STRUCTURE OF MANAGEMENT INFORMATION IN SNMP

Oancea Daniel

Abstract. In this article we present the Structure of Management Information from SNMP, for all three versions of SNMP, as well as the main differences between them. In the first part, two version of SNMP are presented: version 1, version 2; and in the last part the third version, that uses a security model for information protection, is presented.

Keywords:

1. INTRODUCTION

Appeared as a necessity related to the growth of the network and to the higher and higher complexity of their management, SNMP have been imposed in the Internet World as being able to find and resolve the occurring network problems.

The Structure of Management of Information (SMI) from SNMPv1 defines the rules for describing the management information, using Abstract Syntax Notation One (ASN.1). The SMI SNMPv1 specifies that all managed objects have a subset of Abstract Syntax Notation One (ASN.1) data types associated with them.

Three ASN.1 data types are required: name, syntax and encoding. The name serves as the object identifier (object ID). The Syntax defines the data type of the object (for example, integer or string). The encoding field describes how the information associated with the managed object is formatted to be transmitted in the network.

2. SNMP AND ASN.1 DATA TYPES

The SMI SNMPv1 specifies the use of a number of SMI-specific data types, which are divided into two categories: simple data types and application-wide data types.

Three simple data types are defined in the SNMPv1 SMI, all of which are unique values: integers, octet strings and object IDs.

The integer data type is a signed integer in the range of -2,147,483,648 to 2,147,483,647.

Octet strings are ordered sequences of 0 to 65,535 octets.

Object IDs come from the set of all object identifiers allocated according to the rules specified in ASN.1

Seven application-wide data types exist in the SNMPv1 SMI: network addresses, counters, gauges, timeticks, opaques, integers, and unsigned integers. Network addresses represent an address from a particular protocol family.

3. SNMP MIB TABLES

The SNMPv1 SMI defines structured tables that are used to group the instances of a tabular object (an object that contains multiple variables). Tables are composed of zero or more rows, which are indexed in a way that allows SNMP to retrieve or alter an entire row with a simple **Get**, **GetNext**, or **Set** command.

This paper was recommended for publication by Luminița DUMITRIU

4. SNMP PROTOCOL OPERATIONS

SNMP is in fact a simple request/response protocol. The network-management system issues a request, and managed devices, residing in the network, process the information and return responses. This behavior is implemented using one of four protocol operations: Get, GetNext, Set, and Trap.

In the case **Get** command is used, the Network Management System (NMS) retrieves the value of one or more object instances from an agent. If the agent responding to the Get operation cannot provide values for all the object instances in a list, it does not provide any values.

The **GetNext** operation is used by the NMS to retrieve the value of the next object instance in a table or a list within an agent.

The **Set** operation is used by the NMS to set the values of object instances within an agent.

The **Trap** operation is used by agents to asynchronously inform the NMS of a significant event.

5. SNMPV 2

SNMP version 2 (SNMPv2) is an evolution of the initial version, SNMPv1. Originally, SNMPv2 was published as a set of proposed Internet standards in 1993; currently, it is a draft standard. Similarly to SNMPv1, SNMPv2 functions within the specifications of the Structure of Management Information (SMI). Theoretically, SNMPv2 offers a number of improvements to SNMPv1, including additional protocol operations. It makes certain additions and enhancements to the SNMPv1 SMI-specific data types, such as including bit strings, network addresses, and counters. Representatives for SNMPv2 are bit strings; they comprise zero or more named bits that specify a value. In the network address case version1 supports only 32-bit IP addresses, while the second version can support other types of addresses as well. There are also 64 bit counters. In SNMPv1, a 32-bit counter size is specified. In SNMPv2, 32-bit and 64-bit counters are defined.

6. SNMP MANAGEMENT

SNMP is a distributed-management protocol. A system can operate exclusively either as an NMS or as an agent, or it can perform the functions of both. When a system operates as both a NMS and an agent, another NMS might require that the system query manages devices and provides a summary of the information learned, or that it reports locally stored management information.

7. SNMP SECURITY

In what concerns the security, the first two versions of in SNMP do not provide authentication and support for such a capability. Therefore there it is vulnerable in security. These include masquerading occurrences, modification of information, message sequence, timing modifications and disclosure.

Masquerading consists of an unauthorized entity attempting to perform management operations by assuming the identity of an authorized management entity.

Modification of information involves an unauthorized entity attempting to alter a message generated by an authorized entity so that the message results in unauthorized accounting management or configuration management operations.

