



SNMP-Versions

[a deeper look inside]

SNMP versions

- The router supports SNMP version 1 (SNMP-v1), SNMP version 2c (SNMP-v2c) and SNMP Version 3 (SNMP-v3).
- The three versions operate similarly.
 - SNMP-V1
 - SNMP-V2C
 - SNMP-V3

- SNMP-v2c updated the original protocol, and offered the following main enhancements:
 - a new format for trap messages.
 - the get-bulk-request PDU allows for the retrieval of large amounts of data, including tables, with one message.
 - more error codes mean that error responses to set messages have more detail than is possible with SNMP-v1.
 - three new exceptions to errors can be returned for get, get-next and get-bulk-request messages.
 - ▲ These are: *noSuchObject*, *noSuchInstance*, and

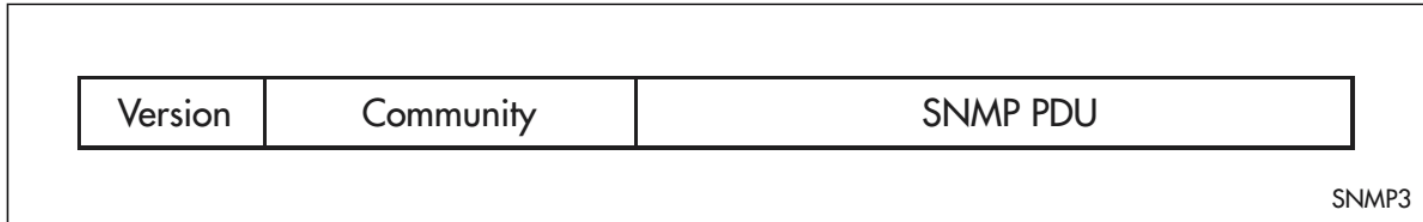
- SNMP-v3 provides significant enhancements to address the security weaknesses existing in the earlier versions.
 - This is achieved by implementing two new major features:
 - ▲ Authentication - by using password hashing and time stamping.
 - ▲ Privacy - by using message encryption.

- Support for multiple versions of SNMP is achieved by responding to each SNMP request with a response of the same version.
 - For example, if an SNMPv1 request is sent to the switch, an SNMPv1 response is returned.
 - If an SNMPv2c request is sent, an SNMPv2c response is returned.
 - Therefore, authentication and encryption functions are not invoked when messages are detected as having either an SNMPv1 or SNMPv2c protocol format.

SNMP Messages

- The SNMP protocol is termed simple because it has only six operations, or messages—
 - get, get-next, get-response, set, and trap, and
 - SNMPv2c also has the **get-bulk-request** message.
 - The two major SNMP operations available to a management station for interacting with a client are
 - the **get** and **set** operations.
 - ▲ The SNMP set operator can lead to security breaches, since SNMP is not inherently very secure.

Message format for SNMP v1 and SNMP v2c



Field	Function
Version	The version of the SNMP protocol. The value is version-1 (0) for the SNMP protocol as defined in RFC 1157, or version-2c (1) for the SNMP protocol as defined in RFC 1902.
Community	The name of an SNMP community, for authentication purposes.
SNMP PDU	An SNMP Protocol Data Unit (PDU).

SNMP PDU's

PDU	Function
get-request	Sent by an NMS to an agent, to retrieve the value of an object.
get-next-request	Sent by an NMS to an agent, to retrieve the value of the next object in the sub-tree. A sub-tree is traversed by issuing a get-request PDU followed by successive get-next-request PDUs.
get-bulk-request	Sent by an NMS to an agent to request a large amount of data with a single message. This is for SNMPv2c messages.
set-request	Sent by an NMS to an agent, to manipulate the value of an object.
get-response	Sent by an agent to an NMS in response to a get-request, get-next-request, get-bulk-response, or set-request PDU.
trap	Sent by an agent to an NMS to notify the NMS of a extraordinary event.
report	Although not explicitly defined in the RFCs, reports are used for specific purposes such as EngineID discovery and time synchronisation.

Generic SNMP Traps

Value	Meaning
coldStart	The agent is re-initialising itself. Objects may be altered.
warmStart	The agent is re-initialising itself. Objects are not altered.
linkDown	An interface has changed state from up to down.
linkUp	An interface has changed state from down to up.
authenticationFailure	An SNMP message has been received with an invalid community name.
egpNeighborLoss	An EGP peer has transitioned to down state.

