网络安全协议及分析
网管安全SNMPv3

密码与网络安全系　刘虹

2025年春季学期

课程体系

第一章 概述

第二章 链路层扩展L2TP

第三章 IP层安全IPSec

第四章 传输层安全SSL和TLS

第五章 会话安全SSH

第六章 代理安全Socks

第七章 网管安全SNMPv3

第八章 认证协议Kerberos

第九章 应用安全

本章学习目标

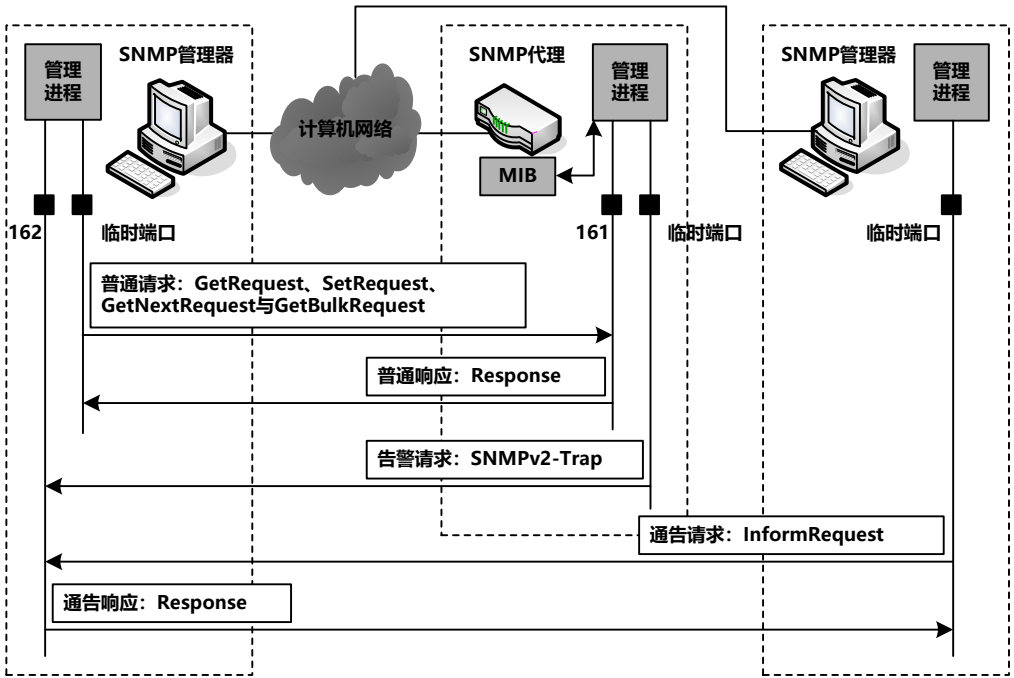
- 基于用户的安全模型USM
- 基于视图访问控制模型VACM

SNMP概述

- 简单网络管理协议（Simple Network Management Protocol，SNMP）是TCP/IP架构下的网络管理标准：
  - 包括至少一个网络管理站、多个被管理节点、管理信息和用于在SNMP实体之间传递管理信息的通信协议。
- SNMP通信协议工作于应用层，它可以在不同的传输层协议上工作
  - UDP、网间分组交换协议（Internetwork Packet Exchange，IPX)等。
- SNMP不应只被看做一个通信协议，它是一个有关网络管理体系结构的整体规范，包括以下要素：
  - 规范语言、MIB定义、协议定义以及安全与管理

三 SNMPv3体系结构

- SNMPv3没定义新的网管操作、消息类型与PDU结构，主要改进在于安全性
  - 定义一种标准化的SNMP框架结构
  - 为两种SNMPv1与SNMPv2消息提供各种安全功能



三 SNMPv3体系结构

- SNMPv3实体的主要类型
  - 管理器（Manager）：
    - 请求发送与响应接收
  - 代理（Agent）：
    - 请求接收与响应发送
  - 代理服务器（Proxy）：
    - 请求与响应转发