Message sequence and timing modifications occur when an unauthorized entity reorders, delays, or copies and later replays a message generated by an authorized entity.

Disclosure is involved when unauthorized entity extracts values stored in managed objects, or learns of noticeable events by monitoring exchanges between managers and agents.

Because SNMP does not implement authentication, many vendors do not implement Set operations

We can resume that SNMPv2 provides extended capabilities: message format transmitted over the network and a new set of operations protocol. SNMPv2 uses a different header format message and Protocol Data Unit beside SNMPv1. SNMPv2 also defines two new protocol operations: GetBulk and Inform.

The **GetBulk** operation is used by the NMS to efficiently retrieve large blocks of data, such as multiple rows in a table. GetBulk fills a response message with as much of the requested data as will fit.

The **Inform** operation allows one NMS to send trap information to another NMS and to receive a response.

Can these two versions coexist in a network? Furthermore, RFC 1908 defines two possible SNMPv1/v2 coexistence strategies: proxy agents and bilingual network-management systems.

7.1. Proxy Agents

A SNMPv2 agent can act as a proxy agent on behalf of SNMPv1 managed devices, as follows:

- When an SNMP NMS issues a command intended for a SNMPv1 agent, the

NMS(Network Management System) sends the SNMP message to the SNMv2 proxy agent.

- The proxy agent forwards the Get, GetNext, and Set messages to the SNMPv1 agent unchanged, while GetBulk messages are converted by the proxy agent to GetNext messages and are forwarded to the SNMPv1 agent.
- The proxy agent maps SNMPv1 trap messages to SNMPv2 trap messages and then forwards them to the NMS

.

7.2. Bilingual Network-Management Network

If bilingual Network Management System is used, its management applications must contact an agent. The NMS then examines the information stored in a local database to determine whether the agent supports SNMPv1 or SNMPv2. Based on the information in the database, the NMS communicates with the agent using the appropriate version of SNMP.

8. SNMPV3

The Security model, used by the SNMPv3, is the User Security Model (USM) it is specified defined by RFC2574. The USM provides authentication and privacy services for SNMP and it is designed to secure against the following threats: modification of information, masquerading, message stream modification and disclosure.

A SNMP community is a relationship between a SNMP agent and a set of SNMP managers that defines authentication, access control and proxy characteristics. The managed system establishes one community for each desired combination of authentication, access control and proxy characteristics. Each community is given a unique (within the agent) community name, and the management stations within that community are provided with and must employ the community name in all Get and Set operations.

Still, the community name may be read from a SNMPv1/SNMPv2 message using a packet capture application; v1/v2 messages are not encrypted.

SNMPv3 message consists of three sections msgGlobalData (header), msgSecurityParameters and msgData (scopedPdu). The middle section depends on the security model in use. SNMPv3 defines USM as a security model of choice but vendors are free to implements their own security models. The last field in the message header is an integer that represents the security model used for this message. If USM is used this field contains the value 3. The header also contains a flag field that carry information about the security level applied to the message. Possible combinations are: no security applied, the message is authenticated, the message is authenticated and encrypted.

The msgSecurityParameters field consists of six fields that ensure protection against the security threats listed above. Note that USM does not prevent Denial of Service and Traffic Analysis attacks. Not all fields in this section are used in every exchange. Which fields are used depends on the security level. Two fields that are always used are msgAuthoritativeEngineID (the snmpEngineID of the authoritative SNMP engine involved in the exchange of this message) and msgUserName (the user on whose behalf the message is being exchanged).

The Principal field identifies the entity on whose behalf services are provided or processing takes place. A principal can be, among other things, an individual acting in a particular role; a set of individuals, each of them acting in a particular role; an application or a set of applications; and combinations thereof. With other words it’s possible to consider an application or an application set.

SecurityName is a human readable string representing a principal. It has a model independent format, and can be used outside a particular Security Model. Model-depend security ID is the model-specific representation of a securityName within a particular Security Model. Model-dependent security IDs may or may not be human readable, and have a model-dependent syntax.

