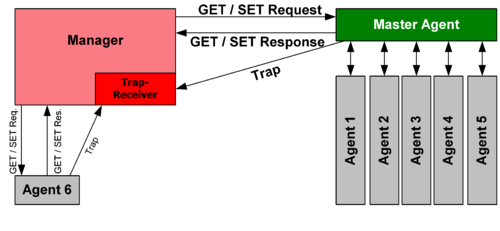
**Simple Network Management Protocol** (**SNMP**) is an [Internet Standard](https://en.wikipedia.org/wiki/Internet_Standard) protocol for collecting and organizing information about managed devices on [IP](https://en.wikipedia.org/wiki/Internet_Protocol) networks and for modifying that information to change device behavior. Devices that typically support SNMP include cable modems, routers, switches, servers, workstations, printers, and more.[[1]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-ESNMP-1)

SNMP is widely used in [network management](https://en.wikipedia.org/wiki/Network_management) for [network monitoring](https://en.wikipedia.org/wiki/Network_monitoring). SNMP exposes management data in the form of variables on the managed systems organized in a [management information base](https://en.wikipedia.org/wiki/Management_information_base) (MIB) which describe the system status and configuration. These variables can then be remotely queried (and, in some circumstances, manipulated) by managing applications.

Three significant versions of SNMP have been developed and deployed. SNMPv1 is the original version of the protocol. More recent versions, SNMPv2c and SNMPv3, feature improvements in performance, flexibility and security.

SNMP is a component of the [Internet Protocol Suite](https://en.wikipedia.org/wiki/Internet_Protocol_Suite) as defined by the [Internet Engineering Task Force](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force) (IETF). It consists of a set of [standards](https://en.wikipedia.org/wiki/Technical_standard) for network management, including an [application layer](https://en.wikipedia.org/wiki/Application_layer) protocol, a database [schema](https://en.wikipedia.org/wiki/Logical_schema), and a set of [data objects](https://en.wikipedia.org/wiki/Data_object).[[2]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-RFC_3411-2)

**Overview and basic concepts**

[](https://en.wikipedia.org/wiki/File:SNMP_communication_principles_diagram.PNG)

Principle of SNMP Communication

In typical uses of SNMP, one or more administrative computers called *managers* have the task of monitoring or managing a group of hosts or devices on a [computer network](https://en.wikipedia.org/wiki/Computer_network). Each managed system executes a software component called an *agent* which reports information via SNMP to the manager.

An SNMP-managed network consists of three key components:

* Managed devices
* [Agent](https://en.wikipedia.org/wiki/Software_agent) – software which runs on managed devices
* [Network management station](https://en.wikipedia.org/wiki/Network_management_station) (NMS) – software which runs on the manager

A *managed device* is a network node that implements an SNMP interface that allows unidirectional (read-only) or bidirectional (read and write) access to node-specific information. Managed devices exchange node-specific information with the NMSs. Sometimes called network elements, the managed devices can be any type of device, including, but not limited to, [routers](https://en.wikipedia.org/wiki/Router_(computing)), [access servers](https://en.wikipedia.org/wiki/Network_access_server), [switches](https://en.wikipedia.org/wiki/Network_switch), [cable modems](https://en.wikipedia.org/wiki/Cable_modems), [bridges](https://en.wikipedia.org/wiki/Network_bridge), [hubs](https://en.wikipedia.org/wiki/Ethernet_hub), [IP telephones](https://en.wikipedia.org/wiki/IP_phone), [IP video cameras](https://en.wikipedia.org/wiki/IP_camera), computer [hosts](https://en.wikipedia.org/wiki/Host_(network)), and [printers](https://en.wikipedia.org/wiki/Computer_printer).

An *agent* is a network-management software module that resides on a managed device. An agent has local knowledge of management information and translates that information to or from an SNMP-specific form.

A *network management station* executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. One or more NMSs may exist on any managed network.

**Management information base**

Main article: [Management information base](https://en.wikipedia.org/wiki/Management_information_base)

SNMP agents expose management data on the managed systems as variables. The protocol also permits active management tasks, such as configuration changes, through remote modification of these variables. The variables accessible via SNMP are organized in hierarchies. SNMP itself does not define which variables a managed system should offer. Rather, SNMP uses an extensible design which allows applications to define their own hierarchies. These hierarchies are described as a [management information base](https://en.wikipedia.org/wiki/Management_information_base) (MIB). MIBs describe the structure of the management data of a device subsystem; they use a [hierarchical namespace](https://en.wikipedia.org/wiki/Hierarchical_namespace) containing [object identifiers](https://en.wikipedia.org/wiki/Object_identifier) (OID). Each OID identifies a variable that can be read or set via SNMP. MIBs use the notation defined by [Structure of Management Information](https://en.wikipedia.org/wiki/Structure_of_Management_Information) Version 2.0 (SMIv2, [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2578](https://tools.ietf.org/html/rfc2578)), a subset of [ASN.1](https://en.wikipedia.org/wiki/ASN.1).