三

SNMPv3体系结构

- 调度程序（Dispatcher Model）：
  - 负责SNMP消息的发送与接收
- 消息处理模块（Message Processor Model）：
  - 负责SNMP消息的分析与处理
- 安全模块（Security Model）：
  - 负责SNMP消息的认证与加密
- 访问控制模块（Access Control Model）：
  - 负责MIB中对象的访问控制

三 SNMPv3应用

- **命令生成器**：用于SNMP管理器，生成SNMPv1与SNMPv2普通请求，接收SNMPv1与SNMPv2响应
- **命令接收器**：用于SNMP代理，接收SNMPv1与SNMPv2普通请求，生成SNMPv1与SNMPv2响应
- **通告生成器**：用于SNMP管理器或SNMP代理，为代理生成SNMPv1与SNMPv2告警请求与SNMPv2通告请求，接收SNMPv2通告响应
- **通告接收器**：用于SNMP管理器，接收SNMPv1与SNMPv2告警请求与SNMPv2通告请求，生成SNMPv2通告响应
- **代理转发器**：用于SNMP代理或代理服务器，转发接收的SNMP命令与响应

提纲

一、基于用户的安全模型USM

二、基于视图访问控制模型VACM

三、序列化

USM的基本概念

- 基于用户的安全模型USM
  - 用户提供用户名/口令，USM则将口令转化为共享密钥，这个过程称为密钥本地化。
  - USM提供数据完整性、机密性保护，提供数据源发认证功能，并能够防止重放攻击。
    - 完整性保护和数据源发认证功能由消息验证码提供。
    - 消息时效性利用时间参数和时间窗口保证。
    - 机密性保护功能则通过数据加密实现。

USM安全机制

- 用户：用户是USM的主体
  - 一个SNMPv3引擎可以有多个用户，它会维护这些用户的信息。
  - 当它需要与另一个引擎通信时，它也必须了解这个引擎已知的用户信息。
  - 同一个用户可以被定义于多个SNMP引擎。
    - “用户名（userName）"字符串，用于标识用户，与所使用的安全模型相关。
    - “安全名（securityName）”字符串，用于标识用户，与所使用的安全模型无关。
    - 用户名和安全名存在一一对应关系。SNMP应用看到的是安全名，USM看到的则是用户名，USM负责将安全名转化为用户名。

用户安全属性：

- 用户名
- 安全名
- 认证协议
- 认证密钥
- 认证密钥更改
- 加密算法
- 加密密钥
- 加密密钥更改

USM安全机制

用户表：SNMP引擎将用户信息存储于本地配置库

- usmUserEngineID可以为当前引擎的ID，也可以为与当前用户通信的远程引擎的ID
- usmUserName用户名
- usmUserSecurityName用户安全名
- usmUserCloneFrom在生成一个新用户时，它将克隆某个模板用户的信息
- usmUserAuthProtocol用户支持的认证协议
- usmUserAuthKeyChange当该对象值被修改时，认证密钥将被更新
- usmUserOwnAuthKeyChange与usmUserAuthKeyChange的功能类似
- usmUserPrivProtocol用户支持的加密协议
- usmUserPrivKeyChange当该对象值被修改时，加密密钥将被更新
- usmUserOwnPrivKeyChange与usmUserPrivKeyChange的功能类似
- usmUserPublic在更新密钥后，可以修改其值
- usmUserStorageType表中当前行的访问权限
- usmUserStatus表中当前行的状态，可以为notReady或active

USM安全机制

- SNMP引擎配置
  - 在安装一个SNMP权威引擎时，通常需要配置一些与角户相关的参数。
  - 第一类安全参数就是安全状态，存在以下三种状态：
    - minimum-secure
    - semi-secure
    - very-secure

