Information Security Management Handbook
Edited by Harold F. Tipton and Micki Krause
Boca Raton: CRC Press LLC, 2003

What's Not So Simple about SNMP?

Chris Hare, CISSP, CISA

The Simple Network Management Protocol, or SNMP, is a defined Internet standard from the Internet Engineering Task Force, as documented in Request for Comment (RFC) 1157. This chapter discusses what SNMP is, how it is used, and the challenges facing network management and security professionals regarding its use.

While several SNMP applications are mentioned in this chapter, no support or recommendation of these applications is made or implied. As with any application, the enterprise must select its SNMP application based upon its individual requirements.

SNMP DEFINED

SNMP is used to monitor network and computer devices around the globe. Simply stated, network managers use SNMP to communicate management information, both status and configuration, between the network management station and the SNMP agents in the network devices.

The protocol is aptly named because despite the intricacies of a network, SNMP itself is very simple. Before examining the architecture, a review of the terminology used is required.

- *Network element:* any device connected to the network, including hosts, gateways, servers, terminal servers, firewalls, routers, switches and active hubs
- *Network management station (or management station):* a computing platform with SNMP management software to monitor and control the network elements; examples of common management stations are HP Openview and CA Unicenter
- *SNMP agent:* a software management agent responsible for performing the network management functions received from the management station

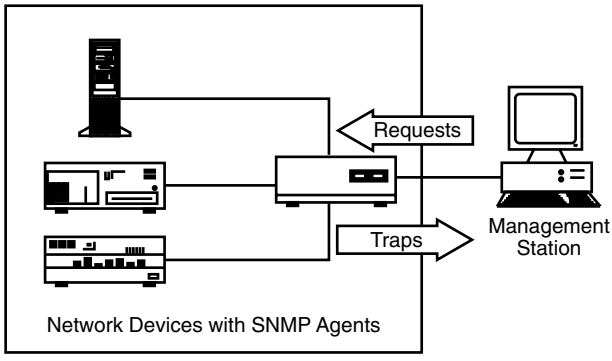

Exhibit 6-1. The SNMP network manager.

- *SNMP request:* a message sent from the management station to the SNMP agent on the network device
- *SNMP trap receiver:* the software on the management station that receives event notification messages from the SNMP agent on the network device
- *Management information base:* a standard method identifying the elements in the SNMP database

A network configured to SNMP for the management of network devices consists of at least one SNMP agent and one management station. The management station is used to configure the network elements and receive SNMP traps from those elements.

Through SNMP, the network manager can monitor the status of the various network elements, make appropriate configuration changes, and respond to alerts received from the network elements (see Exhibit 6-1). As networks increase in size and complexity, a centralized method of monitoring and management is essential. Multiple management stations may exist and be used to compartmentalize the network structure or to regionalize operations of the network.

SNMP can retrieve the configuration information for a given network element in addition to device errors or alerts. Error conditions will vary from one SNMP agent to another but would include network interface failures, system failures, disk space warnings, etc. When the device issues an alert to the management station, network management personnel can investigate to resolve the problem. Access to systems is controlled through knowledge of a community string, which can be compared to a password. Community strings are discussed in more detail later in the chapter, but by themselves should not be considered a form of authentication.

From time to time it is necessary for the management station to send configuration requests to the device. If the correct community string is provided, the device configuration is changed appropriately. Even this simple explanation evidences the value gained from SNMP. An organization can monitor the status of all its equipment and perform remote troubleshooting and configuration management.

THE MANAGEMENT INFORMATION BASE (MIB)

The MIB defines the scope of information available for retrieval or configuration on the network element. There is a standard MIB all devices should support. The manufacturer of the device can also define custom extensions to the device to support additional configuration parameters. The definition of MIB extensions must follow a defined convention for the management stations to understand and interpret the MIB correctly.

The MIB is expressed using the ASN.1 language; and, while important to be aware of, it is not a major concern unless you are specifically designing new elements for the MIB. All MIB objects are defined explicitly in the Internet standard MIB or through a defined naming convention. Using the defined naming convention limits the ability of product vendors to create individual instances of an MIB element for a particular network device. This is important, given the wide number of SNMP capable devices and the relatively small range of monitoring station equipment.

