维普资讯 http://www.cqvip.com

一个基于 PAM 的身份认证系统的设计与实现

李大治　卿斯汉

（中国科学院软件研究所信息安全技术工程研究中心　北京100080）

摘　要　当今网络安全越来越受到人们的重视，身份认证作为网络安全的重要组成部分，已成为保证应用系统安全的有力手段。当前的一些身份认证产品的缺点是认证方法单一，缺少扩展性，很难满足不同应用系统的需要。本文提出了一个基于 Linux PAM(Pluggable Authentication Modules)的身份认证系统，可支持多种认证方法，而且具有良好的扩展性。本文首先介绍了 PAM，讨论了基于 PAM 开发的关键问题，然后给出了系统的设计，最后结合具体实例阐明了系统的工作原理。

关键词　网络安全，身份认证，PAM

Design and Realization of an Authentication System Based on PAM

LI Da-Zhi　QING Si-Han

(Engineering Research Center for Information Security Technology, Institute of Software, Chinese Academy of Sciences, Beijing 100080)

Abstract　Today people pay more attention to network security. Authentication has become a strong measure to guarantee application security. There are some authentication products at present, most of which have few authentication measures and poor scalability, so it is hard to meet various requirements. An authentication system based on Linux PAM (Pluggable Authentication Modules) is presented in this paper. It can support more authentication measures and has good scalability. In this article, first, PAM is introduced and key issues of using PAM are presented. Second, the authentication system is designed. Finally, an authentication example is given to illustrate the operation principle of the system.

Keywords　Network security, Authentication, PAM

1　引言

当今网络安全越来越受到人们的重视，身份认证是网络安全的重要组成部分[1]，认证方法从过去简单的基于用户名/密码的认证发展到今天的强身份认证，如：Kerberos 认证、X509认证等等[1]。但是传统应用系统的认证方法都是硬编码于其中，不更改代码就无法使用新的认证方法。身份认证系统的出现解决了这个问题，身份认证系统主要包括认证服务器和编程接口，认证服务器提供身份认证服务，应用程序开发者通过编程接口就可以利用认证服务器来实现认证，使得具体的认证代码同应用系统相分离。

当前市场上有一些身份认证产品，从技术说明来看，都可以提供强认证服务，但其缺点是认证方法单一，而且系统封闭，不具有可扩展性。本文提出了一个基于 Linux PAM (Pluggable Authentication Modules)的身份认证系统，可支持多种认证方法，借助 PAM 的认证模块开发框架，开发者可以实现自己的认证协议，另外提供了一套 ActiveX 控件作为应用程序接口，使得接口具有 Windows 平台下的语言独立性，多种编程语言都可以使用。

2　PAM 介绍

PAM(Pluggable Authentication Modules)是用来认证用户的一个通用框架，实现为类 Unix 系统下的一套共享库，本文所提到的 PAM 主要是 Linux 平台下的 PAM[2~6]。PAM 机制采用模块化设计，具有插件功能，使得我们可以轻易地在应用程序中插入新的认证模块或替换原有的模块，而不必对应用程序做任何修改，因为认证机制与应用程序之间相对独立。应用程序可以通过 PAM API 方便地使用 PAM 提供的各种认证功能，而不必了解太多的底层细节。

PAM 的结构如图1。每个应用程序有一个配置文件，包含了该应用要使用的认证模块，认证模块可以是一个也可以是多个。应用程序调用 PAM API 就可以实现认证，认证过程中 PAM 会读取配置文件，依次调用配置文件中的认证模块进行认证。

