
USM

The User-based Security Model (USM) and the View Access Control Model (VACM) together detail the security enhancements added with SNMPv3 . Let's start with the USM.

The Basics

We need to get some terminology out of the way before we can look at the USM in any detail:

`snmpEngineID`

This is an unambiguous identifier for an SNMP engine as well as the SNMP entity that corresponds to the engine. The syntax for this identifier is `OctetString` and it cannot be zero length. Most SNMPv3 applications allow for the user to input a value for `snmpEngineID`. If one is not specified, the value is computed using a combination of enterprise ID and IP or MAC address.

`snmpEngineBoots`

A count of the number of times an SNMP engine has rebooted.

`snmpEngineTime`

The number of seconds since the `snmpEngineBoots` counter was last incremented.

`snmpSecurityLevel`

There are three security levels. The first is no authentication or privacy (noAuthNoPriv). Note that if this mode is used, a `securityName` is still required. The second is authentication and no privacy (authNoPriv). The third and final one is authentication and privacy (authPriv). While you can have authentication without privacy, you cannot have privacy without authentication.

Authoritative SNMP engine

A nonauthoritative engine must discover the `snmpEngineId` of the authoritative engine with which it communicates. The rules for designating the authoritative engine are as follows: if the SNMP message requires a response (get, getnext, getbulk, set, or inform), the receiver of these messages is authoritative. If the message does not require a response (trap or report), the sender of the message is authoritative. Generally, an SNMP agent is authoritative and an NMS is nonauthoritative.

An SNMPv3 message (packet) format has the following fields:

`msgVersion`

The SNMP version of the message, set to 3.

`msgID`

The `msgID` is used between a manager and agent to coordinate request and response messages.

`msgMaxSize`

The `msgMaxSize` is the maximum message size supported by a sender of an SNMP message.

`msgFlags`

`msgFlags` is an 8-bit value that specifies whether a report PDU is to be generated, whether privacy is used, and whether authentication is used.

`msgSecurityModel`

Specifies which security model was used by the sender of the message. Current values are 1, 2, and 3 for SNMPv1, SNMPv2c, and SNMPv3, respectively.

`msgSecurityParameters`

`msgSecurityParameters` contains security-specific information.

`contextEngineID`

Uniquely identifies an SNMP entity. An SNMP entity is the combination of an SNMP engine and SNMP applications. This is discussed in the section on VACM.

`contextName`

`contextName` identifies a particular context within an SNMP engine.

`scopedPDU`

A block of data made up of a `contextEngineID`, `contextName`, and SNMP PDU.

The `msgSecurityParameters` in an SNMPv3 message are as follows:

`msgAuthoritativeEngineID`

The `snmpEngineID` of the authoritative engine.

`msgAuthoritativeEngineBoots`

The `snmpEngineBoots` of the authoritative engine.

`msgAuthoritativeEngineTime`

The `snmpEngineTime` of the authoritative engine.

`msgUserName`

The user who may be authenticating and encrypting the message.

`msgAuthenticationParameters`

This value is null if no authentication is used. Otherwise, the field contains the computer HMAC message digest for the message. Currently the RFC specifies that MD5 and SHA must be used.

`msgPrivacyParameters`

This value is null if no encryption is used. Otherwise, this field is used to form the initial value of the Cipher Block Chaining mode of the Data Encryption Standard (CBC-DES) algorithm.

Figure 3-2 [*] shows the entire SNMPv3 message.

