基于 C/S 的分布式远程访问服务安全管理

施嵘 郭宗桂 (上海交通大学网络中心 200030)

摘要:越来越多的远程用户通过 MODEM 拨号上网,同时也带来了更加严重的安全问题和管理问题。本文详细讨论了远程访问用户的安全管理、身份验证、访问授权和记费管理,并描述了一种在 CLIENT/SERVER 架构下的分布式安全管理系统,说明了它的协议及其实现方法,最后给出了实例。

关键词:CLIENT/SERVER 远程访问 网络管理 网络安全 身份验证 加密 记费管理

一、引言

随着信息产业的不断发展及 Internet 的迅速普及,越来越多的普通家庭及小型办公室将通过 MODEM 和公共电话线以拨号方式访问互连网络。提供远程访问服务的典型例子是 Internet 服务提供者(Internet Service Provider)和大学校园网。

在 ISP 或校园网的网络中心,配置了由几台或几十台 MODEM 组成的 MODEM 池(MODEM Pool),一端直接与公共电话线相连,另一端通过网络访问服务器(NAS, Network Access Server)或称通信服务器(Communications Server)与本地网络相连。NAS 是专门的硬件设备,如 IBM 的 8235 Dial In Server 和 Shiva 的 LanRover 和 ShivaPort。NAS 内部运行专门的软件,支持远地用户以各种方式(如 PPP、SLIP、Telnet、Rlogin 等)访问本地网。除了 IP 包,NAS 也可以在远程 PC 和本地 LAN 之间转发 IPX、NetBEUI 和 LLC 等数据包,从而支持访问 Netware、Windows NT 等。这样一旦一个远程用户通过拨号生成连接,电话线就会成为透明的,用户使用本地软件如 Netscape 可以访问所有网络资源,他们的计算机就象是直接和网络相连。形象地说,对用户来讲 MODEM 就象一块网卡,而 NAS 就象远程 PC 和本地 LAN 之间的网关。对 NAS 的一个要求是应能对数据包进行过滤和压缩(如采用 VJ TCP/IP Header Compression 或 IPX Header Compression)。

二、基于 C/S 的分布式安全系统模型

在大型主机年代,可以采用高度安全的机制来保护关键数据;过去完全独立运行的 PC 也基本不存在安全问题。随着网络计算的发展,安全性问题才日益变得突出,而远程访问由于其天然的开放性,对用户的身份验证和安全管理更加困难和复杂。

以前的远地用户管理信息分布在网络上每台 NAS 设备上,每台 NAS 保存相应用户的口令、访问权限等信息,负责进

行用户身份验证和访问授权控制。这种信息分散保存本身就是不安全的，而且也给管理和维护带来很大不便。理想情况下，所有关键数据均保存在一个中央数据库内，数据库本身位于安全的系统上，如 Unix 或 Windows NT。CLIENT/SERVER 架构为远程访问管理提供了较好的分布式解决方案。

在分布式远程访问安全管理系统中，NAS 作为远程访问验证服务(RAAS, Remote Access Authentication Service)的 Client，而 RAAS Server 实际上是一套运行于 Unix 或 Windows NT 之上的软件。Server 通过一个单一的中央用户数据库来应答 Client 的请求，即远程用户的验证和授权，数据库存放的信息包括用户信息、口令、访问权限表等。RAAS Server 也可以作为代理 Client，向其他 RAAS Server 或其他标准的验证 Server(如 Unix 或 Windwos NT Domain Server)发验证请求。而 NAS 既是远程 PC 的通信 Server，又是 RAAS 的 Client。采用这种机制，验证和授权与实际访问分开，而且数据集中保存，大大提高了安全性，而且随着需求的增加，可以方便地对用户进行扩充，给管理和维护带来了极大的灵活性和方便性。