USM安全机制

- 保障消息时效性：**权威实体**
  - 在消息时效性验证机制中，引入了“权威引擎”概念和时间变量。
  - SNMPv3将引擎分为两类：权威（authoritative）、非权威（non-authoritative)
    - 如果一个引擎发出的消息将引发对方返回响应，则收到该类消息的一方为权威；
    - 如果一个引擎发出的消息不会引发对方的响应，则该消息的发送方也是权威实体。
  - SNMPv3的协议数据单元PDU分为两类：
    - 需确认类：GetRequest-PDU、GetNextRequest-PDU、GetBulkRequest-PDU、SetRequest-PDU、InformRequest-PDU
    - 非需确认类：Report-PDU、SNMPv2-Trap-PDU、GetResponse-PDU
    - 确认类PDU的接收者和非确认类PDU的发送者可以被看做权威；反之为非权威。

USM安全机制

- 保障消息时效性：**时间变量**
  - 每个SNMP引擎都有唯一的ID，即snmpEngineID。

- 每个引擎都会维护两个时间变量
  - snmpEngineBoots：SNMP引擎被设置为当前ID后重新启动（或重新初始化）的次数；
  - snmpEngineTime：snmpEngineBoots最近一次加1以来所经过的秒数。

USM安全机制

- 保障消息时效性：时间同步
  - 每个非权威引擎负责与每个与之通信的权威引擎进行时间同步，它会为每个与其通信的权威引擎维护一个四元组：
    - <snmpEngineID，snmpEngineBoots，snmpEngineTime，latestReceivedEngineTime>
    - latestReceivedEngineTime表示非权威引擎收到的权威引擎的最大值，用于防止由于消息重放而造成的对时间变量推进的阻滞。
  - 初始时，非权威引擎的本地时间变量应设置为0。由于时间同步是与消息认证功能绑定在一起的，所以，仅当收到一个来自权威引擎的包含认证信息的消息时，非权威引擎才会更新本地时间变量。

USM安全机制

- 保障消息时效性：**时效性检查**
  - 权威引擎和非权威引擎都会对收到的消息进行时效性检查。
  - 如果消息中包含的msgAuthoritativeEngineID与当前处理消息的引擎ID一致，说明当前是权威引擎；否则是非权威引擎。

USM安全机制

- 保障消息时效性：**引擎发现**
  - 当非权威引擎还未掌握权威引擎的snmpEngineID时，必须使用USM提供的引擎发现功能。
  - 如果期望在通信过程中使用认证功能，则在获取了权威引擎的ID后，非权威引擎还需要与权威引擎的时钟保持同步。

USM安全机制

- 保障消息时效性：分析
  - SNMPv3的消息时效性保护与消息认证功能捆绑在一起，这是因为前者依托消息中所携带的时间数据。如果这些数据被更改，时效性保护将失去意义。
    - 防止重放过时消息；
      防止发往某个权威引擎的消息被重放到另一个权威引擎；
      防止来自某个权威引擎的消息被伪装成来自另一个权威引擎的消息。
  - 每个SNMP消息中都包含三个字段
    - msgAuthoritativeEngineID
    - msgAuthoritativeEngineBoots
    - msgAuthoritativeEngineTime

- 对于权威引擎，包含自身引擎ID以及时间参量；
- 对于非权威，描述目标引擎的相关参量。

USM安全机制

- **密钥本地化：**
  - **认证和加密功能都需要使用共享密钥。**
  - **USM定义了将用户口令转化为密钥的方法。**
    - **一个“本地化的密钥”是用户U与一个特定权威引擎E之间共享的密钥。**
    - **一个非权威引擎的本地用户将和远程的权威引擎建立密钥；**
    - **一个权威引擎的本地用户也将与本地引擎建立密钥。**

将用户口令转化为密钥的过程即为“密钥本地化”

USM安全机制

- 密钥更新
  - USM并未定义密钥交换方法，而是由用户直接指定相关值，即在安装SNMP引擎时，安装者需输入口令，而USM会利用密钥本地化功能将这个**口令转化为密钥**。
  - 这个密钥可通过人工或其他安全协议发送给远程SNMP引擎。
  - 一旦通信双方共享authKey和privKey，即可通过更改远程实体usmUserEntry中的密钥更新对象值以更新密钥。