An understanding of the MIB beyond this point is only necessary for network designers who must concern themselves with the actual MIB structure and representations. Suffice to say for this discussion, the MIB components are represented using English identifiers.

SNMP OPERATIONS

All SNMP agents must support both inspection and alteration of the MIB variables. These operations are referred to as *SNMP get* (retrieval and inspection) and *SNMP set* (alteration). The developers of SNMP established only these two operations to minimize the number of essential management functions to support and to avoid the introduction of other imperative management commands. Most network protocols have evolved to support a vast array of potential commands, which must be available in both the client and the server. The File Transfer Protocol (FTP) is a good example of a simple command set that has evolved to include more than 74 commands.

The SNMP management philosophy uses the management station to poll the network elements for appropriate information. SNMP uses *traps* to send messages from the agent running on the monitored system to the monitoring station, which are then used to control the polling. Limiting the

number of messages between the agent and the monitoring station achieves the goal of simplicity and minimizes the amount of traffic associated with the network management functions.

As mentioned, limiting the number of commands makes implementing the protocol easier: it is not necessary to develop an interface to the operating system, causing a system reboot, or to change the value of variables to force a reboot after a defined time period has elapsed.

The interaction between the SNMP agent and management station occurs through the exchange of protocol messages. Each message has been designed to fit within a single User Datagram Protocol (UDP) packet, thereby minimizing the impact of the management structure on the network.

ADMINISTRATIVE RELATIONSHIPS

The management of network elements requires an SNMP agent on the element itself and on a management station. The grouping of SNMP agents to a management station is called a *community*. The community string is the identifier used to distinguish among communities in the same network. The SNMP RFC specifies an authentic message as one in which the correct community string is provided to the network device from the management station. The authentication scheme consists of the community string and a set of rules to determine if the message is in fact authentic. Finally, the SNMP authentication service describes a function identifying an authentic SNMP message according to the established authentication schemes.

Administrative relationships are called communities, that pair a monitored device with the management station. Through this scheme, administrative relationships can be separated among devices. The agent and management station defined within a community establish the SNMP access policy. Management stations can communicate directly with the agent or, in the event of network design, an SNMP proxy agent. The proxy agent relays communications between the monitored device and the management station.

The use of proxy agents allows communication with all network elements, including modems, multiplexors, and other devices that support different management frameworks. Additional benefits from the proxy agent design include shielding network elements from access policies, which might be complex.

The community string establishes the access policy community to use, and it can be compared to passwords. The community string establishes the password to access the agent in either read-only mode, commonly referred to the public community, or the read-write mode, known as the private community.

SNMP REQUESTS

There are two access modes within SNMP: *read-only* and *read-write*. The command used, the variable, and the community string determine the access mode. Corresponding with the access mode are two community strings, one for each access mode. Access to the variable and the associated action is controlled by:

- If the variable is defined with an access type of *none*, the variable is not available under any circumstances.
- If the variable is defined with an access type of *read-write* or *read-only*, the variable is accessible for the appropriate *get, set,* or *trap* commands.
- If the variable does not have an access type defined, it is available for *get* and *trap* operations.

However, these rules only establish what actions can be performed on the MIB variable. The actual communication between the SNMP agent and the monitoring station follows a defined protocol for message exchange. Each message includes the:

- SNMP version identifier
- Community string
- Protocol data unit (PDU)

The SNMP version identifier establishes the version of SNMP in use — Version 1, 2, or 3. As mentioned previously, the community string determines which community is accessed, either public or private. The PDU contains the actual SNMP trap or request. With the exception of traps, which are reported on UDP port 162, all SNMP requests are received on UDP port 161. RFC 1157 specifies that protocol implementations need not accept messages more than 484 bytes in length, although in practice a longer message length is typically supported.