**Protocol details**

SNMP operates in the [application layer](https://en.wikipedia.org/wiki/Application_layer) of the [Internet protocol suite](https://en.wikipedia.org/wiki/Internet_protocol_suite). All SNMP messages are transported via [User Datagram Protocol](https://en.wikipedia.org/wiki/User_Datagram_Protocol) (UDP). The SNMP agent receives requests on [UDP port](https://en.wikipedia.org/wiki/UDP_port) 161. The manager may send requests from any available source port to port 161 in the agent. The agent response is sent back to the source port on the manager. The manager receives notifications ([*Traps*](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#Trap) and [*InformRequests*](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#InformRequest)) on port 162. The agent may generate notifications from any available port. When used with [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security) or [Datagram Transport Layer Security](https://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security), requests are received on port 10161 and notifications are sent to port 10162.[[3]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-3)

SNMPv1 specifies five core [protocol data units](https://en.wikipedia.org/wiki/Protocol_data_unit) (PDUs). Two other PDUs, *GetBulkRequest* and *InformRequest* were added in SNMPv2 and the *Report* PDU was added in SNMPv3. All SNMP PDUs are constructed as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IP header | UDP header | version | community | PDU-type | request-id | error-status | error-index | variable bindings |

The seven SNMP PDU types as identified by the *PDU-type* field are as follows:

GetRequest

A manager-to-agent request to retrieve the value of a variable or list of variables. Desired variables are specified in variable bindings (the value field is not used). Retrieval of the specified variable values is to be done as an [atomic operation](https://en.wikipedia.org/wiki/Atomic_operation) by the agent. A *Response* with current values is returned.

SetRequest

A manager-to-agent request to change the value of a variable or list of variables. Variable bindings are specified in the body of the request. Changes to all specified variables are to be made as an atomic operation by the agent. A *Response* with (current) new values for the variables is returned.

GetNextRequest

A manager-to-agent request to discover available variables and their values. Returns a *Response* with variable binding for the [lexicographically next](https://en.wikipedia.org/wiki/Lexicographical_order) variable in the MIB. The entire MIB of an agent can be walked by iterative application of *GetNextRequest* starting at OID 0. Rows of a table can be read by specifying column OIDs in the variable bindings of the request.

GetBulkRequest

A manager-to-agent request for multiple iterations of *GetNextRequest*. An optimized version of *GetNextRequest*. Returns a *Response* with multiple variable bindings walked from the variable binding or bindings in the request. PDU specific *non-repeaters* and *max-repetitions* fields are used to control response behavior. *GetBulkRequest* was introduced in SNMPv2.

Response

Returns variable bindings and acknowledgement from agent to manager for *GetRequest*, *SetRequest*, *GetNextRequest*, *GetBulkRequest* and *InformRequest*. Error reporting is provided by *error-status* and *error-index* fields. Although it was used as a response to both gets and sets, this PDU was called *GetResponse* in SNMPv1.

Trap

Asynchronous notification from agent to manager. While in other SNMP communication, the manager actively requests information from the agent, these are PDUs that are sent from the agent to the manager without being explicitly requested. SNMP [traps](https://en.wikipedia.org/wiki/Trap_(computing)) enable an agent to notify the management station of significant events by way of an unsolicited SNMP message. Trap PDUs include current *sysUpTime* value, an OID identifying the type of trap and optional variable bindings. Destination addressing for traps is determined in an application-specific manner typically through trap configuration variables in the MIB. The format of the trap message was changed in SNMPv2 and the PDU was renamed *SNMPv2-Trap*.