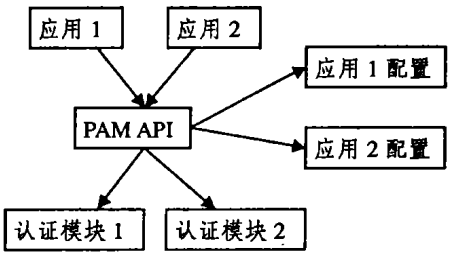

图1　PAM 结构

3　关键问题

PAM 只是一套共享库，为了实现一个完整的身份认证系统，要对 PAM 进行如下的改进和扩充：

(1) 为了使 PAM 可对网络中不同主机上的应用程序提

李大治　硕士生，研究方向为网络安全；卿斯汉　研究员，博导，研究方向为信息安全理论与技术。

维普资讯 http://www.cqvip.com

供认证服务，必须设计一个认证服务进程，它可以接收应用程序的认证请求，然后将认证请求转化成对 PAM 的调用；

(2) 认证通常是由认证端和被认证端通过认证协议来完成认证的过程，PAM 的认证模块在逻辑上属于认证端，在被认证端 PAM 没有实现认证组件，为了使 PAM 可支持更多的认证协议，本文实现了 PAM 的被认证端模块[7]；

(3) C/S 或者 B/S 模式的应用系统，客户端与服务端在逻辑上分离，当客户端通过身份认证之后，必须有一种安全机制使服务端能验证客户端的认证结果。本文设计了一个验证接口和一个称作"访问许可证"的数据结构。"访问许可证"主要包含了客户标识，客户端地址，要访问的服务名称，认证服务器的签名等信息。客户端通过认证之后从认证服务器可以获得一个"访问许可证"，使用验证接口就可以获得客户端的认证结果。

4 身份认证系统的设计

本文提出的身份认证系统的全称为 PAM Based Authentication System，简称 PBAS。系统分成3个部分：认证服务器，认证端组件以及验证端组件。认证服务器包括认证服务进程和 PAM；认证端组件包括认证接口、认证模块客户端、对话接口；验证端组件包括验证接口。图2显示了 PBAS 在一个 C/S 模式的应用系统中的应用，虚线框的内容为系统要实现的部分。

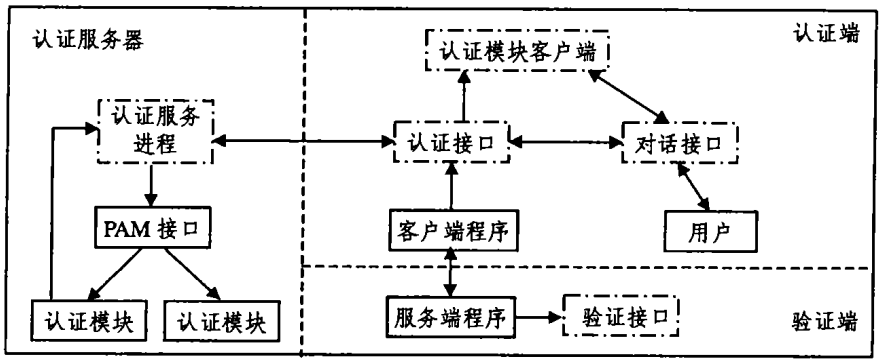

图2 PBAS 系统在 C/S 模式应用系统中的应用

4.1 认证服务进程

认证服务进程使用 C 语言在 Linux 平台下实现，关键模块有：

·命令解析调用模块：将认证接口发过来的各种命令转化成对 PAM 接口函数的调用；

·数据管道模块：实现了一个 PAM 的 conversation 回调函数[5]，将 PAM 认证模块的输出传递给客户端的认证接口，并且将客户端的输入回传给 PAM 认证模块；

·访问许可证处理模块：将认证结果、其它一些关键信息以及签名组成一个称为访问许可证的数据结构。

4.2 认证端组件

认证端组件实现为 Windows 平台下的一套 ActiveX 控件，可供 Windows 平台下支持 ActiveX 技术的各种编程语言调用。认证端组件的 UML 类图见图3，由于篇幅有限，省略了部分函数和参数。

