Unit 2

Internet based Network Management
SNMP V3 Security

- Introduce three important services
  - Authentication
  - Privacy
  - Access Control

*Data Encryption Standard*
SNMP V3 Architecture

- SNMPv3 is defined in a modular fashion
- Each SNMP entity includes a single SNMP engine
- Engine implements functions for
  - Sending and receiving messages
  - Authenticating and encrypting/decrypting messages
  - Controlling access to managed objects
- Role of entity determines what modules are implemented
- Can define different versions for each module (e.g. SNMPv4!)
SNMP V3 Architecture

- **SNMP Engine**
- **Dispatcher**: Allows for concurrent support of multiple versions of SNMP messages in the SNMP engine. It is responsible for
  - (1) accepting protocol data units (PDUs) from applications for transmission over the network and delivering incoming PDUs to applications;
  - (2) passing outgoing PDUs to the Message Processing Subsystem to prepare as messages, and passing incoming messages to the Message Processing Subsystem to extract the incoming PDUs; and
  - (3) sending and receiving SNMP messages over the network.
SNMP V3 Architecture

- **SNMP Engine**
- **Message Processing Subsystem:** Responsible for preparing messages for sending and for extracting data from received messages.

- **Security Subsystem:** Provides security services such as the authentication and privacy of messages.

- **Access Control Subsystem:** Provides a set of authorization services that an application can use for checking access rights.
SNMP V3 Architecture

- **SNMP Applications**
- **Command Generator**: Initiates SNMP Get, GetNext, GetBulk, or Set request PDUs and processes the response to a request that it has generated.
- **Command Responder**: Receives SNMP Get, GetNext, GetBulk, or Set request PDUs destined for the local system.
- The command responder application performs the appropriate protocol operation, using access control, and generates a response message to be sent to the originator of the request.
SNMP V3 Architecture

- **SNMP Applications**
- **Notification Originator**: Monitors a system for particular events or conditions, and generatesTrap or Inform messages based on these events or conditions. A notification originator must have a mechanism for determining where to send messages, and which SNMP version and security parameters to use when sending messages.

- **Notification Receiver**: Listens for notification messages, and generates response messages when a message containing an Inform PDU is received.

- **Proxy Forwarder**: Forwards SNMP messages. Implementation of a proxy forwarder application is optional.
SNMP V3 Message Format

- **msgVersion**: set to SNMPv3
- **msgID**: A unique identifier used between two SNMP entities to coordinate request and response messages. The range of this ID is 0 through $2^{31} - 1$.
- **msgMaxSize**: Conveys the maximum size of a message in octets supported by the sender of the message, with a range of 484 through $2^{31} - 1$.
- **msgFlags**: An octet string containing three flags in the least significant three bits: reportableFlag, privFlag, authFlag.
- **msgSecurityModel**: An identifier that indicates which security model was used by the sender to prepare this message and, therefore, which security model must be used by the receiver to process this message.
SNMP V3 User-Based Security Model

• The User-Based Security Model (USM) uses the concept of an authoritative engine.
• In any message transmission, one of the two entities, transmitter or receiver, is designated as the authoritative SNMP engine.

**Authorative Engine**

• This is decided based on the following rules:

  • When SNMP message contains a payload that expects a response => the receiver of such messages is authoritative.
    • for example, a Get, GetNext, GetBulk, Set, or Inform PDU
  
  • When an SNMP message contains a payload that does not expect a response => the sender of such a message is authoritative.
    • for example, an SNMPv2-Trap, Response, or Report PDU
SNMP V3 User-Based Security Model

• This designation serves two purposes:

  • **Firstly**, the timeliness of a message is determined with respect to a clock maintained by each authoritative engine.
    • Authorative engine sends a message, it contains the current value of its clock
      • Non-authoritative recipient synchronizes on that clock.
      • E.g. authorative sends Trap, Response, Report

  • Each non-authorative engine maintains an estimate of the time value for each authorative engine with which it communicates
    • Non-authoritative engine sends a message, it contains current estimate of the time value at the destination
      • destination can assess the timeliness of the message.
      • E.g. Non-authorative sends Get, GetNext, GetBulk, Set, Inform
SNMP V3 User-Based Security Model

- **Secondly**, a key localization process enables a single principal management station to own keys stored in multiple engines.

  - Keys are localized to the authoritative engine in such a way
    - the principal management station is responsible for a single key
    - avoids the security risk of storing multiple copies of the same key in a distributed network.

  - When an outgoing message is passed to the USM by the Message Processor, the USM fills in the security-related parameters in the message header.

  - When an incoming message is passed to the USM by the Message Processor, the USM processes the values contained in those fields.
SNMP V3 User-Based Security Model

• The security-related parameters include the following:
  • **msgAuthoritativeEngineID**: The snmpEngineID of the authoritative SNMP engine involved in the exchange of this message.

  • **msgAuthoritativeEngineBoots**: The snmpEngineBoots value of the authoritative SNMP engine involved in the exchange of this message. The object snmpEngineBoots is an integer in the range 0 through 2^31 — 1 that represents the number of times that this SNMP engine has initialized or reinitialized itself since its initial configuration.

  • **msgAuthoritativeEngineTime**: The snmpEngineTime value of the authoritative SNMP engine involved in the exchange of this message. The object snmpEngineTime is an integer in the 0 through 2^31 — 1 range that represents the number of seconds since this authoritative SNMP engine last incremented the snmpEngineBoots object.
    • Each authoritative SNMP engine is responsible for incrementing its own snmpEngineTime value once per second.
    • A non-authoritative engine is responsible for incrementing its notion of snmpEngineTime for each remote authoritative engine with which it communicates.

  • **msgUserName**: The user (principal) on whose behalf the message is being exchanged.

  • **msgAuthenticationParameters**: Null if authentication is not being used for this exchange; otherwise, this is a privacy parameter. In USM, the privacy parameter is a parameter used in the encryption algorithm DES.
Privacy Using Conventional Encryption

- The authentication mechanism in SNMPv3 assures received message was transmitted by the principal whose identifier appears as the source in the message header.
  - And that the message was not altered in transit and that it was not artificially delayed or replayed.

- **To achieve authentication:**
  - principal and remote SNMP share a secret authentication key.
  - sending entity provides the authentication code with the SNMPv3 message it is sending.
  - This code is a function of the contents of the message, the identity of the principal and engine, the time of transmission, and a secret key that should be known only to the sender and the receiver.
  - The secret key must initially be set up outside of SNMPv3 as a configuration function.
    - network manager is responsible for distributing initial secret keys to be loaded into the databases of the various SNMP managers and agents.
    - This can be done manually or by using some form of secure data transfer outside of SNMPv3.
View-Based Access Control Model

• The access control facility makes it possible to configure agents to provide different levels of access to the agent's MIB to different managers.

• An agent entity can restrict access to its MIB for a particular manager entity in two ways.
  • First, it can restrict access to a certain portion of its MIB.
  • Second, the agent can limit the operations that a principal can use on that portion of the MIB.

• The access control policy to be used by an agent for each manager must be preconfigured; it essentially consists of a table that details the access privileges of the various authorized managers.

• Unlike authentication, which is done by user, access control is done by group, where a group may be a set of multiple users.
# SNMP Security

<table>
<thead>
<tr>
<th>Version</th>
<th>Level</th>
<th>Authentication</th>
<th>Encryption</th>
<th>What Happens</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>NoAuth NoPriv</td>
<td>Community String</td>
<td>None</td>
<td>Uses a community string match for authentication</td>
</tr>
<tr>
<td>V2c</td>
<td>NoAuth NoPriv</td>
<td>Community String</td>
<td>None</td>
<td>Uses a community string match for authentication</td>
</tr>
<tr>
<td>V3</td>
<td>NoAuth NoPriv</td>
<td>username</td>
<td>None</td>
<td>Uses a username match for authentication</td>
</tr>
<tr>
<td>V3</td>
<td>Auth NoPriv</td>
<td>MD5 or SHA</td>
<td>None</td>
<td>Provides authentication based on the Hash-Based Message Authentication Code (HMAC) Message Digest 5 (MD5) algorithm or the HMAC Secure Hash Algorithm (SHA).</td>
</tr>
<tr>
<td>V3</td>
<td>Auth Priv</td>
<td>MD5 or SHA</td>
<td>DES</td>
<td>Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides Data Encryption Standard (DES) 56-bit encryption in addition to authentication based on the Cipher Block Chaining (CBC) DES (DES-56) standard</td>
</tr>
</tbody>
</table>
SNMP Question

You are analysing a single LAN which has one management station which is responsible for managing a network of agents. Each agent is to be polled every 30 minutes. It takes 99ms to process a query, with a network delay of about 2ms.

a) What is the maximum number of agents that the management station can handle when engaged in full-time polling?

b) If a single poll takes 1KB of bandwidth, and the overall capacity of the network is 100MB per hour, what is the effect of your polling scheme on utilisation at full-time polling?

c) How might you reduce this cost?