InformRequest

Acknowledged asynchronous notification. This PDU was introduced in SNMPv2 and was originally defined as *manager to manager* communication.[[4]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-4) Later implementations have loosened the original definition to allow *agent to manager* communications.[[5]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-5)[[6]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-cisco_a-6)[[7]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-7) Manager-to-manager notifications were already possible in SNMPv1 using a *Trap*, but as SNMP commonly runs over UDP where delivery is not assured and dropped packets are not reported, delivery of a *Trap* was not guaranteed. *InformRequest* fixes this as an acknowledgement is returned on receipt.[[6]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-cisco_a-6)

[RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1157](https://tools.ietf.org/html/rfc1157) specifies that an SNMP implementation must accept a message of at least 484 bytes in length. In practice SNMP implementations accept longer messages.[[8]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Tipton-8):1870 If implemented correctly, an SNMP message is discarded if the decoding of the message fails and thus malformed SNMP requests are ignored. A successfully decoded SNMP request is then authenticated using the community string. If the authentication fails, a trap is generated indicating an authentication failure and the message is dropped.[[8]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Tipton-8):1871

SNMPv1 and SNMPv2 use *communities* to establish trust between managers and agents. Most agents support three community names, one each for read-only, read-write and trap. These three community strings control different types of activities. The read-only community applies to *get* requests. The read-write community string applies to *set* requests. The trap community string applies to receipt of *traps*. SNMPv3 also uses community strings, but allows for secure authentication and communication between SNMP manager and agent.[[9]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-9)

**Protocol versions**

In practice, SNMP implementations often support multiple versions: typically SNMPv1, SNMPv2c, and SNMPv3.[[10]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Jacobs-10)[[11]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-11)

**Version 1**

SNMP version 1 (SNMPv1) is the initial implementation of the SNMP protocol. The design of SNMPv1 was done in the 1980s by a group of collaborators who viewed the officially sponsored OSI/IETF/NSF (National Science Foundation) effort (HEMS/CMIS/CMIP) as both unimplementable in the computing platforms of the time as well as potentially unworkable. SNMP was approved based on a belief that it was an interim protocol needed for taking steps towards large scale deployment of the Internet and its commercialization.

The first [Request for Comments](https://en.wikipedia.org/wiki/Request_for_Comments) (RFCs) for SNMP, now known as SNMPv1, appeared in 1988:

* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1065](https://tools.ietf.org/html/rfc1065) — Structure and identification of management information for TCP/IP-based internets
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1066](https://tools.ietf.org/html/rfc1066) — Management information base for network management of TCP/IP-based internets
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1067](https://tools.ietf.org/html/rfc1067) — A simple network management protocol

In 1990, these documents were superseded by:

* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1155](https://tools.ietf.org/html/rfc1155) — Structure and identification of management information for TCP/IP-based internets
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1156](https://tools.ietf.org/html/rfc1156) — Management information base for network management of TCP/IP-based internets
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1157](https://tools.ietf.org/html/rfc1157) — A simple network management protocol

In 1991, [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1156](https://tools.ietf.org/html/rfc1156) (MIB-1) was replaced by the more often used:

* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1213](https://tools.ietf.org/html/rfc1213) — Version 2 of management information base (MIB-2) for network management of TCP/IP-based internets

SNMPv1 is widely used and is the [de facto](https://en.wikipedia.org/wiki/De_facto_standard) network management protocol in the Internet community.[[12]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-12)

SNMPv1 may be carried by [transport layer](https://en.wikipedia.org/wiki/Transport_layer) protocols such as User Datagram Protocol (UDP), Internet Protocol (IP), OSI [Connectionless-mode Network Service](https://en.wikipedia.org/wiki/Connectionless-mode_Network_Service) (CLNS), AppleTalk [Datagram Delivery Protocol](https://en.wikipedia.org/wiki/Datagram_Delivery_Protocol) (DDP), and Novell [Internetwork Packet Exchange](https://en.wikipedia.org/wiki/Internetwork_Packet_Exchange) (IPX).

Version 1 has been criticized for its poor security.[[13]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-aethis-13) The specification does, in fact, allow room for custom authentication to be used, but widely used implementations "support only a trivial authentication service that identifies all SNMP messages as authentic SNMP messages.".[[14]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-14) The security of the messages, therefore, becomes dependent on the security of the channels over which the messages are sent. For example, an organization may consider their internal network to be sufficiently secure that no encryption is necessary for their SNMP messages. In such cases, the "community name", which is transmitted in [cleartext](https://en.wikipedia.org/wiki/Cleartext), tends to be viewed as a de facto password, in spite of the original specification.