There are five PDUs supported within SNMP:

1. GetRequest-PDU
2. GetNextRequest-PDU
3. GetResponse-PDU
4. SetRequest-PDU
5. Trap-PDU

When transmitting a valid SNMP request, the PDU must be constructed using the implemented function, the MIB variable in ASN.1 notation. The ASN.1 notation, the source and destination IP addresses, and UDP ports are included along with the community string. Once processed, the resulting request is sent to the receiving system.

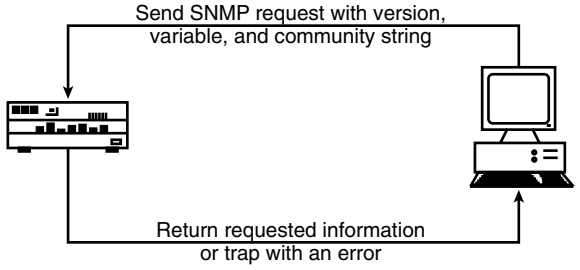

Exhibit 6-2. The SNMP transmission process.

As shown in Exhibit 6-2, the receiving system accepts the request and assembles an ASN.1 object. The message is discarded if the decoding fails. If implemented correctly, this discard function should cause the receiving system to ignore malformed SNMP requests. Similarly, the SNMP version is checked; and if there is a mismatch, the packet is also dropped. The request is then authenticated using the community string. If the authentication fails, a trap may be generated indicating an authentication failure, and the packet is dropped.

If the message is accepted, the object is again parsed to assemble the actual request. If the parse fails, the message is dropped. If the parse is successful, the appropriate SNMP profile is selected using the named community, and the message is processed. Any resulting data is returned to the source address of the request.

THE PROTOCOL DATA UNIT

As mentioned, there are five protocol data units supported. Each is used to implement a specific request within the SNMP agent and management station. Each will be briefly examined to review purpose and functionality.

The *GetRequest* PDU requests information to be retrieved from the remote device. The management station uses the *GetRequest* PDU to make queries of the various network elements. If the MIB variable specified is matched exactly in the network element MIB, the value is returned using the *GetResponse* PDU. We can see the direct results of the *GetRequest* and *GetResponse* messages using the *snmpwalk* command commonly found on Linux systems:

```
[chare@linux chare]$ for host in 1 2 3 4 5
> do
> snmpwalk 192.168.0.$host public system.sysDescr.0
> done
system.sysDescr.0 = Instant Internet version 7.11.2
Timeout: No Response from 192.168.0.2
```

```
system.sysDescr.0 = Linux linux 2.4.9-31 #1 Tue Feb 26
07:11:02 EST 2002 i686
Timeout: No Response from 192.168.0.4
Timeout: No Response from 192.168.0.5
[chare@linux chare]$
```

Despite the existence of a device at all five IP addresses in the above range, only two are configured to provide a response; or perhaps the SNMP community string provided was incorrect.

Note that, on those systems where *snmpwalk* is not installed, the command is available in the net-ucb-cnmp source code available from many network repositories.

The *GetResponse* PDU is the protocol type containing the response to the request issued by the management station. Each *GetRequest* PDU results in a response using *GetResponse*, regardless of the validity of the request.

The *GetNextResponse* PDU is identical in form to the *GetResponse* PDU, except it is used to get additional information from a previous request. Alternatively, table traversals through the MIB are typically done using the *GetNextResponse* PDU. For example, using the *snmpwalk* command, we can traverse the entire table using the command:

```
# snmpwalk localhost public
system.sysDescr.0 = Linux linux 2.4.9-31 #1 Tue Feb 26
07:11:02 EST 2002 i686
system.sysObjectID.0 = OID:
enterprises.ucdavis.ucdSnmpAgent.linux
system.sysUpTime.0 = Timeticks: (4092830521) 473 days,
16:58:25.21
system.sysContact.0 = root@localhost
system.sysName.0 = linux
system.sysLocation.0 = Unknown
system.sysORLastChange.0 = Timeticks: (4) 0:00:00.04
…
<end of snmpwalk output>
```

In our example, no specific MIB variable is requested, which causes all MIB variables and their associated values to be printed. This generates a large amount of output from *snmpwalk*. Each variable is retrieved until there is no additional information to be received.