SNMP Communities (Version v1 and v2c)

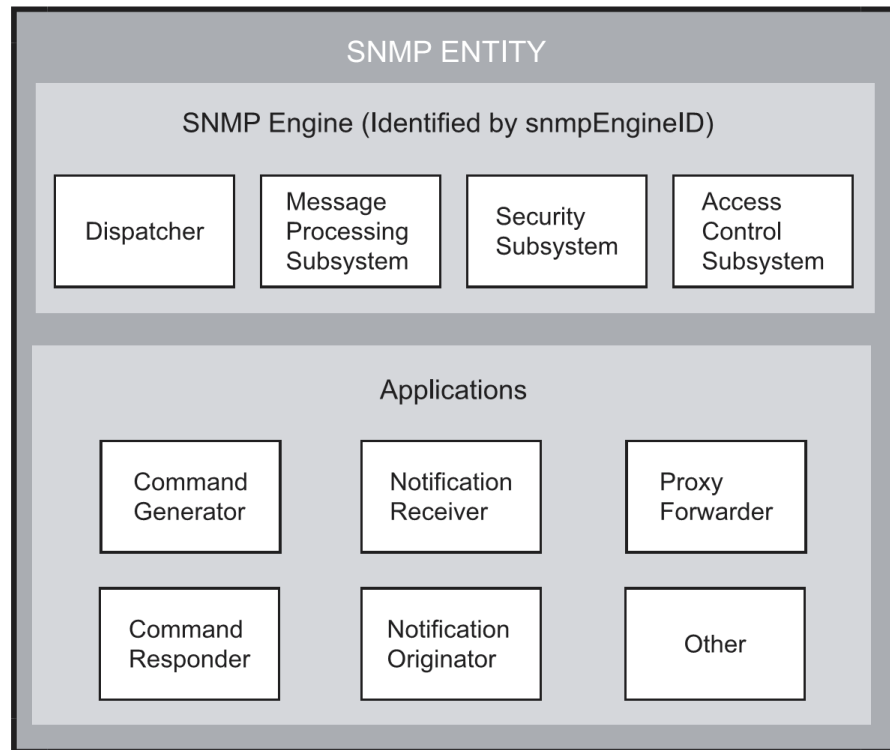
- A community is a relationship between an NMS and an agent.
 - The community name is used like a password for a trivial authentication scheme.
 - Both SNMPv1 and SNMPv2c provide security based on the community name only.
 - The concept of communities does not exist for SNMPv3,
 - ▲ which instead provides for a far more secure communications method using entities, users and groups.

SNMP-v3 Entities

- Entities comprise one of the basic components of the SNMP-v3 enhanced architecture.
- They define the functionality and internal structure of the SNMP managers and agents.
 - An in depth description of entities can be found in RFC 3411.

```
Router1(config)#snmp-server commun
  ASCII string  SNMP community string
Router1(config)#snmp-server commun thisisreadwrite
  ro  Read-only access with this community string
  rw  Read-write access with this community string
Router1(config)#snmp-server commun thisisreadwrite rw
  ASCII string  allowing access with this community string by Standard Access
                name
  <1-99>        Standard IP access list allowing access with this community string by Standard Access number
  ipv6          IPv6 Access control list
  view          Restrict this community to a named MIB view
  <cr>
Router1(config)#snmp-server commun thisisreadwrite rw
Router1(config)#snmp-server commun thisisreadonly ro
Router1(config)#
Router1(config)#
Router1(config)#
```

- SNMP-v3 defines two entity types, a manager and an agent.
 - Both entity types contain two basic components:
 - ▲ SNMP engine
 - ▲ set of applications.



SNMP Engine Functionalities

- The engine provides the basic services to support the agents component applications.
 - These functions include,
 - ▲ message transmission and reception,
 - ▲ authentication and encryption, and
 - ▲ access control to its managed objects database (MIB).