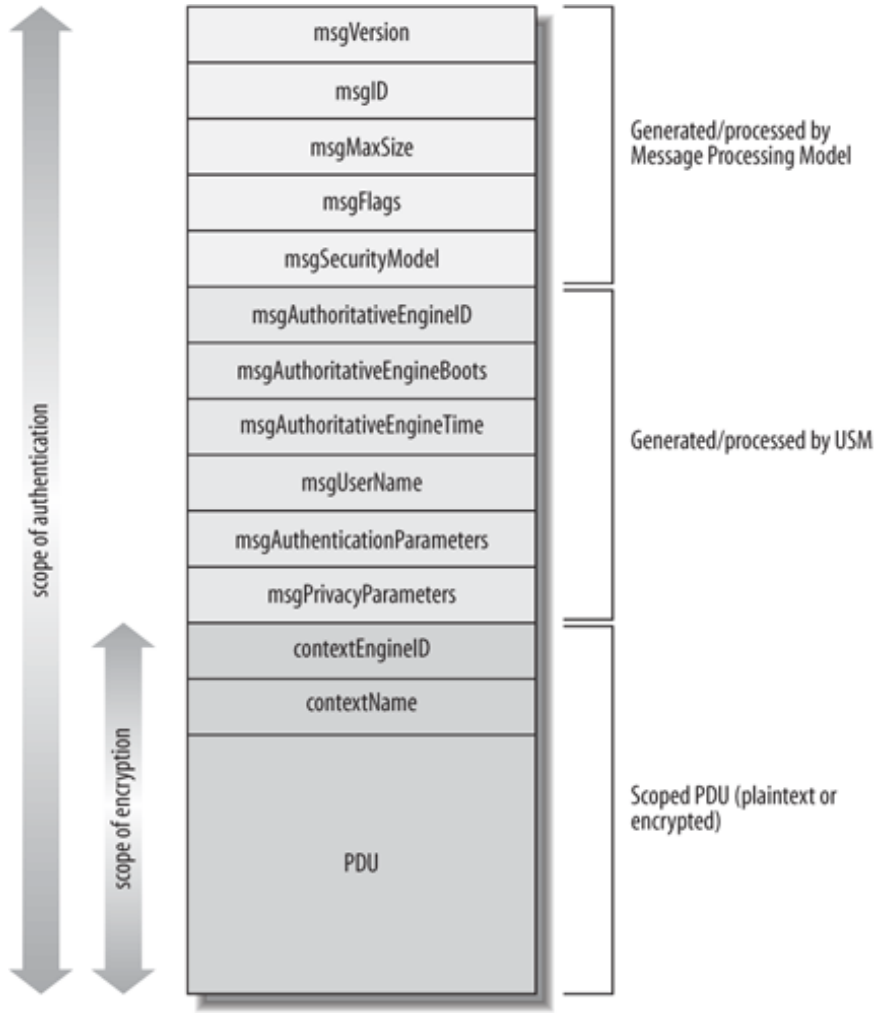

Figure 3-2. SNMPv3 message format

Discovery

The USM requires that the `msgSecurityParameters` contain the `snmpEngineID`, `snmpEngineBoots`, and `snmpEngineTime` of the authoritative engine. Before any get, getnext, or set operation can be used, the nonauthoritative engine must obtain these values from the authoritative engine. A discovery process is used to obtain this information.

USM Timeliness

Once a nonauthoritative engine has learned the value of `snmpEngineBoots` and `snmpEngineTime`, it must maintain its own local notion of what these values are supposed to be. The nonauthoritative engine increments the learned `snmpEngineTime` every second so that it stays up-to-date with the authoritative engine's own notion of `snmpEngineTime`. If `snmpEngineTime` rolls over, `snmpEngineBoots` must be incremented. The USM Timeliness Module is intended to help thwart message delay or replay.

Authentication

MD5, or Message Digest 5, and SHA1, or Secure Hash Algorithm 1, are used for authenticating SNMPv3 messages. MD5 creates a digest of 128 bits and SHA1 creates a digest of 160 bits. Both digests are fixed in size and cannot be used solely for authentication . The keyed Hashing for Message Authentication (HMAC) algorithm is used in conjunction with MD5 and SHA1 to compute message digests. An authentication passphrase or secret key is appended to the data before the digests are computed. The secret key must be known by both the sender and the receiver. The RFCs specify that this passphrase must be at least eight characters long.

Privacy

Encryption of SNMP data is accomplished by using the CBC-DES algorithm. As with authentication, a secret key or passphrase must be known by the sender and receiver and used in the encryption process. A USM User Table is used to store the passphrase and other details transmitted with the packet in the `msgPrivacyParameters`.

USM User Table

Every entity maintains a User Table that stores all the users who have access to the system via SNMP. The User Table includes the following elements :

Username

A textual username. Sometimes referred to as a security name.

Authentication protocol

Details what, if any, authentication protocol is to be used. Valid values include `usmNoAuthProtocol`, `usmHMACMD5AuthProtocol`, and `usmHMACSHAAuthProtocol`.

Authentication key

The passphrase used for authentication. Must be at least eight characters long.

Privacy protocol

Details what, if any, privacy protocol is to be used. Valid values include `usmNoPrivProtocol` and `usmDESPrivProtocol`.

Privacy key

The passphrase used for privacy. Must be at least eight characters long.

`usmUserSpinLock`

The `usmUserSpinLock` is an advisory lock that allows for the coordination of multiple attempts to modify the User Table.

Localized Keys and Changing Keys

A localized key allows for the same passphrase to be used by a single user on many different engines. It keeps an operator from having to remember a different passphrase for each SNMP engine he must interact with. The `KeyChange` type allows for users to change their keys securely.