**Version 2**

SNMPv2, defined by [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1441](https://tools.ietf.org/html/rfc1441) and [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1452](https://tools.ietf.org/html/rfc1452), revises version 1 and includes improvements in the areas of performance, security and manager-to-manager communications. It introduced *GetBulkRequest*, an alternative to iterative GetNextRequests for retrieving large amounts of management data in a single request. The new party-based security system introduced in SNMPv2, viewed by many as overly complex, was not widely adopted.[[13]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-aethis-13) This version of SNMP reached the Proposed Standard level of maturity, but was deemed obsolete by later versions.[[15]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-rfced-15)

*Community-Based Simple Network Management Protocol version 2*, or *SNMPv2c*, is defined in [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1901](https://tools.ietf.org/html/rfc1901)–[RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1908](https://tools.ietf.org/html/rfc1908). SNMPv2c comprises SNMPv2 *without* the controversial new SNMP v2 security model, using instead the simple community-based security scheme of SNMPv1. This version is one of relatively few standards to meet the IETF's Draft Standard maturity level, and was widely considered the [*de facto*](https://en.wikipedia.org/wiki/De_facto) SNMPv2 standard.[[15]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-rfced-15) It was later restated as part of SNMPv3.[[16]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-16)

*User-Based Simple Network Management Protocol version 2*, or *SNMPv2u*, is defined in [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1909](https://tools.ietf.org/html/rfc1909)–[RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1910](https://tools.ietf.org/html/rfc1910). This is a compromise that attempts to offer greater security than SNMPv1, but without incurring the high complexity of SNMPv2. A variant of this was commercialized as *SNMP v2\**, and the mechanism was eventually adopted as one of two security frameworks in SNMP v3.[[17]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-17)

**64-bit counters**

SNMP version 2 introduces the option for 64-bit data counters. Version 1 was designed only with 32-bit counters which can store integer values from zero to 4.29 billion (precisely 4,294,967,295). A 32-bit version 1 counter cannot store the maximum speed of a 10 gigabit or larger interface, expressed in bits per second. Similarly a 32-bit counter tracking statistics for a 10 gigabit or larger interface can roll over back to zero again in less than one minute, which may be a shorter time interval than a counter is polled to read its current state. This would result in lost or invalid data due to the undetected value rollover, and corruption of trend-tracking data.

The 64-bit version 2 counter can store values from zero to 18.4 quintillion (precisely 18,446,744,073,709,551,615) and so is currently unlikely to experience a counter rollover between polling events. For example, 1.6 [terabit Ethernet](https://en.wikipedia.org/wiki/Terabit_Ethernet) is predicted to become available by 2025. A 64-bit counter incrementing at a rate of 1.6 trillion bits per second would be able to retain information for such an interface without rolling over for 133 days.

**SNMPv1 & SNMPv2c interoperability**

SNMPv2c is incompatible with SNMPv1 in two key areas: message formats and protocol operations. SNMPv2c messages use different header and protocol data unit (PDU) formats than SNMPv1 messages. SNMPv2c also uses two protocol operations that are not specified in SNMPv1. To overcome incompatibility, [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3584](https://tools.ietf.org/html/rfc3584) defines two SNMPv1/v2c coexistence strategies: proxy agents and bilingual network-management systems.

**Proxy agents**

An SNMPv2 agent can act as a proxy agent on behalf of SNMPv1 managed devices. When an SNMPv2 NMS issues a command intended for an SNMPv1 agent it sends it to the SNMPv2 proxy agent instead. The proxy agent forwards Get, GetNext, and Set messages to the SNMPv1 agent unchanged. GetBulk messages are converted by the proxy agent to GetNext messages and then are forwarded to the SNMPv1 agent. Additionally, the proxy agent receives and maps SNMPv1 trap messages to SNMPv2 trap messages and then forwards them to the NMS.

**Bilingual network-management system**

Bilingual SNMPv2 network-management systems support both SNMPv1 and SNMPv2. To support this dual-management environment, a management application examines 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.

**Version 3**

|  |  |
| --- | --- |
|  | This section **is in** [**list**](https://en.wikipedia.org/wiki/MOS:LIST) **format, but may read better as** [**prose**](https://en.wikipedia.org/wiki/MOS:PROSE). You can help by [converting this section](https://en.wikipedia.org/w/index.php?title=Simple_Network_Management_Protocol&action=edit), if appropriate. [Editing help](https://en.wikipedia.org/wiki/Help:Editing) is available. *(September 2016)* |

Although SNMPv3 makes no changes to the protocol aside from the addition of cryptographic security, it looks very different due to new textual conventions, concepts, and terminology.[[1]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-ESNMP-1) The most visible change was to define a secure version of SNMP, by adding security and remote configuration enhancements to SNMP.[[18]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-simpletime-18) The security aspect is addressed by offering both strong authentication and data encryption for privacy. For the administration aspect, SNMPv3 focuses on two parts, namely notification originators and proxy forwarders. The changes also facilitate remote configuration and administration of the SNMP entities, as well as addressing issues related to the large-scale deployment, accounting, and fault management.