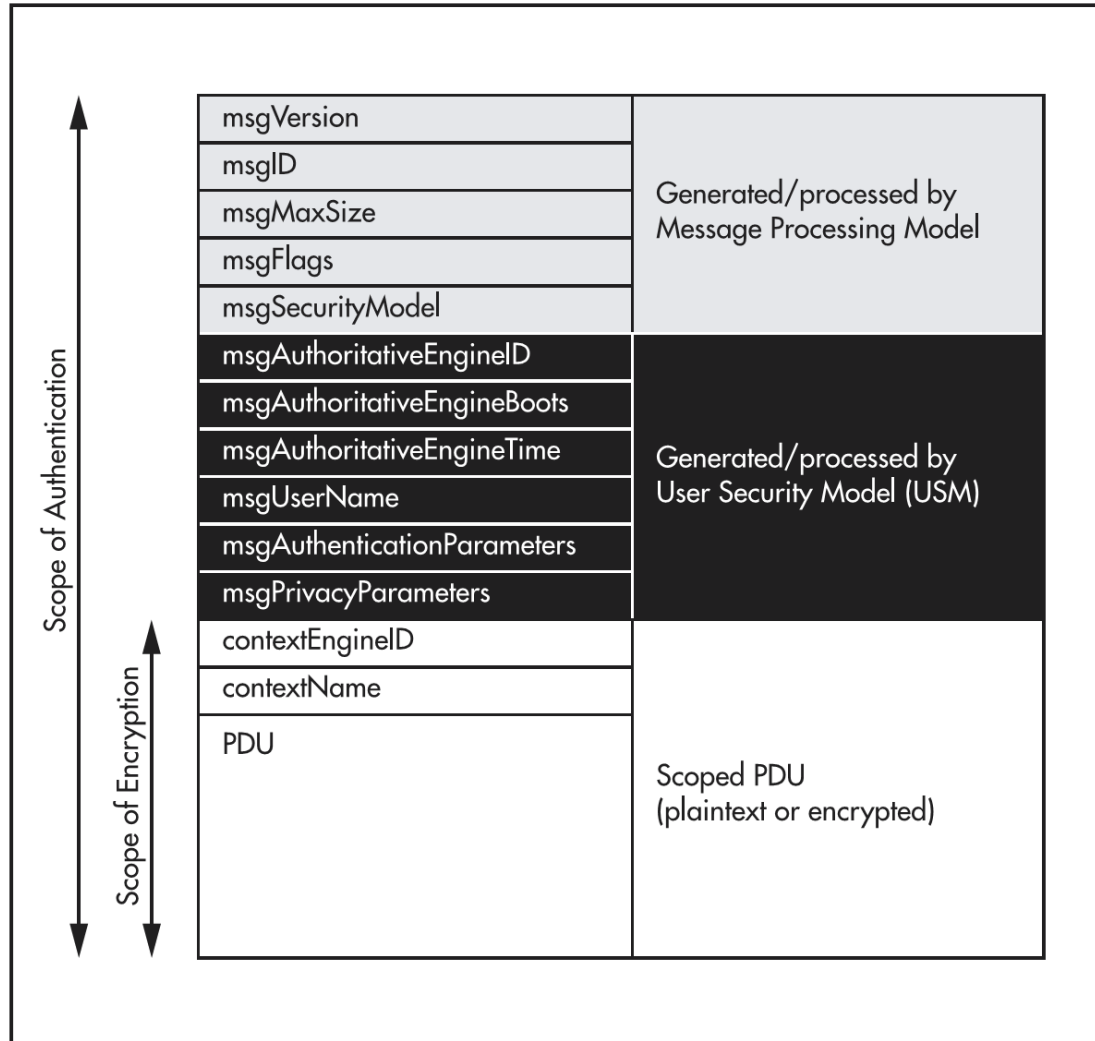
SNMP Engine Components

- The SNMP engine comprises the following components:
 - Dispatcher
 - Message processing Subsystem
 - Security Subsystem
 - Access Control Subsystem
- ▲ Each SNMP engine is identified by an *snmpEngineID* that must be unique within the management system.
- ▲ A one to one association exists between an engine and the entity that contains it.

Entity Applications

- The following applications are defined within the agent applications:
 - Command Generator
 - Notification Receiver
 - Proxy Forwarder
 - Command Responder
 - Notification Originator
 - Other

SNMP-v3 Message Protocol Format



SNMP v3 PDU's

Value	Meaning
msgVersion	Identifies the message format to be SNMPv3.
msgID	An identifier used between SNMP entities to coordinate message requests and responses. Note that a message response takes the msgID value of the
msgMaxSize	Conveys the maximum message size (in octets) supported by the sender of the message. Specified as an integer between 484 and $2^{31}-1$.
msgFlags	A single octet whose last three bits indicate the operational mode for privacy, authentication, and report.
msgSecurityModel	An identifier used to indicate the security mode (i.e. SNMPv1, SNMPv2c or SNMPv3) to be used when processing the message. Note that although only the SNMPv3 identifier is accepted by the router, these earlier version message formats are detected by the <i>msgVersion</i> field and processed appropriately.

