轻量目录访问协议分析

李重武 (中国科学技术大学研究生院 100039)
倪惜珍 (中科院信息安全技术工程研究中心 100080)

摘要：本文详细描述了轻量目录访问协议，包括协议的基本元素、协议模型的分析以及协议操作中处理的各种数据格式。

关键词：目录服务 LDAP 认证中心

1 概述

随着网络信息数量的急剧增长，目录服务作为一种管理网络资源的重要手段日益为人们所重视，ITU-TX.500规范系列中规定了有关目录服务的标准。希望通过分布式目录服务系统对物理上分散存储于网络上的信息提供一种逻辑上统一的管理和访问服务。但X.500目录服务是一个高度复杂的信息存储机制，包括客户机－目录服务器访问协议、服务器－服务器通信协议、服务器链对查询的响应、复杂搜索的过滤等功能。由于协议过于复杂，实现十分困难，而且X.500 DAP需要运行于完全的OSI协议栈上，无法适应TCP/IP的工业标准。因此轻量目录访问协议(LDAP：Lightweight Directory Access Protocol)应运而生。轻量目录访问协议的目标是用较小的代价实现X.500的大多数功能，所以LDAP是目录访问协议中那些易于描述、执行的功能的子集。

2 模型

2.1 协议模型

LDAP协议采用的通用协议模型是一个由客户端发起操作的，客户机－服务器响应模型。在此协议模型中客户机传送给服务器一个协议操作请求，而服务器负责代替客户机在目录上执行操作，并将结果或错误信息返回给客户机。当客户机不再需要与服务器通信时，由客户机断开连接。

在协议版本1和2中，服务器需追踪全部参考节点和执行协议操作，将最终结果返回给客户机，但为了提高效率，在版本3中允许服务器将指向其他服务器的参考指针返回给客户机，而由客户机自己查找。这样做虽然提高了客户端软件的复杂成度，但可以有效分担服务器的负载，提高了服务器的效率和分布式应用能力。

2.2 数据模型

LDAP目录中的数据是按照树形结构组织的，被称为目录信息树(DIT：Directory Information Tree)，其结构如图1所示：

目录信息树的根节点(root)是一个没有实际意义的虚根，树上的节点被称为入口(Entry)，用于存储数据，其相当于关系数据库中表的记录，入口的名称由一个或多个属性组成，称为相对辨识名(RDN：Relative Distinguished Name)，用于区别和它同级别的入口。从特定入口到根的直接下级入口的相关辨识名序列组成了该特定入口的辨识名(DN：Distinguished Name)，用来唯一标识该入口，例如：Uid=philip, Ou=person, Ou=tsinghua, O=edu, C=cn

入口由属性组成，属性由属性类型和与该类型相关的若干个值构成。属性类型是由短的描述名和OID定义的，它决定属性值的类型，包含了属性取值应遵循的文法和对属性查询时应用的匹配规则(Matching Rule)，例如：描述人的属性可能包括姓名(Uid)，电话号码(TL)和邮件地址(E-mail)等。LDAP中关于属性类型定义、对象类定义和其他匹配规则等的规定构成所谓模式(Schema)。服务器使用模式中的信息来决定如何操纵一个入口。子模式(Sub-schema)则用来管理与目录