Aside from the requests to retrieve information, the management station also can set selected variables to new values. This is done using the *SetRequest* PDU. When receiving the *SetRequest* PDU, the receiving station has several valid responses:

- If the named variable cannot be changed, the receiving station returns a *GetResponse* PDU with an error code.
- If the value does not match the named variable type, the receiving station returns a *GetResponse* PDU with a bad value indication.
- If the request exceeds a local size limitation, the receiving station responds with a *GetResponse* PDU with an indication of too big.
- If the named variable cannot be altered and is not covered by the preceding rules, a general error message is returned by the receiving station using the *GetResponse* PDU.

If there are no errors in the request, the receiving station updates the value for the named variable. The typical read-write community is called *private*, and the correct community string must be provided for this access. If the value is changed, the receiving station returns a *GetResponse* PDU with a "No error" indication.

As discussed later in this chapter, if the SNMP read-write community string is the default or set to another well-known value, any user can change MIB parameters and thereby affect the operation of the system.

SNMP TRAPS

SNMP traps are used to send an event back to the monitoring station. The trap is transmitted at the request of the agent and sent to the device specified in the SNMP configuration files. While the use of traps is universal across SNMP implementations, the means by which the SNMP agent determines where to send the trap differs among SNMP agent implementations.

There are several traps available to send to the monitoring station:

- coldStart
- warmStart
- linkDown
- linkUp
- authenticationFailure
- egpNeighborLoss
- enterpriseSpecific

Traps are sent using the PDU, similar to the other message types, previously discussed.

The *coldStart* trap is sent when the system is initialized from a powered-off state and the agent is reinitializing. This trap indicates to the monitoring station that the SNMP implementation may have been or may be altered. The *warmStart* trap is sent when the system restarts, causing the agent to reinitialize. In a *warmStart* trap event, neither the SNMP agent's implementation nor its configuration is altered.

Exhibit 6-3. Router with multiple network interfaces.

Most network management personnel are familiar with the *linkDown* and *linkUp* traps. The *linkDown* trap is generated when a link on the SNMP agent recognizes a failure of one or more of the network links in the SNMP agent's configuration. Similarly, when a communication link is restored, the *linkUp* trap is sent to the monitoring station. In both cases, the trap indicates the network link where the failure or restoration has occurred.

Exhibit 6-3 shows a device, in this case a router, with multiple network interfaces, as seen in a Network Management Station. The failure of the red interface (shown here in black) caused the router to send a *linkDown* trap to the management station, resulting in the change in color for the object. The green objects (shown in white) represent currently operational interfaces.

The *authenticationFailure* trap is generated when the SNMP agent receives a message with the incorrect community string, meaning the attempt to access the SNMP community has failed. When the SNMP agent communicates in an Exterior Gateway Protocol (EGP) relationship, and the peer is no longer reachable, an *egpNeighborLoss* trap is generated to the management station. This trap means routing information available from the EGP peer is no longer available, which may affect other network connectivity.

Finally, the *enterpriseSpecific* trap is generated when the SNMP agent recognizes an *enterpriseSpecific* trap has occurred. This is implementation dependent and includes the specific trap information in the message sent back to the monitoring station.

SNMP SECURITY ISSUES

The preceding brief introduction to SNMP should raise a few issues for the security professional. As mentioned, the default SNMP community

strings are public for read-only access and private for read-write. Most system and network administrators do not change these values. Consequently, any user, authorized or not, can obtain information through SNMP about the device and potentially change or reset values. For example, if the read-write community string is the default, any user can change the device's IP address and take it off the network.

This can have significant consequences, most notably surrounding the availability of the device. It is not typically possible to access enterprise information or system passwords or to gain command line or terminal access using SNMP. Consequently, any changes could result in the monitoring station identifying the device as unavailable, forcing corrective action to restore service.

However, the common SNMP security issues are:

- Well-known default community strings
- Ability to change the configuration information on the system where the SNMP agent is running
- Multiple management stations managing the same device
- Denial-of-service attacks

Many security and network professionals are undoubtedly familiar with the Computer Emergency Response Team (CERT) Advisory CA-2002–03 published in February 2002. While this is of particular interest to the network and security communities today, it should not overshadow the other issues mentioned above because many of the issues in CA-2002-03 are possible due to the other security issues.