Features and enhancements included:

* Identification of SNMP entities to facilitate communication only between known SNMP entities – Each SNMP entity has an identifier called the SNMPEngineID, and SNMP communication is possible only if an SNMP entity knows the identity of its peer. Traps and Notifications are exceptions to this rule.
* Support for security models – A security model may define the security policy within an administrative domain or an intranet. SNMPv3 contains the specifications for a user-based security model (USM).
* Definition of security goals where the goals of message authentication service include protection against the following:
  + Modification of Information – Protection against some unauthorized SNMP entity altering [in-transit messages](https://en.wikipedia.org/wiki/Data_in_transit) generated by an authorized principal.
  + Masquerade – Protection against attempting management operations not authorized for some principal by assuming the identity of another principal that has the appropriate authorizations.
  + Message stream modification – Protection against messages getting maliciously re-ordered, delayed, or replayed to affect unauthorized management operations.
  + Disclosure – Protection against eavesdropping on the exchanges between SNMP engines.
* Specification for USM – USM consists of the general definition of the following communication mechanisms available:
  + Communication without authentication and privacy (NoAuthNoPriv).
  + Communication with authentication and without privacy (AuthNoPriv).
  + Communication with authentication and privacy (AuthPriv).
* Definition of different authentication and privacy protocols – MD5, SHA and HMAC-SHA-2[[19]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-19) authentication protocols and the CBC\_DES and CFB\_AES\_128 privacy protocols are supported in the USM.
* Definition of a discovery procedure – To find the SNMPEngineID of an SNMP entity for a given transport address and transport endpoint address.
* Definition of the time synchronization procedure – To facilitate authenticated communication between the SNMP entities.
* Definition of the SNMP framework MIB – To facilitate remote configuration and administration of the SNMP entity.
* Definition of the USM MIBs – To facilitate remote configuration and administration of the security module.
* Definition of the view-based access control model (VACM) MIBs – To facilitate remote configuration and administration of the access control module.

Security was one of the biggest weakness of SNMP until v3. Authentication in SNMP Versions 1 and 2 amounts to nothing more than a password (community string) sent in clear text between a manager and agent.[[1]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-ESNMP-1) Each SNMPv3 message contains security parameters which are encoded as an octet string. The meaning of these security parameters depends on the security model being used.[[20]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-20) The security approach in v3 targets:[[21]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-cisco-21)

* Confidentiality – [Encryption](https://en.wikipedia.org/wiki/Encryption) of packets to prevent snooping by an unauthorized source.
* Integrity – [Message integrity](https://en.wikipedia.org/wiki/Data_integrity) to ensure that a packet has not been tampered while in transit including an optional packet replay protection mechanism.
* [Authentication](https://en.wikipedia.org/wiki/Authentication) – to verify that the message is from a valid source.

v3 also defines the USM and VACM, which were later followed by a transport security model (TSM) that provided support for SNMPv3 over SSH and SNMPv3 over TLS and DTLS.

* USM (User-based Security Model) provides authentication and privacy (encryption) functions and operates at the message level.
* VACM (View-based Access Control Model) determines whether a given principal is allowed access to a particular MIB object to perform specific functions and operates at the PDU level.
* TSM (Transport Security Model) provides a method for authenticating and encrypting messages over external security channels. Two transports, SSH and TLS/DTLS, have been defined that make use of the TSM specification.

As of 2004 the [IETF](https://en.wikipedia.org/wiki/IETF) recognizes *Simple Network Management Protocol version 3* as defined by [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3411](https://tools.ietf.org/html/rfc3411)–[RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3418](https://tools.ietf.org/html/rfc3418)[[22]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-snmpv3-22) (also known as STD0062) as the current standard version of SNMP. The [IETF](https://en.wikipedia.org/wiki/IETF) has designated SNMPv3 a full [Internet standard](https://en.wikipedia.org/wiki/Internet_standard),[[23]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-23) the highest [maturity level](https://en.wikipedia.org/wiki/IETF_RFC#Status) for an RFC. It considers earlier versions to be obsolete (designating them variously "Historic" or "Obsolete").[[15]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-rfced-15)

**Implementation issues**

SNMP's powerful write capabilities, which would allow the configuration of network devices, are not being fully utilized by many vendors, partly because of a lack of security in SNMP versions before SNMPv3, and partly because many devices simply are not capable of being configured via individual MIB object changes.