8.1. Message Exchange

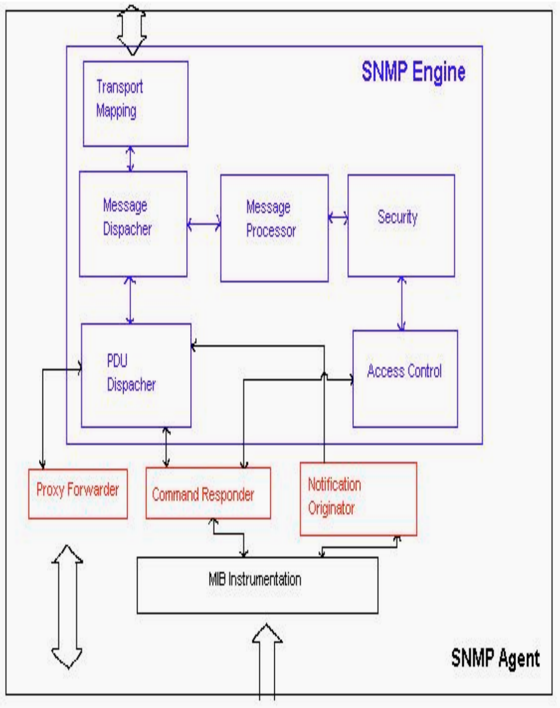
Let’s see how message exchange is performed. In the first place, as mentioned above, even if no security is applied to the message, two fields from the msgSecurityParameters section must be populated: authoritative engine ID and user name.

A manager that wants to retrieve/modify an object in the agent has to know that agent’s engine id and has to insert into the message the appropriate user name.

If we want to assure the authenticity of the communication between the manager and the agent (the agent wants to be sure that the message comes from the manager that it claims, and vice versa, the manager wants to be sure that the response really comes from the targeted agent), then the sender has to calculate the message digest (using MD5 or SHA hash functions) and insert it into msgAuthenticationParameters field of the msgSecurityParameters. The receiver removes the digest from the message, stores it into a temporary location, fills in the empty space with zero octets and calculates the message digest. If the calculated digest is the same as the received digest then the message is authentic; otherwise, a possible impostor is trying to illegally perform the operation.

In order to prevent replay attack, USM uses timeliness mechanism. The idea is simple. The authoritative engine maintains two objects, snmpEngineBoots and snmpEngineTime that refer the local time. Non-authoritative engines must remain loosely synchronized with each authoritative SNMP engine with which it communicates. For that purpose non-authoritative engines keep a local copy of three variables per remote engine ID: snmpEngineBoots, snmpEngineTime and latestReceivedEngineTime.

8.2. SNMP Entities



The following three types of applications have been defined (see the figure above):

- *Command Generator Application* that initiates the protocol operations: Get, GetNext, GetBulk and Set;
- *Command Response Application* where SNMP Get, GetNext, GetBulk and/or Set requests are received;
- *Notification Generator Application* which monitors the system events and generates Trap and/or Inform messages based on these;
- *Notification Receiver Application* that behaves like a notification messages listener and generates response messages when a messages containing an Inform PDU is received and
- *Proxy Forwarder Application* that forwards SNMP messages.

9. CONCLUSIONS

There are several differences the between Structure Management Information data types used in the version one and two. In what concerns protocol operation: the Get, GetNext, and Set operations used in SNMPv1 are exactly the same as those used in SNMPv2. However, SNMPv2 adds and enhances some protocol operations. The SNMPv2 Trap operation, for example, serves the same function as that used in SNMPv1, but it uses a different message format and is designed to replace the SNMPv1 Trap. SNMPv2 also defines two new protocol operations: GetBulk and Inform. In SNMPv2, if the agent responding to GetBulk operations cannot provide values for all the variables in a list, it provides partial results.

SNMPv3 is much more complex than SNMPv1 or SNMPv2c. Complexity comes from the security mechanism used in SNMPv3. SNMPv1 and SNMPv2c provide only a primitive and limited capability for security based on community names. It is obvious that community based security does not provide protection against the threats.

Encryption without authentication is not valid. msgSecurityParameters section consists of six fields that ensure protection against security.

We intend to continue our studies on a new security model, USM-based, enhanced with powerful crypting algorithms.

Two major algorithms are considered interesting as additional data coding protocols, namely, AES and Triple DES. The Internet Engineering Task Force is attempting to implement AES as an additional coding solution.

10. REFERENCES

Internetworking Technologies Handbook Reference: SNMPv2 Message Format. *Simple Network Management Protocol SNMP Hands on SNMPv3 Tutorial & Demo Manual* – NuDesign Team Inc.

John Larmouth (1999) *ASN.1 Complete by Prof. John Larmouth* –Open Systems Solutions

Michael Pott (1997) *.HP OpenView IT/Administration* HP OpenView

Serban Simu–*Managementul de retea, de la simplu la complex* Computer Press Agora