Well-Known Community Strings

As mentioned previously, there are two SNMP access polices, read-only and read-write, using the default community strings of public and private, respectively. Many organizations do not change the default community strings. Failing to change the default values means it is possible for an unauthorized person to change the configuration parameters associated with the device.

Consequently, SNMP community strings should be treated as passwords. The better the quality of the password, the less likely an unauthorized person could guess the community string and change the configuration.

Ability to Change SNMP Configuration

On many systems, users who have administrative privileges can change the configuration of their system, even if they have no authority to do so. This ability to change the local SNMP agent configuration can affect the operation of the system, cause network management problems, or affect the operation of the device.

Consequently, SNMP configuration files should be controlled and, if possible, centrally managed to identify and correct configuration changes. This can be done in a variety of ways, including tools such as *tripwire*.

Multiple Management Stations

While this is not a security problem per se, multiple management stations polling the same device can cause problems ranging from poor performance, to differing SNMP configuration information, to the apparent loss of service.

If your network is large enough to require multiple management stations, separate communities should be established to prevent these events from taking place. Remember, there is no constraint on the number of SNMP communities that can be used in the network; it is only the network engineer who imposes the limits.

Denial-of-Service Attacks

Denial of service is defined as the loss of service availability either through authorized or unauthorized configuration changes. It is important to be clear about authorized and unauthorized changes. The system or application administrator who makes a configuration change as part of his job and causes a loss of service has the same impact as the attacker who executes a program to cause the loss of service remotely.

A key problem with SNMP is the ability to change the configuration of the system causing the service outage, or to change the SNMP configuration and imitate a denial of service as reported by the monitoring station. In either situation, someone has to review and possibly correct the configuration problem, regardless of the cause. This has a cost to the company, even if an authorized person made the change.

The Impact of CERT CA-2002–03

Most equipment manufacturers, enterprises, and individuals felt the impact of the CERT advisory issued by the Carnegie Mellon Software Engineering Institute (CM-SEI) Computer Emergency Response Team Coordination Center (CERT-CC). The advisory was issued after the Oulu University Secure Programming Group conducted a very thorough analysis of the message-handling capabilities of SNMP Version 1. While the advisory is specifically for SNMP Version 1, most SNMP implementations use the same program code for decoding the PDU, potentially affecting all SNMP versions.

The primary issues noted in the advisory as it affects SNMP involve the potential for unauthorized privileged access, denial-of-service attacks, or other unstable behavior. Specifically, the work performed by Oulu University found problems with decoding trap messages received by the SNMP

management station or requests received by the SNMP agent on the network device.

It was also identified that some of the vulnerabilities found in the SNMP implementation did not require the correct community string. Consequently, vendors have been issuing patches for their SNMP implementations; but more importantly, enterprises have been testing for vulnerabilities within their networks.

The cost of the vulnerabilities in code, which has been in use for decades, will cost developers millions of dollars for new development activities to remove the vulnerabilities, verify them, and release patches. The users of those products will also spend millions of dollars on patching and implementing other controls to limit the potential exposures.

Many of the recommendations provided by CERT for addressing the problem are solutions for the common security problems when using SNMP. The recommendations provided by CERT can be considered common sense, because SNMP should be treated as a network service:

- *Disable SNMP.* If the device in question is not monitored using SNMP, it is likely safe to disable the service. Remember, if you are monitoring the device and disable SNMP in error, your management station will report the device as down.
- *Implement perimeter network filtering.* Most enterprises should filter inbound SNMP requests from external networks to prevent unauthorized individuals or organizations from retrieving SNMP information about your network devices. Sufficient information exists in the SNMP data to provide a good view of how to attack your enterprise. Secondly, outbound filtering should be applied to prevent SNMP requests from leaving your network and being directed to another enterprise. The obvious exceptions here are if you are monitoring another network outside yours, or if an external organization is providing SNMP-based monitoring systems for your network.
- *Implement authorized SNMP host filtering.* Not every user who wants to should be able to issue SNMP queries to the network devices. Consequently, filters can be installed in the network devices such as routers and switches to limit the source and destination addresses for SNMP requests. Additionally, the SNMP configuration of the agent should include the appropriate details to limit the authorized SNMP management and trap stations.
- *Change default community strings.* A major problem in most enterprises, the default community strings of public and private should be changed to a complex string; and knowledge of that string should be limited to as few people as possible.
- *Create a separate management network.* This can be a long, involved, and expensive process that many enterprises do not undertake. A separate

management network keeps connectivity to the network devices even when there is a failure on the network portion. However, it requires a completely separate infrastructure, making it expensive to implement and difficult to retrofit. If you are building a new network, or have an existing network with critical operational requirements, a separate management network is highly advisable.

The recommendations identified here should be implemented by many enterprises, even if all their network devices have the latest patches implemented. Implementing these techniques for other network protocols and services in addition to SNMP can greatly reduce the risk of unauthorized network access and data loss.

SUMMARY

The goal of SNMP is to provide a simple yet powerful mechanism to change the configuration and monitor the state and availability of the systems and network devices. However, the nature of SNMP, as with other network protocols, also exposes it to attack and improper use by network managers, system administrators, and security personnel.

Understanding the basics of SNMP and the major security issues affecting its use as discussed here helps the security manager communicate concerns about network design and implementation with the network manager or network engineer.

Acknowledgments

The author thanks Cathy Buchanan of Nortel Network's Internet Engineering team for her editorial and technical clarifications.

And thanks to Mignona Cote, my friend and colleague, for her continued support and ideas. Her assistance continues to expand my vision and provides challenges on a daily basis.

References

Internet Engineering Task Force (IETF) Request for Comments (RFC) documents:

RFC-1089 SNMP over Ethernet

RFC-1157 SNMP over Ethernet

RFC-1187 Bulk Table Retrieval with the SNMP

RFC-1215 Convention for Defining Traps for Use with the SNMP

RFC-1227 SNMP MUX Protocol and MIB

RFC-1228 SNMP-DPI: Simple Network Management Protocol Distributed Program

RFC-1270 SNMP Communications Services

RFC-1303 A Convention for Describing SNMP-Based Agents

RFC-1352 SNMP Security Protocols
RFC-1353 Definitions of Managed Objects for Administration of SNMP
RFC-1381 SNMP MIB Extension for X.25 LAPB
RFC-1382 SNMP MIB Extension for the X.25 Packet Layer
RFC-1418 SNMP over OSI
RFC-1419 SNMP over AppleTalk
RFC-1420 SNMP over IPX
RFC-1461 SNMP MIB Extension for Multiprotocol Interconnect over X.25
RFC-1503 Algorithms for Automating Administration in SNMPv2 Managers
RFC-1901 Introduction to Community-Based SNMPv2
RFC-1909 An Administrative Infrastructure for SNMPv2
RFC-1910 User-Based Security Model for SNMPv2
RFC-2011 SNMPv2 Management Information Base for the Internet Protocol
RFC-2012 SNMPv2 Management Information Base for the Transmission Control Protocol
RFC-2013 SNMPv2 Management Information Base for the User Datagram Protocol
RFC-2089 V2ToV1 Mapping SNMPv2 onto SNMPv1 within a Bi-Lingual SNMP Agent
RFC-2273 SNMPv3 Applications
RFC-2571 An Architecture for Describing SNMP Management Frameworks
RFC-2573 SNMP Applications
RFC-2742 Definitions of Managed Objects for Extensible SNMP Agents
RFC-2962 An SNMP Application-Level Gateway for Payload Address
CERT Advisory CA-2002–03

ABOUT THE AUTHOR

Chris Hare, CISSP, CISA, is an information security and control consultant with Nortel Networks in Dallas, Texas. A frequent speaker and author, his experience includes application design, quality assurance, systems administration and engineering, network analysis, and security consulting, operations, and architecture.