USM安全机制

- 安全参数
  - 每个SNMP消息中都包含安全参数字段，用于消息认证和机密性保护。
- 认证处理
  - 基于MD5的HMAC、基于SHA的HMAC。
- 加密处理
  - USM仅对SNMP消息的PDU部分进行加密处理。

USM流程

- 对外发送的请求消息

```
statusInformation= generateRequestMsg（
                   INmessageProcessingModel
                   INgloableData
                   INmaxMessageSize
                   INsecurityModel
                   INsecurityEngineID
                   INsecurityName
                   INsecurityLevel
                   INscopedPDU
                   OUTsecurityParameters
                   OUTwholeMsg
                   OUTwholeMsgLength)
```

USM流程

- 进入的消息

```
statusInformation= processIncomingMsg（
                   INmessageProcessingModel
                   INmaxMessageSize
                   INsecurityParameters
                   INsecurityModel
                   INsecurityLevel
                   INwholeMsg
                   INwholeMsgLength
                   OUTsecurityEngineID
                   OUTsecurityName
                   OUTscopedPDU
                   OUTmaxSizeResponseScopedPDU
                   OUTsecurityStateReference)
```

USM流程

- 对外发送的应答消息

```
statusInfortnation= generateResponseMsg（
                    INmessageProcessingModel
                    INgloableData
                    INmaxMessageSize
                    INsecurityModel
                    INsecurityEngineID
                    INsecurityName
                    INsecurityLevel
                    INscopedPDU
                    INsecurityStateReference
                    OUTsecurityParameters
                    OUTwholeMsg
                    OUTwholeMsgLength)
```

提纲

一、基于用户的安全模型USM

二、基于视图访问控制模型VACM

三、序列化

二 VACM要素

- VACM要素：
  - **组**：一个组包含了0个或多个<securityModel，securityName>二元组。每个二元组都描述了一个用户，同一组中的用户具有相同的访问权限。每个组用"groupName"标识。
  - **安全级别**：SecurityLevel描述一个组的安全需求，其含义及取值与USM所定义的安全级别相同。
  - **上下文**：上下文是SNMP实体可以访问的管理信息集合。一个上下文中可包含多条管理信息，一条管理信息也可以位于不同的上下文中。

二 VACM要素

- VACM要素：
  - MIB视图：一个上下文中管理对象的子集
  - 视图家族：由标识符（家族名）与字符串（家族掩码）组成
  - 访问策略：由不同视图获得的访问权限
    - 读视图：包含这个组可以实施读访问的对象实例集合。
    - 写视图：包含这个组可以实施写访问的对象实例集合。
    - 通知视图：包含这个组生成的通知消息中可包含的对象实例集合。