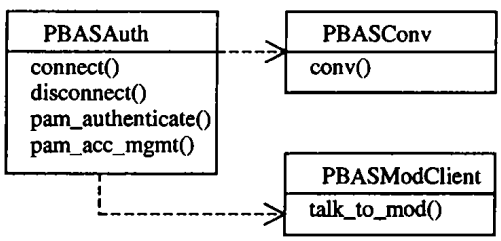

图3 认证端组件的 UML 类图

PBASAuth 类为认证接口，通过调用该类可实现认证功能。PBASConv 类为对话接口，在认证过程中需要输入输出信息时，PBASAuth 会调用该类的 conv 方法，开发者通过实现自己的 PBASConv 类就可以定制界面。PBASModClient 为 PAM 认证模块客户端，当 PAM 认证模块在认证过程中需要客户端计算时，就会通过预先定以的消息通知 PBASAuth 类，PBASAuth 会调用 PBASModClient 的 talk_to_mod 函数进行处理。

4.3 验证接口与访问许可证

访问许可证主要是为 C/S 和 B/S 模式的应用系统设计的，这样的应用系统认证是在客户端完成的，如何让服务端的程序知道认证的结果是个关键问题。本文设计了一个访问许可证的数据结构，当客户端程序认证之后，就会从认证服务器获得一个访问许可证，服务端程序可以通过验证接口获得客户端的认证情况。访问许可证的结构如下：

LICENSE ={Type,INFORM,$sign_{as}$(md5(INFORM))}

INFORM = (UserID,ClientIP,ServerIP,ServiceName, AuthTime,ExpireTime)

Type：访问许可证类型，目前支持两种类型，0x01 表示认证服务器证实了用户的身份；0x02 表示认证服务器证实了用户具有访问系统的权利；UserID：登录用户的唯一标识；

ClientIP：应用系统客户端的 IP 地址；ServerIP：应用系统服务端的 IP 地址；ServiceName：服务名称，认证服务器同时为多个应用系统提供认证服务，用服务名称来唯一标识应用系统；AuthTime：认证结束的时间；ExpireTime：访问许可证的过期时间(每个系统的许可证有效期可以在认证服务器中配置)。signas(md5(INFORM))表示用认证服务器的私钥对 INFORM 的 MD5值进行的签名。

验证接口除了可以提取访问许可证中的信息外，最重要的就是验证访问许可证，已决定客户端是否有权限访问。验证的伪代码如下：

```
BOOL verify() {
  if ( signas(md5(INFORM)) 无效 )
      return FALSE;
  if ( ClientIP !=
      当前请求访问的客户端地址 )
  return FALSE;
  if ( ServerIP != 本机 IP 地址 )
      return FALSE;
```

维普资讯 http://www.cqvip.com

```
    if ( ServiceName != 
        自己在认证服务器中定义的服务名称 )
        return FALSE;
    if ( ExpireTime <= 当前时间 )
        return FALSE;
        return TRUE ;
}
```

4.4 系统工作原理

PBASAuth 中有两个方法最为重要:pam_authenticate 用来验证用户的身份,成功后返回0x01类型的访问许可证;pam_acct_mgmt 用来验证用户是否有权利访问这个系统,成功后返回0x02类型的访问许可证。通过这两个函数的执行流程,就可以理解整个系统的工作原理。以下通过一个简单示例说明 pam_authenticate 和 pam_acct_mgmt 的执行流程。

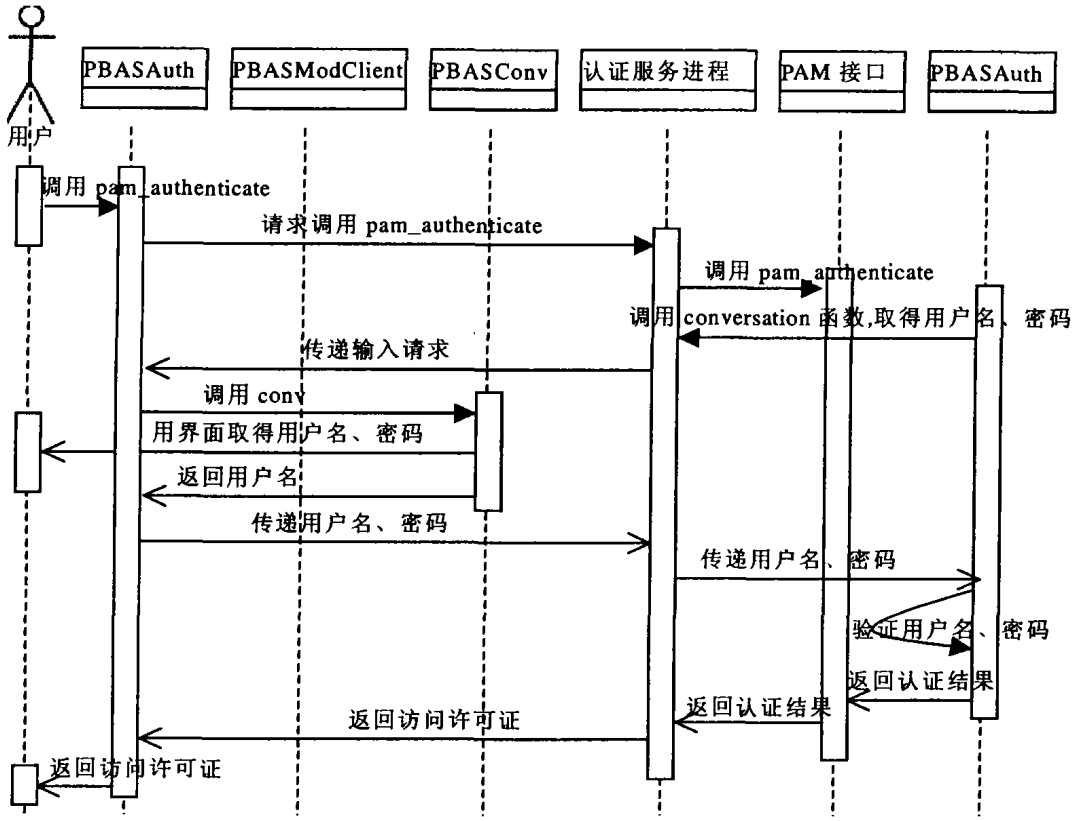

图4 PBASAuth 的 pam_authenticate 方法的 UML 顺序图

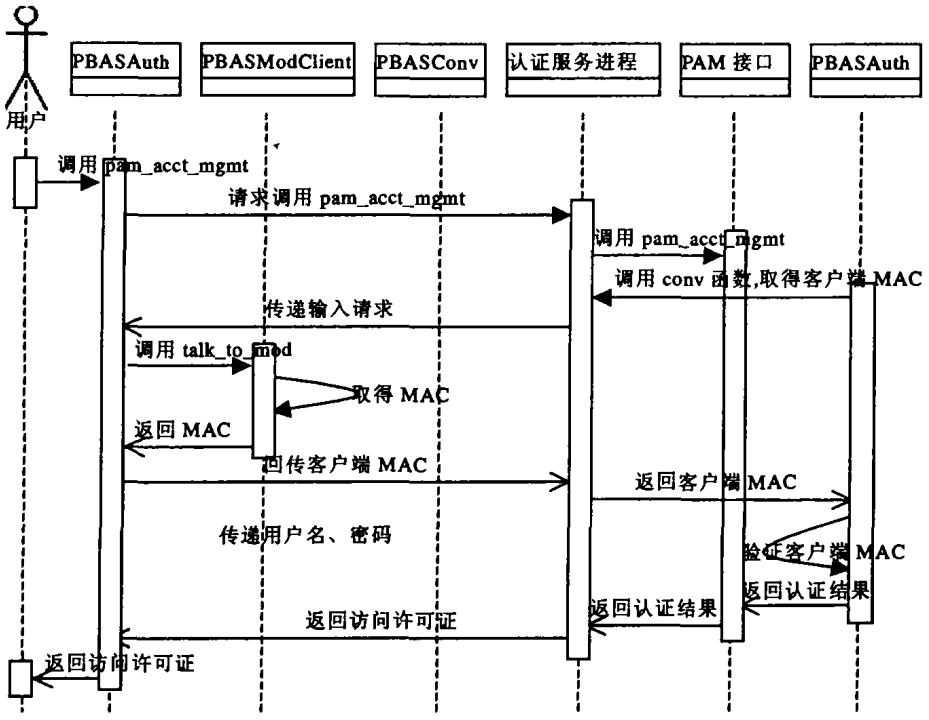

图5 PBASAuth 的 pam_acct_mgmt 方法的 UML 顺序图

一个系统采用简单的用户名、密码方式来验证用户的身份,而且用户只有在预先指定的机器上才有权访问系统,也就是用户名要与登录机器的 MAC 地址绑定。

(下转第79页)

维普资讯 http://www.cqvip.com

布的标题为"goingOffLine"的通知被发送给通知使用者B。过了一段时间，通知发布者D也选择"goingOffLine"这个标题来发布通知，因为这个标题符合预定者A和通知使用者之间的预定，所以，D发布的标题"goingOffLine"的通知也被发送给通知使用者B。图4说明了代理模式下发布通知过程[3]。

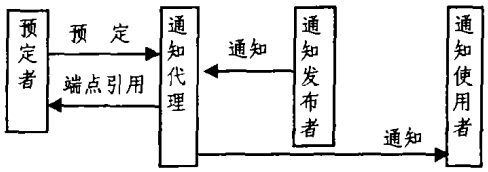

图4 代理方式发布通知过程

在有代理的情况下，通知代理起到一个中间桥梁的作用，代替了直接通知模式下的通知产生者，使用通知代理的目的是用来代替和最终通知使用者的互操作。首先，预定者发送标题请求信息给通知代理，通知代理创建一个预定资源，并返回这个预定资源端点引用给预定者。通知发布者向通知代理请求发布，然后，通知代理传递通知信息给任何符合这个预定的通知使用者。交换通知信息有两个作用：发布通知信息给代理和发布通知信息给通知使用者。

3 WSRF规范和Web服务通知规范(WS-Notification)

对有状态资源操作、管理在WSRF子规范中都有详细的说明，WSRF规范包括五个子技术规范[2]。Web服务资源生命周期(WS-ResourceLifetime)定义了WS-Resource消除管理机制，服务请求者可以立即消除或定时消除WS-Resource；Web服务资源属性(WS-Resource Properties)用XML文档定义一个WS-Resource，提供了管理WS-Resource属性的机制；Web服务更新引用(WS-RenewableReferences)说明了WS-Resource如何通过某种通知策略重新找回失效的端点；Web服务服务组(WS-ServiceGroup)定义了描述、管理不同种类的Web服务的方法；Web服务的基础错误(WS-Base Faults)定义了在Web服务信息交换发生错误时，可使用的XML错误类型。除此之外，还发布了Web服务通知(WS-Notification)规范，该规范定义了一种用发布/预定模式去预定和通知事件的方法。Web服务资源生命周期、Web服务资源属性、Web服务通知这3个规范的草稿已于2004年1月发布，不久，其他的3个技术规范也将发布。

结论 WSRF和WS-Notification提出的目的是为商业应用、网格资源和系统管理提供公共的、标准的基础设施。WSRF引入有状态资源的概念，规定通过对一个由Web服务地址和资源标识组成的端点来引用来访问Web资源，该规范提出一种用WS-Resource方式来创建、管理有状态资源方法；通过对Web服务属性文档操作来对状态进行检查和设置；通过信息交换来管理有状态资源的生命周期。WS-Notification提出了基于Web服务有状态资源间通信方式。变化和事件能够以一种标准的方式发布，然后再直接或通过代理将通知发送出去。WSRF仅仅是一个开始，未来发展趋势是建立在WSRF上的Web服务分布式管理(WS Distributed Management，简称WSDM)框架。

参考文献

1 Foster I. What is the Grid? A Three Point Checklist. July 2002. http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf
2 The Web Services Resource Framework. http://www-106.ibm.com/developerworks/library/WS-Resource/ws-wsrf.pdf
3 WS-Notification. http://www-106.ibm.com/developerworks/library/ws-pubsub/WS-PubSub.pdf
4 Foster I, Kesselman C, Nick J, Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Jan. 2002. http://www.Globus.org/research/papers/ogsa.pdf
5 Open Grid Services Infrastructure (OGSI) V1.0 http://XML.coverpages.org/OGSI-SpecificationV110.pdf
6 Czajkowski K, Ferguson D, Foster I, Frey J, Graham S, Snelling D, Tuecke S. OGSI-Refactor. From Open Grid Services Infrastructure to Web Services Resource Framework: Refactoring and Evolution, 2004. http://xml.coverpages.org/OGSI-to-WSRFv11-20040305.pdf
7 都志辉，陈渝，刘鹏. 网格计算. 北京：清华大学出版社，2002
8 Modeling Sateful Resource in Web Services. http://www-106.ibm.com/developerworks/library/WS-Resource/ws-modelingresources.pdf
9 WS-ResourceLifetime. http://www-106.ibm.com/developerworks/library/wsresource/WS-Resourcelifetime.pdf
10 WS-ResourceProperties. http://www-106.ibm.com/developerworks/library/WS-Resource/WS-Resourceproperties.pdf

(上接第65页)

pam_authenticate的执行顺序图见图4。(1)首先，向认证服务进程发出调用PAM的pam_authenticate的申请，即而，认证服务进程调用pam_authenticate，pam_authenticate会调用具体的认证模块完成认证；(2)认证服务进程的conversation函数会在认证模块与PBASAuth之间传递认证数据，将认证模块的提示信息传给PBASAuth，从客户端取得用户名、密码回传给认证模块，客户端的输入、输出功能是由PBASConv完成的；(3)如果认证成功，认证服务进程最后组成访问许可证返回给客户端。图5为pam_authenticate的顺序图，与pam_authenticate的唯一不同是，pam_acct_mgmt执行过程中不需要用户参与，在客户端是由PBASModClient的talk_to_mod方法取得客户端MAC，不需要调用PBASConv。

结论 PAM是Linux下一个非常成熟的认证框架，本文提出的身份认证系统基于PAM开发，不仅继承了PAM的灵活性、可扩展性、稳定性，更可以利用现有的大量PAM认证模块，以减少开发认证协议的复杂度，节约成本。另外，PAM源代码的开放性，更使我们进一步的深入开发成为可能。

参考文献

1 冯登国. 计算机通信网络安全. 清华大学出版社，2001
2 韩波. 深入Linux PAM体系结构. http://www-900.ibm.com/developerWorks/cn/linux/l-pam/index.shtml，2002
3 Samar V, Schemers R. Unified Login With PluggableAuthentication Modules (PAM). RFC 86.0, Oct. 1995
4 Morgan A G. The Linux-PAM System Administrator's Guide. http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html，2002
5 Morgan A G. The Linux-PAM System Administrator's Guide. http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam_appl.html，2002
6 Morgan A G. The Linux-PAM System Administrator's Guide. http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam_modules.html，2002
7 Morgan A G. Pluggable Authentication Modules (PAM). http://www.kernel.org/pub/linux/libs/pam/pre/doc/current-draft.txt，Dec. 2001