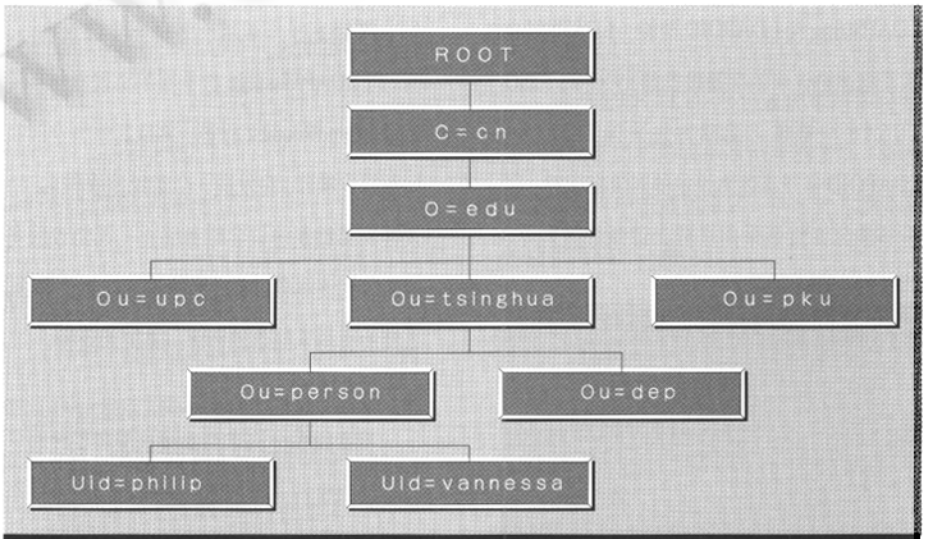

图1 目录信息树示例

模式相关的信息，如该目录服务器所支持的属性和对象种类。

3 协议基本元素

3.1 消息封装

协议的公共要素主要包括LDAPMessage封套协议数据单元(PDU：Protocol Data Unit)的格式，和用于协议操作的数据类型的定义。为了协议交换的目的，所有协议的操作数据都被封装在一个名为LDAPMessage的通用数据结构中，数据结构定义如下：

LDAPMessage ::= SEQUENCE {

messageID MessageID,

protocolOp CHOICE { ...},

controls [0] Controls OPTIONAL }

MessageID：标识一次会话中不同消息的ID号。

CHOICE：全部协议操作的可选项列表。

Controls OPTIONAL：控制信息扩展选项

LDAPMessage的功能是提供一个包含所有协议交换公共字段的封套。现阶段仅有的公共字段是message ID和controls。

3.2 字符串类型

LDAP协议中字符串类型包括LDAPString和LDAPOID两种。LDAPString用于描述协议中的字符串类型，它使用了ISO 10646字符集，编码方法采用UTF-8。LDAPOID用于描述协议中的对象标识符，字符串的允许值是采用点分十进制数值表示，并经过UTF-8编码。

3.3 结果数据报

服务器使用LDAPResult数据结构向客户机返回协议操作成功或失败的指示。其数据报格式如下：

LDAPResult ::= SEQUENCE {

resultCode 结果代码列举{... },

matchedDN 匹配的辨识名，

errorMessage 返回的错误信息文本，

referral 指向其他服务器的参考指针 }

4 协议操作

4.1 绑定和解绑定操作

绑定操作的功能是允许在客户机和服务器之间交换身份认证信息。使客户能以服务器上一个合法对象的身份对目录进行操作。

绑定请求：绑定请求是一个请求队列，绑定请求的定义为：

BindRequest ::= [APPLICATION 0] SEQUENCE

{

version 版本号整型数 (1 .. 127),

name LDAP辨识名，

authentication 认证选择 }

绑定响应：

BindResponse ::= [APPLICATION 1] SEQUENCE

{

LDAPResult 请求结果，

serverSaslCreds 服务器的SASL证书

}

解绑定操作被用于终止一个协议会话。当一个LDAPMessage数据报的协议操作字段使用unbindRequest标签，并且其值为空时执行解绑定操作。在发送一个解绑定操作后，协议客户端假定协议会晤被终止，而协议服务器端收到该消息后假定客户端已经终止会话，丢弃所有尚未处理的请求，并关闭连接。

4.2 搜索操作

搜索操作允许客户端请求服务器代替它在目录中执行一个搜索。它可以用于从一个单独入口、从紧挨着特殊入口的下一层其他入口或整个子树中读取属性。

SearchRequest ::= [APPLICATION 3] SEQUENCE {

baseObject LDAP辨识名，

scope 搜索范围列举{... },

sizeLimi 整型数 (0 .. maxInt),

timeLimit 整型数(0 .. maxInt),

typesOnly 布尔值，

filter 满足的过滤条件，

attributes 属性描述列表 }

搜索范围 (scope)：搜索被执行的范围指针。

大小限制 (sizeLimit)：限制了搜索结果中返回的最大入口数。

只含类型 (typeOnly)：指示搜索结果是包含属性类型和属性值，或只含属性类型。

查询结果是通过搜索响应 (Search Responses) 消息返回的，搜索响应是一些LDAP Message数据报，每个数据报中至少含有SearchResultEntry, SearchResultReference, ExtendedResponse、SearchResultDone中的数据类型之一。

例：假设服务器hosta中有入口"O=MNN, C=WW"和"CN=Manager, O=MNN,C=WW"并且知道服务器hostb中存有"OU=People, O=MNN,C=WW"，服务器hostc中保存有子树"OU=Roles, O=MNN,C=WW"。如果有客户机要求查询"O=MNN,C=WW"，则从服务器返回的查询结果如下：

SearchResultEntry for O=MNN, C=WW

SearchResultEntry for CN=Manager,O=MNN, C=WW

SearchResultReference { ldap://hostb/OU=People,O=MNN,C=WW}

SearchResultReference { ldap://hostc/OU=Roles, O=MNN,C=WW }

SearchResultDone (success)