三、分布式安全管理系统的协议和实现

1. 协议原理

RAAS 通过 Client 和 Server 之间的一系列通信进行用户验证，其底层建立在 RPC(Remote Procedure Call)基础之上。

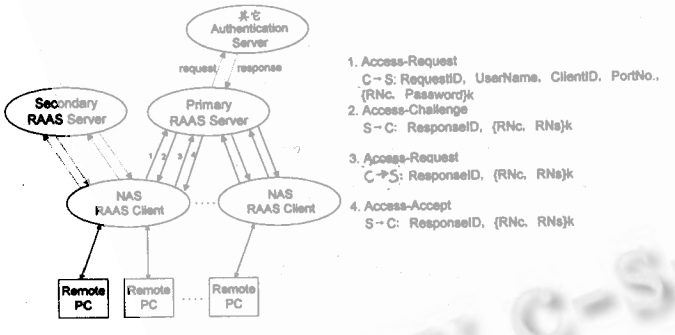

图 1　验证协议示意图

(1)C→S: RequestID, UserName, ClientID, PortNO., {RNc, Password}k

当远程用户通过 MODEM 拨通 NAS 端的 MODEM，NAS 要求用户输入用户名和口令，然后 NAS 按如下方法产生 Access - Request 包：为了与以前的请求信息区别开来，Client 首先产生一个随机数(RNc)，此随机数的用户口令(Password)一起用密钥 K(与 Server 共享)按某种加密算法，如 RSA 的 MD5 进行加密，然后将加密结果和 RequestID(用于标识一对请求和应答)、用户名、Client 标识、NAS 用于 MODEM 连接的端口标识一起打入 Request 数据包，传送给 Server。访问请求提交后，若经过一指定时间没有收到应答，请求重传若干次，若仍没有响应或主 Server 瘫痪，该请求可转发至备用 SERVER。

(2)S→C: ResponseID, {RNc, RNs}k

Server 接到请求后，可以向其他验证 Server 发请求，否则它首先确认该 Request 为合法 Client 所发，然后用共享密钥 K 解密数据，与已有数据库文件对照。此数据库可以是本机上的 RAAS 数据文件，也可以是其他标准验证 Server 的验证数据库，如 Unix 的 passwd 文件、DCE(分布式计算环境)的 Kerberos、Windows NT 的域帐户数据库等。查询后若某些条件不满足，Server 发 Accwss - Reject 应答说明该请求无效。若条件满足，为确保进一步安全，还要发 Access - Challenge 应答：Server 产生一个随机数(RNs)，要求用户按约定好的算法求出相应结果，然后将此 RNs 和解密后的 RNc 一起用共享密钥 K 加密，使用 RNc 是为了使 Client 确信此应答确实是 Server 应自己的请求所发。Accwss - Challenge 数据包中的 ResposeID 值与上面的 RequestID 相同，表明是一对请求和应答 Challenge 中还包括给用户的提示信息，如“请对 58792750 计算并输入结果：”。

(3) C → S: RequestID', UserName, ClientID, PortNO., {RNc', Result}k

Client 收到应答后，先解密确认确实为 Server 所发。若收到的是 Access - Reject，则提示用户该次访问请求被拒绝；若收到的是 Access - Challenge，则提示用户作出解答。合法用户可以利用某些特殊的设备和智能卡(Smart Card)或专用软件对 RNs 计算出 Result 提交给 NAS。NAS 产生一个新的随机数 RNc'用于标识新的会话，然后对 Result 和 RNc'用共享密钥 K 加密，打入第二个 Accwss - Request 数据包(新的 RequestID 值)传送至 Server。

(4)S→C: ResponseID', ServiceType, ConfigINFO, {RNc'}k

Server 收到新的 Request 后对 Result 解密，若与预期结果不同，则发 Access - Reject 应答，否则或者再发一个 Challenge，或者发 Access - Accept，其中包括加密的 RNc'、用户所希望的服务类型(如 PPP、SLIP、Login User 等，已在数据库文件中指明)及与该服务相关的配置信息(ConfigINFO)，如对 SLIP 和 PPP，应包括 IP 地址、子网掩码、最大传输单元(MTU)、压缩方式和包过滤标识(限制用户访问某些特定网络资源)等。协议实现时，这些信息和 RNc'一起用共享密钥加密。

经过这四次信息传递，远程用户或者被拒绝访问，或者通过 NAS 和 RAAS Server 获得相应的访问服务。

2. 设计和实现

(1)传输协议的选择。TCP/IP 的传输层协议有 TCP 和 UDP，基于以下考虑，本系统的实现选用 UDP 协议。首先，若在请求验证时主 Server 失败，该请求应转发到备用 Server，这就要求请求的副本应保存在传输层以上以备重传，这也意味着重