Some SNMP values (especially tabular values) require specific knowledge of table indexing schemes, and these index values are not necessarily consistent across platforms. This can cause correlation issues when fetching information from multiple devices that may not employ the same table indexing scheme (for example fetching disk utilization metrics, where a specific disk identifier is different across platforms.)[[24]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-24)

Some major equipment vendors tend to over-extend their proprietary [command line interface](https://en.wikipedia.org/wiki/Command_line_interface) (CLI) centric configuration and control systems.[[25]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-25)[[*failed verification*](https://en.wikipedia.org/wiki/Wikipedia:Verifiability)]

In February 2002 the [Carnegie Mellon Software Engineering Institute](https://en.wikipedia.org/wiki/Carnegie_Mellon_Software_Engineering_Institute) (CM-SEI) Computer Emergency Response Team Coordination Center (CERT-CC) issued an Advisory on SNMPv1,[[26]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-26) after the [Oulu University Secure Programming Group](https://en.wikipedia.org/wiki/Oulu_University_Secure_Programming_Group) conducted a thorough analysis of SNMP message handling. Most SNMP implementations, regardless which version of the protocol they support, use the same program code for decoding [protocol data units](https://en.wikipedia.org/wiki/Protocol_data_units) (PDU) and problems were identified in this code. Other problems were found with decoding SNMP trap messages received by the SNMP management station or requests received by the SNMP agent on the network device. Many vendors had to issue patches for their SNMP implementations.[[8]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Tipton-8):1875

**SNMP security implications**

**Using SNMP to attack a network**

Because SNMP is designed to allow administrators to monitor and configure network devices remotely it can also be used to penetrate a [local area network](https://en.wikipedia.org/wiki/Local_area_network) (LAN). If SNMP is not used in a network it should be turned off, because aside from creating a vulnerability it will consume available network bandwidth and needlessly use [CPU](https://en.wikipedia.org/wiki/CPU) cycles. A significant number of software tools can scan the entire network over SNMP, therefore mistakes in the configuration of the read-write mode can make a network susceptible to attacks.[[27]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Mason-27):52

In 2001 [Cisco](https://en.wikipedia.org/wiki/Cisco) released information that even in read mode the SNMP implementation of [Cisco IOS](https://en.wikipedia.org/wiki/Cisco_IOS) 11.0 and 12.0 (the [operating system](https://en.wikipedia.org/wiki/Operating_system) used by network [switches](https://en.wikipedia.org/wiki/Network_switch) and [routers](https://en.wikipedia.org/wiki/Router_(computing))) is vulnerable to certain [denial of service](https://en.wikipedia.org/wiki/Denial_of_service) attacks. These security issues can be fixed through an IOS upgrade.[[28]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-28) When configuring SNMP read-only mode close attention should be paid to the configuration of the [access control](https://en.wikipedia.org/wiki/Access_control) and from which IP addresses SNMP messages are accepted. If the SNMP servers are identified by their IP, SNMP is only allowed to respond to these IPs and SNMP messages from other IP addresses would be denied. However, [IP address spoofing](https://en.wikipedia.org/wiki/IP_address_spoofing) remains a security concern.[[27]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Mason-27):54

**SNMP authentication**

SNMP is available in different versions 1, 2 and 3, each has its own security issues. SNMP v1 sends passwords in clear-text over the network. Therefore, passwords can be read with [packet sniffing](https://en.wikipedia.org/wiki/Packet_sniffing). SNMP v2 allows [password hashing](https://en.wikipedia.org/wiki/Password_hashing) with [MD5](https://en.wikipedia.org/wiki/MD5), but this has to be configured. Virtually all [network management software](https://en.wikipedia.org/wiki/Network_management_software) support SNMP v1, but not necessarily SNMP v2 or v3. SNMP v2 was specifically developed to provide [data security](https://en.wikipedia.org/wiki/Data_security), that is [authentication](https://en.wikipedia.org/wiki/Authentication), [privacy](https://en.wikipedia.org/wiki/Privacy) and [authorization](https://en.wikipedia.org/wiki/Authorization), but only SNMP version 2c gained the endorsement of the [Internet Engineering Task Force](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force) (IETF), while versions 2u and 2\* failed to gain IETF approval due to security issues. SNMP v3 uses MD5, [Secure Hash Algorithm](https://en.wikipedia.org/wiki/Secure_Hash_Algorithm) (SHA) and keyed algorithms to offer protection against unauthorised data modification and masquerade attacks. If a higher level of security is needed the [Data Encryption Standard](https://en.wikipedia.org/wiki/Data_Encryption_Standard) (DES) can be optionally used in the [cipher block chaining](https://en.wikipedia.org/wiki/Cipher_block_chaining) mode. SNMP v3 is implemented on Cisco IOS since release 12.0(3)T.[[27]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Mason-27):52