如果服务器中没有所要查询的目标，则返回一个参考节点。例：如果客户机向服务器hosta查询子树"O=XYZ,C=US"，返回结果如下：SearchResultDone (referral){ldap://hostg/ }

4.3 修改操作

修改操作允许客户机请求服务器代替自己在目录中对入口进行修改。修改请求的定义为：

ModifyRequest ::= [APPLICATION 6] SEQUENCE {
object 被修改的对象辨识名，
modification SEQUENCE OF SEQUENCE {
operation 操作的列举 {添加、删除、替换}，
modification 属性的类型和值的列举 } }

4.4 增加操作

增加操作允许客户机请求服务器代替自己在目录中增加一个入口。增加请求的定义为：

AddRequest ::= [APPLICATION 8] SEQUENCE {
entry 被增加入口的辨识名，
attributes AttributeList }

AttributeList：由被增加入口的属性列表构成，列表中必须包括可辨识值（用于形成入口自己的相对辨识名），对象类属性和列表对象类的任何强制属性值。

4.5 删除操作

删除操作允许客户机请求服务器代替自己在目录中删除一个入口。删除请求的定义为：

DelRequest ::= [APPLICATION 10] 被删除入口的LDAP辨识名

服务器在解析将被删除的目标入口时仍会对别名进行参考，并且只有处于叶子节点位置的入口才能被删除。服务器用"删除响应"对客户机的"删除请求"的执行结果进行回答，其定义如下：DelResponse ::= [APPLICATION 11] LDAPResult。

4.6 修改辨识名操作

修改辨识名的操作允许客户端改变目录中入口名最最左端部分，或者将目录中的入口子树移到一个新的位置，修改辨识名请求定义如下：

ModifyDNRequest ::= [APPLICATION 12] SEQUENCE {
entry LDAP辨识名，
newrdn LDAP相对辨识名，
deleteoldrdn 为布尔值，
newSuperior [0] LDAPDN OPTIONAL }

Entry：将被修改的入口的辨识名。

newrdn::新RDN名用来构成入口新名字的最左部分。

Deleteoldrdn：用于控制旧的RDN属性值是否仍然保存在入口中，或被删除。

NewSuperior：表示已存在入口的直接上级入口的辨识名。

4.7 对比操作

对比操作允许客户机请求服务器代替自己对某指定的入口和属性值在目录的入口中进行对比操作。对比请求的定义为：

CompareRequest ::= [APPLICATION 14] SEQUENCE {
entry 对比的辨识名，
ava 对比的属性值断言 }

5 协议特点分析

(1) 为了网络信息传递的准确性和非奇异性，LDAP使用ANS.1语法规则对交换中的协议要素进行描述，并使用BER规则进行编码。

(2) 基于LDAP的目录服务运行于TCP/IP协议栈上，不需要完整的OSI协议栈的支持。

(3) 采用Client/Server模型，由服务器和客户机两部分组成。LDAP服务器根据客户的请求查询、处理和更新LDAP目录。

(4) 虽然LDAP目录是树形组织结构，但并没有对实际用来存储数据的后台数据库类型做出具体规定，可根据应用需要来选择。

(5) 从面向公共服务的角度来看，由于目录服务器是专门为那些检索服务多余更新服务的应用设计的，所以其检索功能强大，而增删改等数据库功能相对较弱。因为不支持数据库管理系统常用的事务处理，免去了数据完整性约束，所以大大简化了数据操作。

(6) X.509标准最早是作为目录服务标准X.500的一部分，从数据结构来看，X.509证书天生适合存储于LDAP目录服务器中。LDAP自身带有多种灵活的条目检索、匹配和维护功能，这使证书状态的维护更为简便。通过各种LDAP开发工具，应用系统可以方便的在线获取证书或查询证书的状态。LDAP的目录复制功能 (Replica) 可以帮助解决大型分布式应用中各证书库的数据一致性维护问题。所以用LDAP技术实现证书库已经成为证书库设计的主流趋势。

6 总结

本文分析了LDAP协议使用的协议模型、数据模型和协议基本元素的细节，描述了协议操作中的数据格式，并给出了该协议的应用特点。

参考文献

1 M. Wahl, T. Howes, and S. Kille, "Lightweight Directory Access Protocol", RFC 2251, Dec 1997.

2 ITU-T Rec. X.500, "The Directory: Overview of Concepts, Models and Service", 1993.