

# 网络分流诱捕系统的设计和实现

韩俊杰, 康 乐, 刘胜利

(上海交通大学计算机科学与工程系, 上海 200030)

**摘 要:** 提出了一种 Windows 环境下网络分流诱捕系统的设计和实现方法。它根据 IDS 规则库, 利用 Windows DDK 网络驱动 NDIS 中间层技术实现网络流量的过滤和分流, 为 Honeypot 收集大量非法流量, 提高了 Honeypot 的效率, 同时阻隔面向真实服务器的攻击流量, 保护了真实服务器。

**关键词:** Honeypot; NDIS; 入侵检测

## Design and Implementation of Traffic-analysis-based Honeypot System

HAN Junjie, KANG Le, LIU Shengli

(Dept. of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200030)

**【Abstract】** A traffic-analysis-based honeypot for Windows system is designed and implemented. Based on the library of IDS rules, the network driver of Windows DDK and NDIS middle-ware technology are used to realize filtering and diversion of network flow. This mechanism diverts the unauthorized flow from reaching the real server. In the meantime, it collects a vast amount of “hacking techniques” from the unauthorized flow to continuously adapt the system to the various “hacking techniques”. It improves the efficiency of Honeypot and protects the real servers.

**【Key words】** Honeypot; NDIS; IDS

Honeypot 是近年来兴起的一项全新的从战争欺骗思想发展而来的网络安全技术。通过设置诱骗服务来诱惑已入侵到网络系统的攻击者将时间和精力花费在攻击 Honeypot 上, 从而减小或避免真正的网络系统受到攻击的机会。

在实际应用中, 一方面, Honeypot 并不直接面对攻击者, 攻击者只有通过扫描等技术来发现它, 如何提高 Honeypot 的密度来吸引攻击者一直是 Honeypot 领域的重要课题。另一方面, 真实主机的地址是众所周知的, 特别是局域网里, 真实主机完全暴露在攻击者面前, 承受着攻击者的各种攻击尝试。如果能把攻击流量主动引入 Honeypot, 那么既能提高 Honeypot 的密度, 又能降低真实主机面临的威胁。本文设计了一种根据 IDS 规则库, 利用 Windows DDK 网络驱动 NDIS 中间层技术实现网络流量的过滤和分流, 以提高 Honeypot 的效率, 同时降低真实系统受到攻击或随机刺探的可能性的方法, 并进行了部分实现。

本文首先介绍了 HoneyPot 陷阱技术和 Ndis 中间层网络驱动技术, 然后详细说明了基于这两种技术的 Windows 环境下网络分流诱捕系统的设计和实现。

### 1 HoneyPot 陷阱技术

Honeypot 就是建立一个虚拟的环境, 上面装有模拟或真实的操作系统和应用程序, 故意留有各种弱点或漏洞, 引诱黑客来攻击, 从而监视、学习并分析攻击行为, 进而提高自己系统或网络的安全系数。

传统 IDS 普遍的问题之一便是它们会产生大量的警示, 其中有很多误报。这些庞大的“噪声”会很耗费时间和资源。如果 IDS 持续产生误报, 那么管理员最后可能会开始忽略这个系统。而对于 Honeypot, 与之有关的活动, 就定义上来说都是未经授权的, 都是非法流量。这样, 少量的资料收集可

以让辨识和针对未经授权的活动采取行动变得更加容易。使用 Honeypot 可以消耗攻击者的时间, 会让攻击者对于现有的安全措施产生错误的印象。如果搭配其他技术如防火墙和 IDS 等一起使用, 能够形成企业级的坚固信息安全架构。

Honeypot 的部署策略有: 布雷区(Minefield), 防护罩(Shield), Honeynet 等。在 Honeypot 的部署中, 每个 Honeypot 都会和一台或几台它所保护的服务器作为配对。进出服务器的正常流量不受影响, 但指向这台服务器的任何可疑通信, 都会交由 Honeypot 处理。作为真实服务器的代替, Honeypot 不仅能侦测到潜在的攻击, 还可以代替实际的攻击目标做出响应, 以吸引后续的攻击, 保护真实服务器。

为了使 Honeypot 能够像真实系统一样对网络请求做出反应, Honeypot 应该部署在 DMZ 区中, 它的配置应该尽量接近真实系统以提高诱捕的价值, 可以在 Honeypot 上安装和真实系统一样的操作系统和应用软件, 从它所防护的服务器复制部分或全部的非机密内容, 人为地留下多种漏洞等待攻击。为了让 Honeypot 能记录下攻击者在其中的一切活动, 同时不能被攻击者作为跳板攻击别的系统, Honeypot 要配置严密的日志记录措施如远程日志服务器以安全地保存攻击者的踪迹, 设置外连限制措施防止 Honeypot 成为攻击跳板。

Honeypot 在单一部署下可以顺利运作, 但是效率通常不高。孤立的 Honeypot 在吸引攻击者和保护服务器方面能力是有限的。本文考虑使用一种类似于智能防火墙和路由器的技术, 根据 IP 地址、目的连接端口、通信内容来过滤网络通信,

**基金项目:** 国家自然科学基金资助项目(60303026)

**作者简介:** 韩俊杰(1972 - ), 男, 硕士生, 主研方向: 信息安全技术; 康 乐, 硕士生; 刘胜利, 副教授

**收稿日期:** 2006-01-28 **E-mail:** lilack@citiz.net

然后根据策略来做入侵流量的重定向,以实现正常流量和攻击流量的分离,降低真实系统受到攻击或随机刺探的可能性,保护真实服务器。通过 Honeypot 的应答来留住攻击者,提高 Honeypot 的效率,收集到大量攻击信息用于研究分析。在 Windows 系统中,通过 NDIS 中间层驱动技术实现这个方案。图 1 是 Honeypot 的部署。

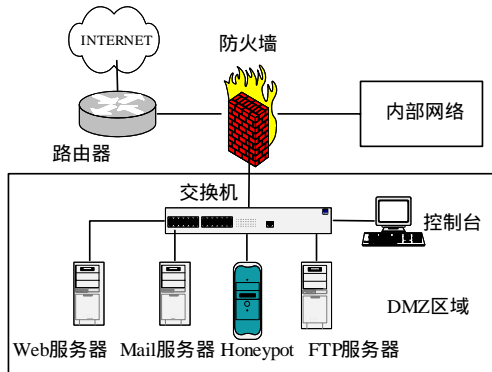


图 1 Honeypot 的部署

## 2 NDIS 中间层网络驱动技术

Windows 操作系统总体架构分为 2 个层次:上面为应用层,下面为核心层。应用程序(.exe)和动态连接库(.dll)工作于应用层,驱动程序(.sys, .vxd)工作于核心层。这种分层结构有利于实现代码共享和安全保护。

对于 Windows 操作系统的网络架构,物理层就是网卡,数据链路层是网卡驱动程序,网络层是 NDIS,传输层是 TDI,会话层是 SPI,表示层是 API。其中,NDIS 是 Windows 的网络驱动接口规范,从网卡驱动到协议驱动都要利用 NDIS 这个规范来操作。由于 NDIS 为整个网络驱动提供接口,它其实横跨数据链路层、网络层和传输层这 3 层,它的工作核心是提供网络层接口。

NDIS 支持编写 3 种类型的网络驱动程序:Miniport 驱动程序,中间层驱动程序,Protocol 驱动程序。中间层驱动程序位于 Miniport 和 Protocol 中间,并同时具有 Miniport 和 Protocol 两种驱动程序接口。在工作时,它在自己的上下两端分别开放出一个 Miniport 接口和一个 Protocol 接口,其中位于上面的 Miniport 接口和上层驱动程序的 Protocol 接口进行对接,下面的 Protocol 接口和底层驱动程序的 Miniport 接口对接。利用中间层驱动程序可以在网卡驱动程序和传输驱动中间插入一层自己的处理,从而实现截获网络封包并重新进行封包、加密、网络地址转换、过滤等操作。图 2 是 NDIS 中间层驱动程序结构。

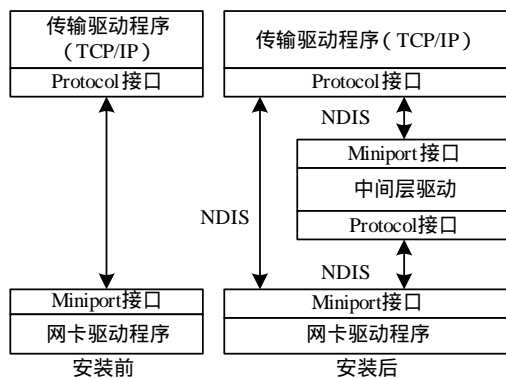


图 2 NDIS 中间层驱动程序结构

中间层驱动技术与用户级的数据包拦截技术如 SPI 相比,用户级的拦截有其优势,实现方便、便于移植、通用性

强,但是,在用户级不能得到所有的数据报。而中间层驱动程序是属于内核级的,它位于网卡和传输驱动程序之间,所以它可以截获较为底层的封包,可以完成更为低级的操作,安全系数也较高。使用中间层驱动技术的优势非常明显。首先,在驱动级别上做过滤和重组,速度快,效率高;其次,所有的数据报无一例外,只要网卡上传的数据报均可以截获,避免了用户级无法得到所有数据报的缺点。

利用 NDIS 中间层驱动技术,可以在中间层拦截网络流量,加入我们想要过滤的数据报的特征,实现基于中间层驱动的内核级包过滤和包重组重定向,从而把真实服务器和 Honeypot 关联起来,既阻止了通向真实服务器的非法流量,保护了服务器,又把非法流量引流到 Honeypot,增加了 Honeypot 的流量收集途径,为记录和研究攻击行为提供大量的内容。

## 3 Windows 环境下网络分流诱捕系统的设计和实现

本系统中,用户对真实服务器的请求信息首先进入核心分流系统,系统根据 IDS 规则库判断流量是否非法并进行路由分流,正常流量进入真实服务器,非法流量则转到蜜罐机以供后续监控研究。本系统包含基于中间层驱动的核心分流系统、IDS 分流规则库、蜜罐机和远程日志服务器。

### (1)体系结构

图 3 所示的是体系结构。

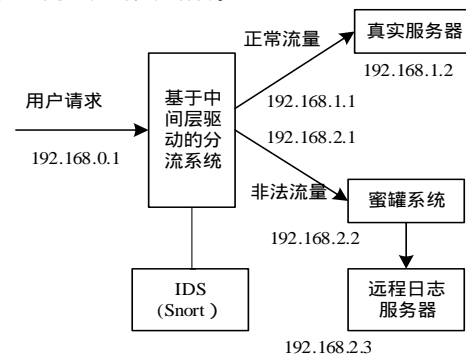


图 3 体系结构

1)真实服务器和 Honeypot 将安装同样的服务器应用软件以提高交互能力,二者的区别在于真实服务器打上各种补丁而 Honeypot 留有各种漏洞。

2)系统核心为一个基于中间层驱动的分流系统。它拥有 3 个网络接口,分别与用户、真实系统、Honeypot 相连接。3 个接口将在不同的网段,以提高防护等级。分流系统将利用入侵监测系统的规则库作为合法和非法的判断依据进行 IP 包转发。

3)非法流量将只转发到蜜罐系统以保护真实系统不受侵犯。

4)Honeypot 将记录下所有的流量,并把日志存入远程日志服务器数据库,以供后续分析。

### (2)核心分流系统的实现

NDIS 中间层驱动接口对上层显示为一个 Miniport 驱动,类似于网卡驱动程序,对下显示为一个 Protocol 驱动,类似于协议层驱动程序。这里中间层驱动用 PASSTHRU 类来实现。

#### 1)PASSTHRU 的初始化

Passthru 通过 DriverEntry 函数把自身注册为一个中间层的 Protocol 和 Miniport 驱动,通过 BindAdapterHandler 函数打开下面的物理网卡,通过 UpperBindings 关键字得到物理

网卡的虚拟网卡,绑定到自身,初始化每块虚拟网卡。对于上面的传输层 Passthru 中间层驱动输出一系列 MiniportXXX 函数在它的上边沿并与上层的 Protocol 驱动连接。这样,中间层驱动对上面传输层来说是一个虚拟的网卡,对下面的链路层来说就是一个虚拟的传输层。

#### 2)PASSTHRU 包的接收

Protocol 设备位于中间驱动程序的下方,重点处理包的接收。接收函数主要包括 ProtocolReceive 和 ProtocolReceive Packet,分别用于不同类型的网卡。

当 Passthru 中间层驱动收到下层网卡驱动来的包时,它通过 NdisGetReceivedPacket 得到下层驱动收到的包,并用 NdisMIndicateReceivePacket 或 NdisMXxxIndicateReceive 向上指示数据到达。如果包比较大,将调用 NdisTransferData 来要求下层网卡把剩余数据传上来。

NdisMTransferDataComplete 是 TransferData 的回调函数,表示数据传输已经完成。NdisMXxxIndicateReceiveComplete 是 NdisMXxxIndicateReceive 的回调函数,表示数据接收已经完成。

#### 3)PASSTHRU 包的发送

Miniport 设备处在中间驱动程序的上方,重点处理包的发送。发送函数主要有 MiniportSend 和 MiniportSendPackets,使用哪个取决于注册了哪个 Miniport 设备。

Passthru 使用 NdisSend 或 NdisSendPacket 往下层发送一个或多个包。发送请求完成后,Passthru 调用 NdisMSendComplete 函数来表示发送操作已完成。

#### 4)PASSTHRU 包的过滤和重组

为了实现包的过滤和重组功能,需要在 Passthru 接收到数据报的时候进行规则判断,也就是说,修改 Passthru 的接收部分和发送部分,读取接收到的内容,判断是合法流量还是非法流量,然后更改数据包内容,根据不同情况路由到真实主机或 Honeypot。用 NdisQueryPacket 函数取得包的内容,根据 TCP/IP 的结构找到 IPHeader、TcpHeader、UdpHeader 等,根据不同的协议类型找到它的数据区,通过数据匹配判断是合法包还是非法包,实现过滤。

根据判断的不同情况,修改包的 IP 字段,实现重组和转发。为了实现转发,需要分配一段新内存,给包一个新的描述符并把内容复制到新内存中,可以用 NdisMoveMemory 来实现。然后,发送新的包,并对旧的包通过设置状态 Ndis\_Status\_Not\_Accepted 来阻止向上层转发。

#### (3)IDS 规则库的应用

鉴于每天都会出现新的攻击行为,使用入侵检测系统的入侵特征库作为 Passthru 的判断规则库会比较有效。这里使用广泛又开放原码的 SNORT 规则库。

把规则库作为一个文件放在分流器所在的机器上。在

Miniport 初始化时,利用 NdisOpenFile 函数打开磁盘文件,然后用 NdisMapFile 把打开的文件的内容放到一个缓冲区里,从而得到分流规则,在后续程序收到包时加以利用。

#### (4)日志记录

数据收集是设置 Honeypot 的另一项技术挑战。Honeypot 监控者只要记录下进出系统的每个数据包,就能够对攻击者的所作所为为一清二楚。Honeypot 本身的日志文件也是很好的数据来源。但日志文件很容易被攻击者删除,所以通常的办法就是让 Honeypot 向在同一网络上但防御机制较完善的远程系统日志服务器发送日志备份。为了安全起见,其间的通信可以严格控制并设置为专线、VPN 等。

#### (5)实验结果

为了测试本系统的有效性,针对 IIS5.0 的 UNICODE 漏洞进行了试验。此漏洞攻击特征码是“%c1%1c”,即 IIS 中“%c1%1c”解码成了(c1-c0)\*40+1c=5c=“\”,可能造成突破 IIS 路径访问到上级目录。实验中,建立规则文件 rule.txt,内容包含“/..%c1%1c./”。客户机 A,IPA=192.168.2.2,A 的网关为 192.168.2.1。分流机 B,2 块网卡,IPB1=192.168.2.1,IPB2=192.168.0.169。蜜罐机 C,IPC=192.168.0.143,C 的网关为 192.168.0.169。真实服务器 D 由于设备限制,在此实验里是个虚拟地址,IPD=192.168.3.1。

实验中,在客户机上访问 192.168.3.1,数据包由网关蜜罐机处理,发现特征码时就发往蜜罐机。同样,蜜罐机回复的数据包也由网关蜜罐机处理,更改为真实主机地址,然后发回给客户机。经实验发现,非法流量能够成功地转向到蜜罐机。

## 4 总结

在网络迅速发展的今天,安全仅靠单一的技术是远远不够的,只有多种技术相结合,从底层的操作系统和驱动着手,朝着多样化、智能化、系统化的方向发展,才能有一个真正安全的环境。本文提出了一种中间层网络驱动和新兴的 Honeypot 技术相结合实现 Windows 环境下网络分流诱捕系统的方法,并进行了实验测试。关于系统的效率和移植性问题,将是我们进一步的研究方向。

## 参考文献

- 1 Lance S. Honeypot: 追踪黑客[M]. 邓云佳,译. 北京:清华大学出版社,2004.
- 2 朱雁辉. Windows 防火墙与网络封包截获技术[M]. 北京:电子工业出版社,2002.
- 3 Brian C. Snort2.0 入侵检测[M]. 北京:国防工业出版社,2004.
- 4 基于IMD的包过滤防火墙原理与实现[Z]. <http://www.xfocus.net>.
- 5 入侵检测系统:诱捕式网络技术[Z]. <http://www.symantec.com>.
- 6 TCP-IP详解,卷1:协议[Z]. <http://www.china-pub.com/>.

(上接第 135 页)

- 12 杨树国,李春霞,孙尧,等.小波域图象零水印技术研究.[J].中国图象图形学报,2003,8(6):664-669.
- 13 杨红梅,张承明,张问银.一种基于小波变换的零水印算法[J].山东农业大学学报,2004,35(3):407-409.

- 14 季称利,杨晓云.结合空域不变量的变换域零水印二次检测方案[J].计算机工程,2004,30(14):105-107.
- 15 温泉,孙峰,王树勋.零水印概念与应用[J].电子学报,2003,31(2):215-216.
- 16 周东其.零水印技术研究[J].计算机安全,2004,(5):41-42.