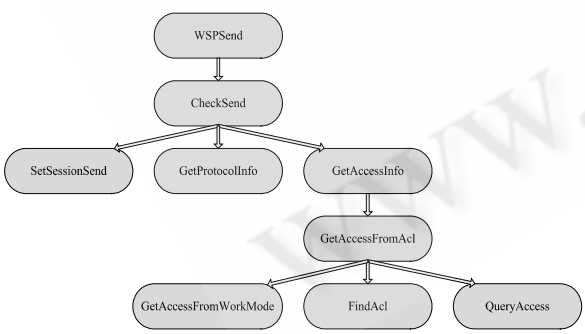

图 2 WSASend 包过滤实现流程

以 WSPSend 为例,当 WSASend 调用被截获后,首先调用 CheckSend 检查连接权限,如果 CheckSend 没有返回“ PASS ”, WSPSend 就不转发给底层函数。CheckSend 其实是先设置调用 SetSession 封包,然后调用 GetAccessInfo 来进行访问权限检查,GetAccessInfo 是直接调用 GetAccessFromAcl 进行检查。GetAccessFromAcl 首先检查前台界面进程是否已启动,没有启动就直接放行过滤的网络访问调用;然后检查进程名无效也放行;检查当前封包的远端 IP 是自身也放行;然后 GetAccessFromAcl 调用 GetAccessFromWorkMode 检查工作模式是否为询问模式,如果不是询问而是拒绝所有或放行所有就直接返回 PASS 或 DENY。

在询问的工作模式下,GetAccessFromAcl 调用 FindAcl 按应用程序路径和名称查找。FindAcl 是根据应用程序路径和名称从指定下标开始在控管规则中查找与之相匹配地记录,注意它只返回第一个满足的下标。如果 FindAcl 第一次查找没有找到说明在控管规则中没有该应用程序进程的控管规则,返回 QUERY 表示需要调用 QueryAccess 向用户询问后返回访问动作。

如果 FindAcl 找到还要继续判断进出方向和当前连接是否一样;判断是否协议类型相同;判断是否端口相同;判断是否当前连接封包的时间在这条控管规则的允许访问时间段内;判断是否当前连接封包的远端 IP 在这条控管规则的允许访问网络类型内。如果都相同直接取这条控管规则的访问动作 PASS 或 DENY。这递进的几步中有一步不能推进时已说明不是第一次查找,再次调用 FindAcl 从新的下标开始一次匹配查找,并在找到后继续上面的递进步骤。一旦不满足同上面的做法——FindAcl 从新的下标开始一次匹配查找,直至 FindAcl 没有找到或者完全满足。完全满足时就对该封包取该条控管规则的访问动作,不是第一次调用 FindAcl 而没有找到时,就直接取上次找到的那条控管规则中的相反的访问动作。

GetAccessFromAcl 函数是 CheckXxx 管制函数的核心部分,可以看到它具有自学习确定控管规则——当控管规则中没有匹配的进程名时询问用户来定义,以及一定的智能——当没有控管规则与当前封包完全匹配时,取最接近该条控管规则的相反的访问来动作。

6 展望

作者已使用 SPI 包过滤实现了一 Windows 下的个人防火墙,对发送方的截获抢先于金山网镖等,响应性能较好。但也应看到 SPI 包过滤存在被旁路的风险。应用程序若使用 TDI 接口提供函数例程直接发起访网

(下转第 127 页)

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

(上接第 111 页)

通信,工作在其之上的 SPI 包过滤后台进程对此无法感知。与此类似,那些由 Tcpip. sys 接收后直接处理的数据包,如利用 ICMP 协议的应答包进行探测网络可达性的 Ping 和 Tracert 等应用,由于位于 TDI 的上层,SPI 包过滤对此旁路也无能为力。由第 4 节分析知,若不考虑通用性,NDIS 的优点就是 SPI 的缺点,因此,采用两者结合的复合包过滤将是更好的实现方案。

参考文献

1 黄允聪,严望佳. 计算机网络安全工具. 北京:清华大学出版社,1999. 35 – 41.

2 Andrew S. Tanenbaum. 计算机网络. 第四版. 北京:清华大学出版社,2004. 60 – 62.

3 Marcus Goncalves. Firewalls:A Complete Guide. 北京:机械工业出版社,2000. 77 – 79.

4 Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman. Building the Internet Firewall. 2nd ed. America: McGraw – Hill, 2003:90 – 95.

5 王树华,周利华. 基于 Windows 2000 平台的防火墙技术研究与实现. 微机发展,2004,14(5):78 – 80.

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn