SNMP Version 3

(Another attempt to finish work on secure SNMP)

Jürgen Schönwälder

Lucent Technologies
Bell-Laboratories
Room 2B-422
600 Mountain Avenue
Murray Hill, NJ 07974

E-mail: jschon@research.bell-labs.com
Phone: +1 908 582 6794

Structure of the Talk

1. SNMP Architecture
2. SNMPv3 Applications and MIBs
3. SNMP View-Based Access Control
4. SNMPv3 Message Processing Model
5. SNMPv3 User-Based Security Model
6. Discussion

Goals of the SNMP Architecture

- An architecture which allows to upgrade portions of SNMP without disrupting an entire SNMP framework.
- Allow inexpensive minimal implementations while providing the features required in large networks.
- Address the need for secure SNMP set operations.
- Keep SNMP as simple as possible.
- Use existing material as much as possible (SNMPv2u, SNMPv2*).
- Provide MIBs for remote configuation of SNMP engines.

Elements of the SNMP Architecture

- Dispatcher

  Responsible to dispatch messages and PDUs to the correct Message Processing Model or Application within an SNMP entity.

- Message Processing Subsystem

  Responsible for preparing outgoing messages and processing received messages. Can contain multiple Message Processing Models.

- Security Subsystem

  Provides security services such as authentication and privacy of messages by means of one or more Security Models.

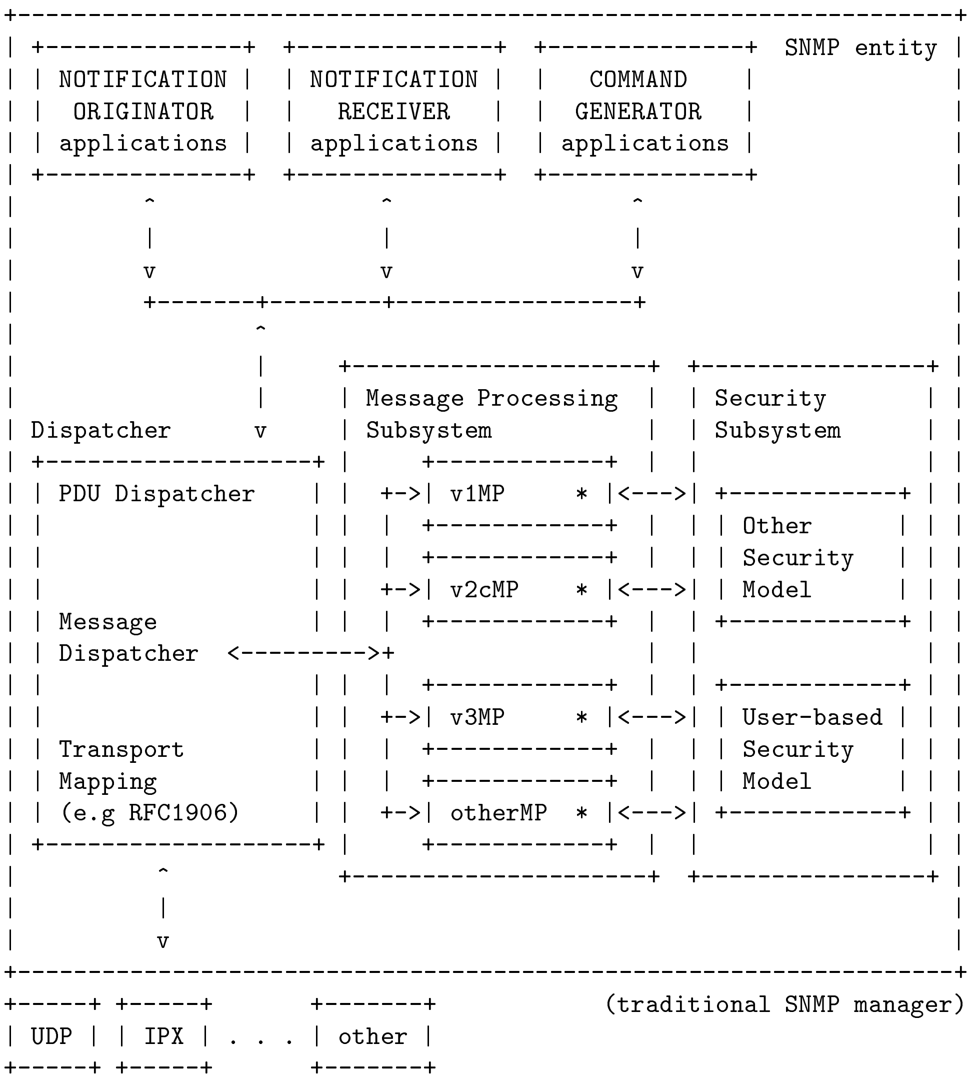
- Access Control Subsystem

  Provides authorization services by means of one or more Access Control Models.

- Applications

  Applications provide access to and/or manipulate management data. Applications can emit notifications or forward SNMP messages between SNMP entities.

```
+-------------------------------------------------------------------+
|  SNMP entity                                                      |
|                                                                   |
|  +-------------------------------------------------------------+  |
|  |  SNMP engine (identified by snmpEngineID)                   |  |
|  |                                                             |  |
|  |  +------------+ +------------+ +-----------+ +-----------+  |  |
|  |  |            | |            | |           | |           |  |  |
|  |  | Dispatcher | | Message    | | Security  | | Access    |  |  |
|  |  |            | | Processing | | Subsystem | | Control   |  |  |
|  |  |            | | Subsystem  | |           | | Subsystem |  |  |
|  |  |            | |            | |           | |           |  |  |
|  |  +------------+ +------------+ +-----------+ +-----------+  |  |
|  |                                                             |  |
|  +-------------------------------------------------------------+  |
|                                                                   |
|  +-------------------------------------------------------------+  |
|  |  Application(s)                                             |  |
|  |                                                             |  |
|  |  +-------------+  +--------------+  +--------------+        |  |
|  |  | Command     |  | Notification |  | Proxy        |        |  |
|  |  | Generator   |  | Receiver     |  | Forwarder    |        |  |
|  |  +-------------+  +--------------+  +--------------+        |  |
|  |                                                             |  |
|  |  +-------------+  +--------------+  +--------------+        |  |
|  |  | Command     |  | Notification |  | Other        |        |  |
|  |  | Responder   |  | Originator   |  |              |        |  |
|  |  +-------------+  +--------------+  +--------------+        |  |
|  |                                                             |  |
|  +-------------------------------------------------------------+  |
|                                                                   |
+-------------------------------------------------------------------+
```



```
+-------------------------------------------------------------------+
| +--------------+  +--------------+  +--------------+  SNMP entity |
| | NOTIFICATION |  | NOTIFICATION |  |   COMMAND    |              |
| |  ORIGINATOR  |  |   RECEIVER   |  |  GENERATOR   |              |
| | applications |  | applications |  | applications |              |
| +--------------+  +--------------+  +--------------+              |
|         ^                ^                 ^                      |
|         |                |                 |                      |
|         v                v                 v                      |
|         +-------+--------+-----------------+                      |
|                 ^                                                 |
|                 |     +---------------------+  +----------------+ |
|                 |     | Message Processing  |  | Security       | |
| Dispatcher      v     | Subsystem           |  | Subsystem      | |
| +-------------------+ |     +------------+  |  |                | |
| | PDU Dispatcher    | |  +->| v1MP     * |<--->| +------------+ | |
| |                   | |  |  +------------+  |  | | Other      | | |
| |                   | |  |  +------------+  |  | | Security   | | |
| |                   | |  +->| v2cMP    * |<--->| | Model      | | |
| | Message           | |  |  +------------+  |  | +------------+ | |
| | Dispatcher  <--------->+                  |  |                | |
| |                   | |  |  +------------+  |  | +------------+ | |
| |                   | |  +->| v3MP     * |<--->| | User-based | | |
| | Transport         | |  |  +------------+  |  | | Security   | | |
| | Mapping           | |  |  +------------+  |  | | Model      | | |
| | (e.g RFC1906)     | |  +->| otherMP  * |<--->| +------------+ | |
| +-------------------+ |     +------------+  |  |                | |
|          ^            +---------------------+  +----------------+ |
|          |                                                        |
|          v                                                        |
+-------------------------------------------------------------------+
+-----+ +-----+       +-------+            (traditional SNMP manager)
| UDP | | IPX | . . . | other |
+-----+ +-----+       +-------+
```

```
+-----+ +-----+       +-------+
| UDP | | IPX | . . . | other |
+-----+ +-----+       +-------+              (traditional SNMP agent)
+-------------------------------------------------------------------+
|              ^                                                    |
|              |        +---------------------+  +----------------+ |
|              |        | Message Processing  |  | Security       | |
| Dispatcher   v        | Subsystem           |  | Subsystem      | |
| +-------------------+ |     +------------+  |  |                | |
| | Transport         | |  +->| v1MP     * |<--->| +------------+ | |
| | Mapping           | |  |  +------------+  |  | | Other      | | |
| | (e.g. RFC1906)    | |  |  +------------+  |  | | Security   | | |
| |                   | |  +->| v2cMP    * |<--->| | Model      | | |
| | Message           | |  |  +------------+  |  | +------------+ | |
| | Dispatcher  <--------->+                  |  |                | |
| |                   | |  |  +------------+  |  | +------------+ | |
| |                   | |  +->| v3MP     * |<--->| | User-based | | |
| |                   | |  |  +------------+  |  | | Security   | | |
| |                   | |  |  +------------+  |  | | Model      | | |
| | PDU Dispatcher    | |  +->| otherMP  * |<--->| +------------+ | |
| +-------------------+ |     +------------+  |  |                | |
|              ^        +---------------------+  +----------------+ |
|              |                                                    |
|      +-------+-------------------------+---------------+         |
|      v                                 v               v         |
| +-------------+   +---------+   +--------------+  +-------------+ |
| |   COMMAND   |   | ACCESS  |   | NOTIFICATION |  |    PROXY  * | |
| |  RESPONDER  |<->| CONTROL |<->|  ORIGINATOR  |  |  FORWARDER  | |
| | application |   |         |   | applications |  | application | |
| +-------------+   +---------+   +--------------+  +-------------+ |
|      ^                                 ^                          |
|      v                                 v                          |
| +----------------------------------------------+                  |
| |             MIB instrumentation              |      SNMP entity |
+-------------------------------------------------------------------+
```

SNMPv3 Applications and MIBs

- Command Generators:

  Applications which initiate SNMP Get, GetNext, GetBulk and/or Set operations.

- Command Responders:

  Applications which respond to SNMP Get, GetNext, GetBulk and/or Set operations.

- Notification Originators:

  Applications which generate notifications (Traps and Informs!).

- Notification Receivers:

  Applications which receive notifications.

- Proxy Forwarders:

  Applications which forward SNMP messages to other SNMP engines.

$\Longrightarrow$ The terms "manager" and "agent" are historic now!

SNMP Target MIB

- The snmpTargetAddrTable lists target addresses and transport related parameters (timeout, retries).
- The snmpTargetParamsTable lists SNMP parameters such as security mode, security level or security name.
- SNMP parameters can be shared between multiple targets.
- Targets addresses can have a list of tags which identify properties of targets.
- A single management operation can be applied to a set of target addresses by specifying restrictions on the set of tags (late binding).
- Tags are important hooks for the DISMAN work.

SNMP Notification MIB

- The snmpNotifyTable selects the set of targets that should receive a notification and whether the notification is send as a trap PDU or as a inform PDU.

- The snmpNotifyFilterProfileTable associates a particular notification filter to a target.

- The snmpNotifyFilterTable defines a filter which matches the notification type (the value of snmpTrapOID) against a subtree pattern.

- A positive match can inculde or exclude a notification type.

- No filtering is performed on targets that do not have a filter.

SNMP Proxy MIB

- The snmpProxyTable defines translations between management targets for use when forwarding messages.
- Incoming operations to read or write MIB objects are forwarded to a target identified by an entry in the snmpTargetAddrTable.
- Incoming notifications can be forwarded to a set of targets (fan out).
- SNMPv1 messages can be translated into SNMPv2 or SNMPv3 messages.
- SNMPv2 or SNMPv3 messages are not translated into SNMPv1 messages.

View-Based Access Control

- Multiple securityNames (principals) can be mapped to a single group.
- Different access rights for members of a group can be defined for different Levels of Security.
- A MIB view is the union of a set of “view subtrees”, where each view subtree is a subtree within the managed object naming tree.
- The read-view represents the set of object instances authorized for the group when reading objects.
- The write-view represents the set of object instances authorized for the group when writing objects.
- The notify-view represents the set of object instances authorized for the group when sending objects in a notification.
- A MIB is provided to configure MIB views, groups and group membership.

SNMPv3 Message Processing Model

- SNMPv3 uses scoped PDUs:
  - RFC 1905 PDUs encapsulated in a ScopedPDU header.
  - Scope of a PDU defined by `contextEngineID` and `contextName`.
  - The `request-id` field in a RFC 1905 PDU is not used anymore.
- SNMPv3 messages transport scoped PDUs:
  - The `msgVersion` is similar to the SNMPv1/SNMPv2c version field.
  - The `msgID` replaces the `request-id` field in a RFC 1905 PDU.
  - The `msgMaxSize` is used to make getbulk processing work.
  - The `msgFlags` are used to encode the security level and report flags.
  - The `msgSecurityModel` defines which Security Model should be applied.

SNMPv3 Message Processing Model

- A MIB module is provided to count SNMPv3 message specific errors.
- Problems with the current message format:
  - The security information is currently embedded in the SNMP message header.
  - The security model needs to have knowledge about the message format because it is responsible to encode the SNMPv3 message.
  - A proposal to address these problems by putting a security envelope around SNMPv3 messages instead of carrying the security information inside SNMPv3 messages did not reach concensus.
  - The `msgMaxSize` field needs to be broken down into a value usable for command responder applications.
  - This computation requires to make assumptions about sizes required for BER encoded security and message header.

SNMPv3 User-Based Security Model

- Protection against the following security threads:
  - Modification of Information
  - Masquerade
  - Message Stream Modification
  - Disclosure
- Security services provided by the User-Based Security Model:
  - Data Integrity
  - Data Origin Authentication
  - Data Confidentiality
  - Message timeliness and limited replay protection

SNMPv3 User-Based Security Model

- Keyed-Hashing for Message Authentication (HMAC).
- Authentication with HMAC-MD5 (required) and HMAC-SHA-1 (optional).
- Replay protection through loosely synchronized clocks.
- Encryption with the CBC-DES Symmetric Encryption Protocol (optional).
- ASN.1 definition of the security header format still under discussion.
- A MIB module which contains security related error statistics and a table used to configure users and secrets.