© 中国科学院软件研究所 http://www.c-s-a.org.cn

传定时器仍是必要的，但这里的定时要求与 TCP 所能提供的有很大差别。一方面，RAAS 根本不需要检测数据丢失与否，用户一般不在乎等待几秒钟以待验证完成，因此 TCP 的重传及确认帧也就没有必要；另一方面，用户又不可能为验证等待几分钟或更长时间，为达到可靠传送数据而有可能耗时几分钟的 TCP 协议同样没有必要，这时用户可以转到备用 Server 请求验证。其次，一般一次验证过程只有几次报文交换，假如为此建立连接和拆除连接，开销是相当大的，而使用简单高效的 UDP 协议就比较合适了。另外，当前的 Server 一般都是多线程的，对同时到来的许多请求，可以对每一请求生成一个线程进行处理，每一线程可以使用简单的 UDP 数据包直接与 NAS 进行通信，因此 UDP 也使 Server 的实现变得简单。

(2)包格式。每个 RAAS 数据包被封装在一个 UDP 数据报的数据区内，RAAS 数据包的格式如下：

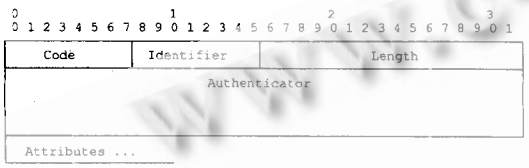

图 2

Code 域是一个八位串，标识 RAAS 包的类型，包括：Access - Request, Access - Accept, Access - Reject, Access - Challenge, Accounting - Request, Accounting - Response。其中后两种用于记费管理。

Identifier 域是一个八位串，用于请求和应答相配，即一个 RequestID 对应一个 ResponseID。

Length 域是两上八位串，指明整个数据包的长度，从左向右起超过该长度的数据将被丢弃。

Authenticator 域包括十六八位串，用于验证 Server 应答的有效性以及口令加密算法。在 Access - Request 包中，Authenticator 是一个由十六个八位串组成的随机数，此随机数和共享密钥按照 RSA 的 MD5 算法求得加密结果，此结果再与用户口令进行异或操作，最后所得结果作为用户口令属性的值存放于用户口令属性域。而在 Access - Response 包(包括 Access - Accept, Access - Reject, Access - Challenge)中 Authenticator 是按以下方法算出的加密值：将 Code、Identifier 和 Length 域的八位串以及 Access - Request 包中的 Authenticator、Response Attribute、共享密钥 K 一起作为输入按照 MD 5 算法求得结果。

不同的数据包有不同的属性，其长度也不同。Attribute 域指明属性的类型、长度和值。如在 Accwss - Request 包中有属性 User - Name，它的值就是用户名的字符串。又如 Access - Accept 包中有属性 Service - Type，其 Value = PPP(或 SLIP 等等)。根据需要可以定义几十个属性，主要包括：User - Name, User - Password, NAS - IP - Address, NAS - Port, Service - Type, Framed - IP - Address, Framed - IP - Network, Filter - ID, Framed - Mtu, Framed - Compression, Callback - Number 等等，至于具体的每一属性及其可取值，这里就不详细讨论了。

(3)进一步的安全性。为确保安全，Client 和 Server 的共享密钥 K 应足够长，并且不能通过网络传输。不同的 Client 密钥也应该不同，Server 可以通过 UDP 数据包的源 IP 地址决定采用哪个密钥，这样验证请求也可以使用代理(Proxy)方式，即可以验证不是直接来自 NAS 的验证请求。

为确保对验证请求的答复不是非法黑客(hacker)所发，RAAS Server 可以向 Client 签发数字签名，以证实自己的身份，收到此信息后 NAS 才作出必要的配置，为用户提供相应的网络服务。

(4)记费管理。记费管理是网络管理的一个重要组成部分，尤其对 ISP 而言。RAAS 的记费管理也基于 CLIENT/SERVER 架构，Client 仍为 NAS，其作用是向指定的 Server 提供记费信息，Accounting Server 接收记费请求并向 Client 发确认信息。Accounting Server 可以运行在与 RAAS Server 不同的主机上，但为了方便起见，它可以作为 RAAS Server 的一个子进程运行于同一主机上。这样两个 daemon 的 UDP 端口号应该是不同的，如 RAAS Server 倾听的 UDP Port 为 1645，而 Accounting Server 倾听的 UDP Port 为 1646。Accounting Server 同样可以作为 Client 向其他 Server 发出请求。