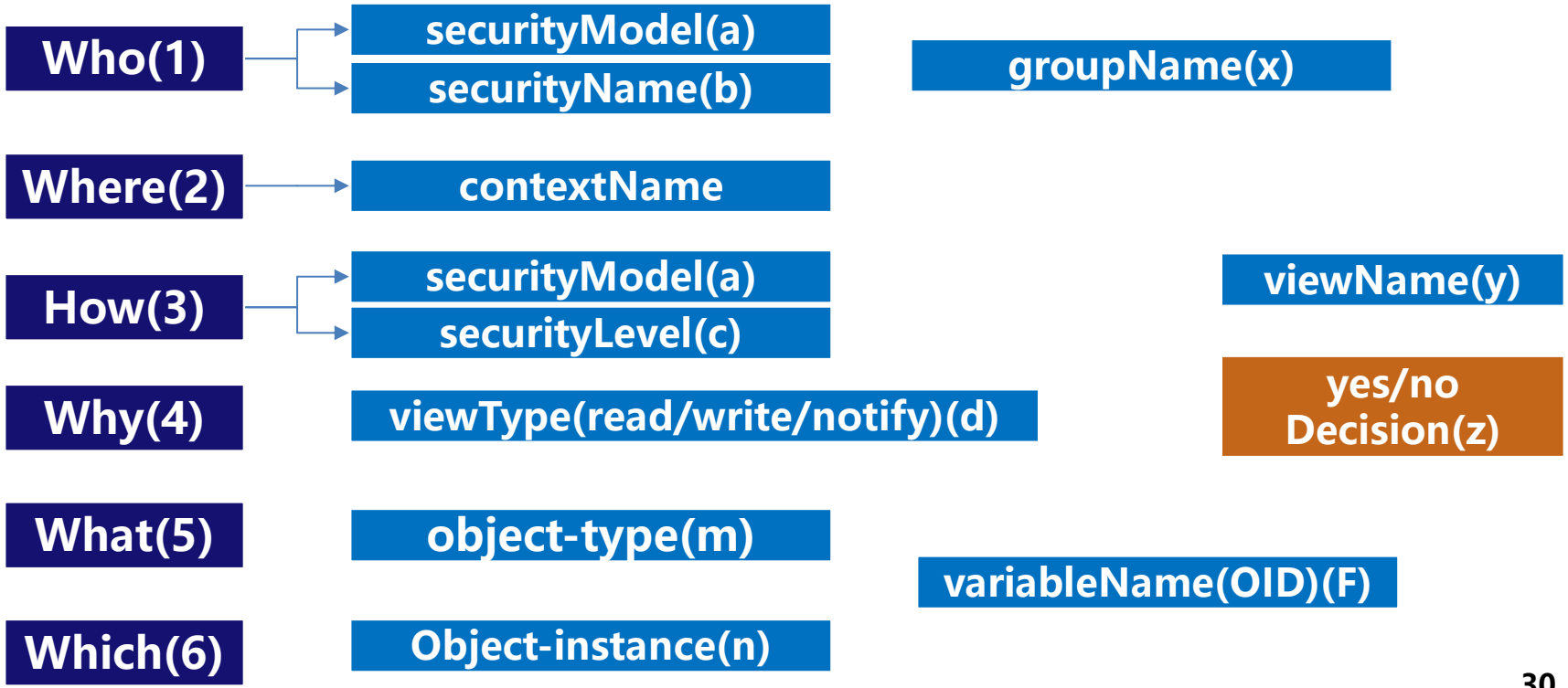
二 VACM管理对象

- 认证流程：**接口原语**
  - 所使用的安全模型、用户安全名、所需的安全级别、访问类型（读、写、通知）、待访问的上下文以及上下文中的变量。

```
statusInformation = isAceessAllowed(
                    INsecurityModel
                    INsecurityName
                    INsecurityLevel
                    INviewType
                    INcontextName
                    INvariableName)
```

二 VACM管理对象

- 执行流程：



二 VACM管理对象

- 执行流程：
  - 用contextName从vacmContextTable中查找上下文，如果没有找到，返回noSuchContext
  - 用securityModel与securityName从vacmSecurityToGroupTable中查找组，如果没有找到，返回noSuchGroup
  - 用securityModel、securityLevel、contextName与groupName从vacmAccessTable中查找行，如果没有找到，返回noAccessEntry

二 VACM管理对象

- 执行流程：
  - 用viewType从vacmAccessTable中查找视图类型，如果没有找到，返回noSuchView
  - 用viewName在vacmViewTreeFamilyTable中查找视图子树，如果没有找到，返回noSuchView
  - 用variableName在vacmViewTreeFamilyTable查找管理对象，如果找到，返回notInView；否则，返回accessAllowed

二 VACM管理对象

- 认证流程：
  - VACM配置
    - Initial-minimum-security-configuration
    - Initial-semi-security-configuration
    - initial-no-access-configuration
  - 初始的访问权限
    - noAuthNoPriv：不认证不加密
    - authNoPriv：认证不加密
    - authPriv：认证且加密