Cont...

msgAuthoritativeEngineID	The ID of the authoritative engine that relates to a particular message, i.e. the source engine ID for Traps, Responses and Reports, and the destination engine for Gets, GetNexts, Sets, and Informs.
msgAuthoritativeEngineBoots	A value that represents the number of times the authoritative engine has rebooted since its installation. Its value has the range 1 to $2^{31}-1$.
msgAuthoritativeEngineTime	The number of seconds since the authoritative engine snmpEngineBoots counter was last incremented.
msgUserName	The name of the user (principal) on whose behalf the message is being exchanged.
msgAuthenticationParameters	If the message has been authenticated, this field contains a serialized OCTET STRING representing the first 12 octets of the HMAC-MD5-96 output done over the whole message.

Cont...




msgPrivacyParameters	For encrypted data, this field contains the "salt" used to create the DES encryption Initialisation Vector (IV).
ContextEngineID	Within a particular administrative domain, this field uniquely identifies an SNMP entity that may realize an instance of a context with a particular contextName.
ContextName	A unique name given to a context within a particular SNMP entity.



SNMP-v1 and SNMP-v2c on Router

- The router's implementation of SNMPv1 is based on RFC 1157
 - "A Simple Network Management Protocol (SNMP)", and RFC 1812, "Requirements for IP Version 4 Routers".
- The router's implementation of SNMP v2c is based on the RFCs listed in
 - "Network Management Framework"

- The SNMP agent can be enabled or disabled by using the commands:
 - `enable snmp`
 - `disable snmp`
 - When the SNMP agent is disabled, the agent does not respond to SNMP request messages.
 - The agent is disabled by default.
 - The current state and configuration of the SNMP agent can be displayed by using the command:
 - `show snmp`
- 

SNMP MIB Views for SNMP-v1 and SNMP-v2c

- An SNMP MIB view is an arbitrary subset of objects in the MIB.
- Objects in the view may be from any part of the object name space, and not necessarily the same sub-tree.
- An SNMP community profile is the pairing of an SNMP access mode (read-only or read-write) with the access mode defined by the MIB for each object in the view.
 - For each object in the view, the community profile defines the operations that can be performed on the object.

Community profiles for objects in a MIB view

SNMP Access Mode	Object Access Defined by MIB			
	Read-Only	Read-Write	Write-Only	Not Accessible
Read-Only	get, get-next, trap	get, get-next, trap	<i>None</i>	<i>None</i>
Read-Write	get, get-next, trap	get, get-next, set, trap	get, get-next, set, trap(*)	<i>None</i>

- When an agent receives an SNMP message, it checks the community name encoded in the message.
- If the agent knows the community name, the message is deemed to be authentic and the sending SNMP entity is accepted as a member of the community.

SNMP Communities

- SNMP communities were introduced into SNMP-v1 and retained in version-v2c.
- Although the router's software still supports communities, this is to provide backward compatibility with legacy management systems.
 - For security reasons communities should NOT be used within an SNMPv3 environment.

- An SNMP community is a pairing of an SNMP agent with a set of SNMP application entities.
 - Communities are the main configuration item in the router's implementation of SNMP-v1 and v2c, and
 - are defined in terms of a list of IP addresses which define the SNMP application entities (trap hosts and management stations) in the community.

- An SNMP community is created by using the command:

```
create snmp community=name [access={read|write}]  
    [traphost=ipadd] [manager=ipadd]  
    [open={on|off|yes|no|true|false}] [v1traphost=ipadd]  
    [v2ctraphost=ipadd]
```

- which defines the name of the community (e.g. “public”), and specifies the IP address of a trap host and/or a management station.
 - This command also specifies the version of SNMP received by trap hosts.

- A community can be modified by using the command:

```
set snmp community=name [access={read|write}]  
[open={on|off|yes|no|true|false}]
```

- Community can be destroyed using,

```
destroy snmp community=name
```

- Additional trap hosts and management stations can be added to or removed from a community by using the commands:

```
add snmp community=name [traphost=ipadd] [manager=ipadd]  
[v1traphost=ipadd] [v2ctraphost=ipadd]  
delete snmp community=name [traphost=ipadd] [manager=ipadd]  
[v1traphost=ipadd] [v2ctraphost=ipadd]
```

- An SNMP community, or the generation of traps by the community, can be temporarily enabled or disabled by using the commands:

```
disable snmp community=name trap  
enable snmp community=name trap
```

- Information about the configuration of SNMP communities can be displayed by using the command:

```
show snmp community=name
```



Thanks