记费的工作过程如下：一个 CLIENT 被配置成记帐后，每当它为一个远程用户提供服务，它就发一个 Accounting - Start 包给 Server，其中包括服务类型、用户信息等。Server 收到后向 Client 发确认。当服务结束，即用户拆除 MODEM 连接，Client 向 Server 发 Accounting - Stop 包，其中包括服务类型和一些统计信息，如服务持续时间等，Server 收到后再向 Client 发确认。

Accounting 包的格式与上面相同。其 Code 有 Accounting - Request 和 Accounting - Response 两种。因为没有用户口令需要加密，Request Authenticator 中放的是以 Code、Identifier、Length、16 个全"0"八位串、Request Attributes 和共享密钥为输入进行加密所得的结果，而 Response Authenticator 算法其相同，只是 16 个全"0"八位串换成 Request Authenticator 中的值。主要的属性有记帐状态(只有 Start 和 Stop 两个值)、会话 ID(使某一会话的 Start、Stop 记录相配)、会话时间等。

记帐的输出是一个文本文件，存放用户信息及其访问时间，作为记费的依据。若用户很多，也可通过接口调用标准数据库如 SQL 的例程进行管理，从而使维护更加方便。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

四、实例

1.验证

(1)一个用户名为 PPPUser1 的远程用户拨通地址为 202.100.100.100 的 NAS 上的端口 7,NAS 和 RAAS Server 发 Access－Request 包。

Code＝1(Access－Request)

ID＝2

Request Authenticator＝{16 个八位串随机数}

Attributes:

User－Name＝"PPPUser1"

User－Password＝{MD5(共享密钥|Request Authenticator)与 Password 异或}

NAS－IP－Address＝202.100.100.100

NAS－Port＝7

(2) Server 应答,向 Client 发 Access－Challenge 数据包。

Code＝11(Access－Challenge)

ID＝2(与 Access－Reuqest 相同)

Response Authenticator＝{MD5(Code(11)|ID(2)|Request Authenticator|

本应答的 Attributes|共享密钥)}

Attributes:

Reply－Message＝"请对 58792750 计算并输入结果:"

(3)用户输入解答,NAS 生成新的 Access－Request,将解答传给 Server。

Code＝1 (Access－Request)

ID＝3(新的 ID)

Request Authenticator＝{新的随机数}

Attributes:

User－Password＝{MD5(Request Authenticator|共享密钥)与用户输入的 Result 异或}

NAS－IP－Address＝202.100.100.100

NAS－Port＝7

(4)用户的 Result 不正确,Server 通知 NAS 拒绝其访问请求。

Code＝3(Access－Reject)

ID＝3 (与 Access－Request 的相同)

Response Authenticator＝{MD5(Code(3)|ID(3)|上面的 Request Authenticator|

本应答的 Attributes|共享密钥)}

Attributes:(空,或者是给用户的提示信息)

2.记费

以下为记费输出的实例。

Wed Oct 5 22:00:55 1996

Acc－Session－ID＝"06000003"

User－Name＝"PPPUser1"

Client－ID＝202.100.100.100

Client－Port－ID＝7

Acct-Status-Type＝Start

User－Service－Type＝Framed－User

Framed－Protocol－PPP

Wed Oct 5 23:15:31 1996

Acc－Session－ID＝"06000003"

User－Name＝"PPPUser1"

Client－ID＝202.100.100.100

Client－Port－ID＝7

Acct－Session－Time＝4480

User－Service－Type＝Framed－User

Framed－Protocol＝PPP

五、结论

随着 Internet 在我国的迅速普及和发展,今后将有越来越多的远程用户通过 MODEM 访问入网。采用基于 CLIENT/SERVER 的分布式安全管理和记费管理系统,既简化了管理,也极大地提高了网络安全性,必将为网络计算的发展起到进一步的推动作用。

参考资料

[1] Shiuh－Pyng Shieh, and Wen－her Yang, "An Authentication and Key Distribution System for Open Network System", ACM Operating Systems Review, pp 32－41, April 1996.

[2] Postel, J., "User Datagram Protocol", Internet RFC 768, August 1980.

[3] Rivest, R., and S. Dusse, "The MD5 Message－Digest Algorithm", Internet RFC 1321, April 1992.

[4] Kaufman, C., Perlman, R., and Speciner, M., "Network Security: Private Communications in a Public World", Prentice Hall, March 1995.

(来稿时间:1996 年 11 月)

© 中国科学院软件研究所 http://www.c-s-a.org.cn