- 一个默认的上下文
- 一个初始组
- 初始的访问权限

提纲

一、基于用户的安全模型USM

二、基于视图访问控制模型VACM

三、序列化

三 序列化

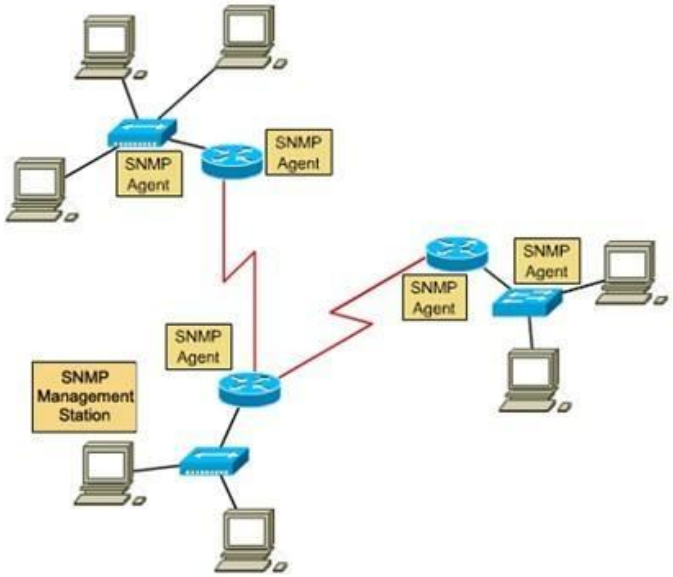
- 数据类型
  - 简单类型
    - 整数、字符串、OID
  - 简单结构类型
    - 列表（list）、表格（table）
  - 应用数据类型
    - IpAddress以网络字节顺序表示的IP地址
    - Counter3232b计数器
    - TimeTicks时间计数器
    - Opaque特殊的数据类型，可看作无格式的二进制串；
    - Unsigned32(Gauge32)
    - Counter64b计数器

三 序列化

- SNMPv3报文序列化
  - 在传输SNMP数据之前，必须用BER对消息进行编码；收到消息后，必须对消息进行还原处理。
  - 整个消息以及消息中的每个字段都被编码成三元组。

三 SNMPv3应用

- 网络管理系统是SNMP的直接应用
  - 思科的CiscoWorks
  - 华为的eSight
- SNMP虽然称为简单网络管理协议，但从程序开发者的角度看并不简单，因为编写这类应用不仅要进行消息安全处理，还必须进行复杂的序列化和解码操作。



小结

- SNMP是TCP/IP协议族下的网络管理框架
  - SNMP框架包括管理站、被管节点、通信协议和管理信息
- SNMP最早的版本是SNMPv1，这个版本使用基于共同体名的访问控制机制，这个共同体名相当于一个明文口令。
  - SNMPv1不提供数据机密性和完整性保护。
- SNMPv3沿用SNMPv2的框架
  - 增加机密性、完整性保护功能，同时可以防止重放攻击。

小结

- SNMPv3实体由一个SNMP引擎和若干个应用构成，引擎包括：
  - 调度程序、消息处理模块、安全模块、访问控制模块
- SNMPv3应用包括：
  - 命令生成器、命令接收器、通告生成器、通告接收器、代理转发器
- SNMPv3定义了8种ScopedPDU：
  - 读类：读管理信息，包括GetRequest-PDU、GetNextRequest-PDU以及CetBulkRequest-PDU；
  - 写类：更改管理信息，即SetRequest-PDU；
  - 响应类：对请求的响应，包括Response-PDU和Report-PDU
  - 通知类：通知操作，包括SNMPv2-Trap-PDU和InformRequest-PDU；
  - 内部类：用于SNMP引擎之间的通信，即Report-PDU。

小结

- 基于用户的安全模型USM
  - MAC实现消息完整性保护和数据源发认证
  - 认证协议可以使用HMAC-MD5-96和HMAC-SHA-96
  - 使用CBC-DES加密数据以实现机密性保护

- 基于视图的访问控制模型VACM
  - 包含组、安全级别、上下文、MIB视图和视图族、访问策略，描述主体、访问目标及访问方式。
  - 以组为单位设置访问权限，一个组中包含多个用户。一个上下文中可包含多个MIB视图和视图族，它们是这个上下文对象的子集。
  - VACM将视图分为读、写、通知，用以描述不同的访问策略。

办公地点：理科大楼B1209

联系方式：17621203829

邮箱：liuhongler@foxmail.com