SNMPv3 may be subject to [brute force](https://en.wikipedia.org/wiki/Brute_force_attack) and [dictionary attacks](https://en.wikipedia.org/wiki/Dictionary_attack) for guessing the authentication keys, or encryption keys, if these keys are generated from short (weak) passwords, or passwords that can be found in a dictionary. SNMPv3 allows both providing random uniformly distributed cryptographic keys, and generating cryptographic keys from password supplied by user. The risk of guessing authentication strings from hash values transmitted over the network depends on the [Hash function](https://en.wikipedia.org/wiki/Hash_function) used and the length of the hash value.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] SNMPv3 uses the [HMAC](https://en.wikipedia.org/wiki/HMAC)-[SHA-2](https://en.wikipedia.org/wiki/SHA-2) [Authentication Protocol](https://en.wikipedia.org/wiki/Authentication_protocol) for the User-based Security Model (USM).[[29]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-RFC7630-29) A [challenge-response handshake](https://en.wikipedia.org/wiki/Challenge-handshake_authentication_protocol) was not used to improve security. SNMPv3 (like other SNMP protocol versions) is a [stateless protocol](https://en.wikipedia.org/wiki/Stateless_protocol), and it has been designed with minimal amount of interactions between the agent and the manager. Thus introducing a challenge-response handshake for each command would impose a burden on the agent (and possibly on the network itself) that the protocol designers deemed excessive and unacceptable.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

The security deficiencies of all SNMP versions can be mitigated by [IPsec](https://en.wikipedia.org/wiki/IPsec) authentication and confidentiality mechanisms.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] The implementation of SNMP over [Datagram Transport Layer Security](https://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security) (DTLS) is also available.[[10]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Jacobs-10)

**SNMP autodiscovery**

SNMP based network management software send passwords repeatedly during normal operations across the network. Therefore, clear-text passwords are a significant security risk. If SNMP v2 is used, the network administrator should enable password encryption on network devices, that is the SNMP servers running on them.

Many SNMP implementations include a type of [automatic discovery](https://en.wikipedia.org/w/index.php?title=Automatic_discovery&action=edit&redlink=1) where a new network component, such as a switch or router, is discovered and pooled automatically. In SNMPv1 and v2c this is done through a *community string* that is broadcast in clear-text to other devices.[[10]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Jacobs-10) Because of its default configuration on community strings, they are public for read-only access and private for read-write[[8]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Tipton-8):1874 SNMP topped the list of the [SANS Institute](https://en.wikipedia.org/wiki/SANS_Institute)'s Common Default Configuration Issues and was number ten on the SANS Top 10 Most Critical Internet Security Threats for the year 2000.[[30]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-30) System and network administrators frequently do not change these configurations.[[8]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Tipton-8):1874 The community string sent by SNMP over the network is not encrypted. Once the community string is known outside the organisation it could become the target for an attack. To prevent the easy discovery of the community, SNMP should be configured to pass community-name authentication failure traps and the SNMP management device needs to be configured to react to the authentication failure trap.[[27]](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#cite_note-Mason-27):54

SNMPv1 and v2 are vulnerable to [IP spoofing](https://en.wikipedia.org/wiki/IP_spoofing) attacks, whether it runs over TCP or UDP, and is a subject to bypassing device access lists that might have been implemented to restrict SNMP access. SNMPv3 security mechanisms such as USM or TSM prevent a successful attack. It would be pointless to employ SNMPv3 VACM (View-based Access Control) without securing messages with USM or TSM.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**RFC references**

* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1155](https://tools.ietf.org/html/rfc1155) (STD 16) — *Structure and Identification of Management Information for the TCP/IP-based Internets*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1156](https://tools.ietf.org/html/rfc1156) (Historic) — *Management Information Base for Network Management of TCP/IP-based internets*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1157](https://tools.ietf.org/html/rfc1157) (Historic) — *A Simple Network Management Protocol (SNMP)*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1213](https://tools.ietf.org/html/rfc1213) (STD 17) — *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1452](https://tools.ietf.org/html/rfc1452) (Informational) — *Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework* (Obsoleted by [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1908](https://tools.ietf.org/html/rfc1908))
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1901](https://tools.ietf.org/html/rfc1901) (Experimental) — *Introduction to Community-based SNMPv2*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1902](https://tools.ietf.org/html/rfc1902) (Draft Standard) — *Structure of Management Information for SNMPv2* (Obsoleted by [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2578](https://tools.ietf.org/html/rfc2578))
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1908](https://tools.ietf.org/html/rfc1908) (Standards Track) — *Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2570](https://tools.ietf.org/html/rfc2570) (Informational) — *Introduction to Version 3 of the Internet-standard Network Management Framework* (Obsoleted by [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3410](https://tools.ietf.org/html/rfc3410))
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [2578](https://tools.ietf.org/html/rfc2578) (STD 58) — *Structure of Management Information Version 2 (SMIv2)*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3410](https://tools.ietf.org/html/rfc3410) (Informational) — *Introduction and Applicability Statements for Internet Standard Management Framework*
* STD 62
  + [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3411](https://tools.ietf.org/html/rfc3411) — *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*
  + [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3412](https://tools.ietf.org/html/rfc3412) — *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*
  + [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3413](https://tools.ietf.org/html/rfc3413) — *Simple Network Management Protocol (SNMP) Applications*
  + [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3414](https://tools.ietf.org/html/rfc3414) — *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
  + [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3415](https://tools.ietf.org/html/rfc3415) — *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
  + [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3416](https://tools.ietf.org/html/rfc3416) — *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*
  + [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3417](https://tools.ietf.org/html/rfc3417) — *Transport Mappings for the Simple Network Management Protocol (SNMP)*
  + [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3418](https://tools.ietf.org/html/rfc3418) — *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3430](https://tools.ietf.org/html/rfc3430) (Experimental) — *Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3584](https://tools.ietf.org/html/rfc3584) (BCP 74) — *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [3826](https://tools.ietf.org/html/rfc3826) (Proposed) — *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [4789](https://tools.ietf.org/html/rfc4789) (Proposed) — *Simple Network Management Protocol (SNMP) over IEEE 802 Networks*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [5343](https://tools.ietf.org/html/rfc5343) (STD 78) — *Simple Network Management Protocol (SNMP) Context EngineID Discovery*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [5590](https://tools.ietf.org/html/rfc5590) (STD 78) — *Transport Subsystem for the Simple Network Management Protocol (SNMP)*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [5591](https://tools.ietf.org/html/rfc5591) (STD 78) — *Transport Security Model for the Simple Network Management Protocol (SNMP)*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [5592](https://tools.ietf.org/html/rfc5592) (Proposed) — *Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [5608](https://tools.ietf.org/html/rfc5608) (Proposed) — *Remote Authentication Dial-In User Service (RADIUS) Usage for Simple Network Management Protocol (SNMP) Transport Models.*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [6353](https://tools.ietf.org/html/rfc6353) (STD 78) — *Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)*
* [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [7630](https://tools.ietf.org/html/rfc7630) (Standards Track) — *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*

**See also**

* [AgentX](https://en.wikipedia.org/wiki/AgentX), a subagent protocol for SNMP
* [CMIP over TCP/IP](https://en.wikipedia.org/wiki/Common_management_information_protocol) (CMOT)
* [Common management information protocol](https://en.wikipedia.org/wiki/Common_management_information_protocol) (CMIP), a management protocol by ISO/OSI used by telecommunications devices
* [Common management information service](https://en.wikipedia.org/wiki/Common_management_information_service) (CMIS)
* [IEC 62379](https://en.wikipedia.org/wiki/IEC_62379)
* [Net-SNMP](https://en.wikipedia.org/wiki/Net-SNMP), an open source reference implementation of SNMP
* [Netconf](https://en.wikipedia.org/wiki/Netconf), a protocol which is an XML-based configuration protocol for network equipment
* [Network monitoring comparison](https://en.wikipedia.org/wiki/Network_monitoring_comparison)
* [Object identifier](https://en.wikipedia.org/wiki/Object_identifier) (OID)
* [Remote monitoring](https://en.wikipedia.org/wiki/Remote_monitoring) (RMON)
* [Simple Gateway Monitoring Protocol](https://en.wikipedia.org/wiki/Simple_Gateway_Monitoring_Protocol) (SGMP), an obsolete protocol replaced by SNMP
* [SNMP simulator](https://en.wikipedia.org/wiki/SNMP_simulator)