Chapter 10
Securing SNMP Access

Simple Network Management Protocol (SNMP) is a set of protocols for managing complex networks. SNMP sends messages, called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters.

The chapter “Securing Access to Management Functions” on page 2-1 introduced a few methods used to secure SNMP access. They included the following:

- “Using ACLs to Restrict SNMP Access” on page 2-5
- “Restricting SNMP Access to a Specific IP Address” on page 2-8
- “Restricting SNMP Access to a Specific VLAN” on page 2-9
- “Disabling SNMP Access” on page 2-11

This chapter presents additional methods for securing SNMP access to HP devices. It contains the following sections:

- “Establishing SNMP Community Strings” on page 10-1
- “Using the User-Based Security Model” on page 10-5
- “Defining SNMP Views” on page 10-10

Restricting SNMP access using ACL, VLAN, or a specific IP address constitute the first level of defense when the packet arrives at an HP device. The next level uses one of the following methods:

- Community string match In SNMP versions 1 and 2
- User-based model in SNMP version 3

SNMP views are incorporated in community strings and the user-based model.

Establishing SNMP Community Strings

SNMP versions 1 and 2 use community strings to restrict SNMP access. The default passwords for Web management access are the SNMP community strings configured on the device.

- The default read-only community string is “public”. To open a read-only Web management session, enter “get” and “public” for the user name and password.
- There is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web management interface. You first must configure a read-write community string using the CLI. Then you can log on using “set” as the user name and the read-write community string you configure

as the password.

You can configure as many additional read-only and read-write community strings as you need. The number of strings you can configure depends on the memory on the device. There is no practical limit.

The Web management interface supports only one read-write session at a time. When a read-write session is open on the Web management interface, subsequent sessions are read-only, even if the session login is "set" with a valid read-write password.

NOTE: If you delete the startup-config file, the device automatically re-adds the default "public" read-only community string the next time you load the software.

NOTE: As an alternative to the SNMP community strings, you can secure Web management access using local user accounts or ACLs. See "Setting Up Local User Accounts" on page 2-16 or "Using an ACL to Restrict Web Management Access" on page 2-5.

Encryption of SNMP Community Strings

The software automatically encrypts SNMP community strings. Users with read-only access or who do not have access to management functions in the CLI cannot display the strings. For users with read-write access, the strings are encrypted in the CLI but are shown in the clear in the Web management interface.

Encryption is enabled by default. You can disable encryption for individual strings or trap receivers if desired. See the next section for information about encryption.

Adding an SNMP Community String

To add a community string, use one of the following methods. When you add a community string, you can specify whether the string is encrypted or clear. By default, the string is encrypted.

USING THE CLI

To add an encrypted community string, enter commands such as the following:

```
ProCurveRS(config)# snmp-server community private rw
ProCurveRS(config)# write memory
```

Syntax: snmp-server community [0 | 1] <string>
ro | rw [view <viewname>] [<standard-acl-name> | <standard-acl-id>]

The <string> parameter specifies the community string name. The string can be up to 32 characters long.

The **ro | rw** parameter specifies whether the string is read-only (**ro**) or read-write (**rw**).

The **0** | **1** parameter affects encryption for display of the string in the running-config and the startup-config file. Encryption is enabled by default. When encryption is enabled, the community string is encrypted in the CLI regardless of the access level you are using. In the Web management interface, the community string is encrypted at the read-only access level but is visible at the read-write access level.

The encryption option can be omitted (the default) or can be one of the following:

- **0** – Disables encryption for the community string you specify with the command. The community string is shown as clear text in the running-config and the startup-config file. Use this option if you do not want the display of the community string to be encrypted.
- **1** – Assumes that the community string you enter is the encrypted form, and decrypts the value before using it.

NOTE: If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter **0** or **1**. Instead, omit the encryption option and allow the software to use the default behavior.

If you specify encryption option **1**, the software assumes that you are entering the encrypted form of the community string. In this case, the software decrypts the community string you enter before using the value for authentication. If you accidentally enter option **1** followed by the clear-text version of the community string, authentication will fail because the value used by the software will not match the value you intended to use.

The command in the example above adds the read-write SNMP community string "private". When you save the new community string to the startup-config file (using the **write memory** command), the software adds the following command to the file:

```
snmp-server community 1 <encrypted-string> rw
```

To add an non-encrypted community string, you must explicitly specify that you do not want the software to encrypt the string. Here is an example:

```
ProCurveRS(config)# snmp-server community 0 private rw
ProCurveRS(config)# write memory
```

The command in this example adds the string "private" in the clear, which means the string is displayed in the clear. When you save the new community string to the startup-config file, the software adds the following command to the file:

```
snmp-server community 0 private rw
```

The **view** <viewstring> parameter is optional. It allows you to associate a view to the members of this community string. Enter up to 32 alphanumeric characters. If no view is specified, access to the full MIB is granted. The view that you want must exist before you can associate it to a community string. Here is an example of how to use the view parameter in the community string command:

```
ProCurveRS(config)# snmp-s community myread ro view sysview
```

The command in this example associates the view "sysview" to the community string named "myread". The community string has read-only access to "sysview". For information on how create views, see the section "Defining SNMP Views" on page 10-10.

The <standard-acl-name> | <standard-acl-id> parameter is optional. It allows you to specify which ACL group will be used to filter incoming SNMP packets. You can enter either the ACL name or its ID. Here are some examples:

```
ProCurveRS(config) # snmp-s community myread ro view sysview 2
ProCurveRS(config) # snmp-s community myread ro view sysview myacl
```

The command in the first example indicates that ACL group 2 will filter incoming SNMP packets; whereas, the command in the second example uses the ACL group called "myacl" to filter incoming packets. See "Using ACLs to Restrict SNMP Access" on page 2-5 for more information.

USING THE WEB MANAGEMENT INTERFACE

NOTE: To make configuration changes, including changes involving SNMP community strings, you must first configure a read-write community string using the CLI. Alternatively, you must configure another authentication method and log on to the CLI using a valid password for that method.

To use the Web interface to add a community string, do the following:

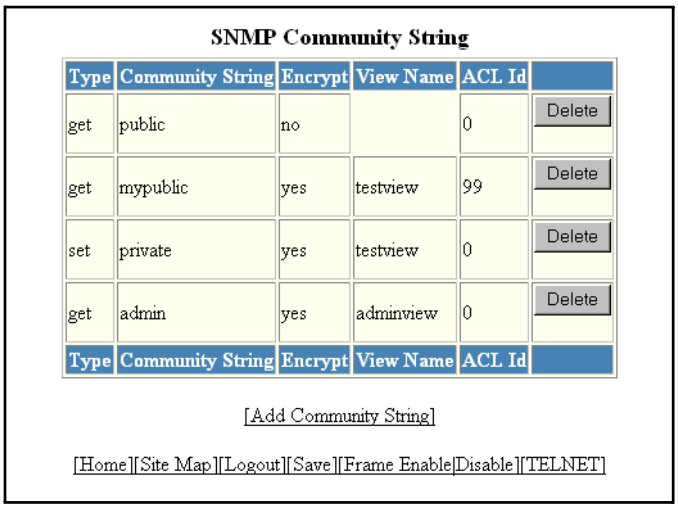
1. Log on to the device using a valid user name and password for read-write access.

   NOTE: If you have configured the device to secure Web management access using local user accounts, you must instead enter the user name and password of one of the user accounts. See "Setting Up Local User Accounts" on page 2-16.

2. Click the Management link on the System configuration panel to display the Management configuration panel.

3. Click the Community String link to display the SNMP Community String panel. This panel shows a list of configured community strings.

   For example,

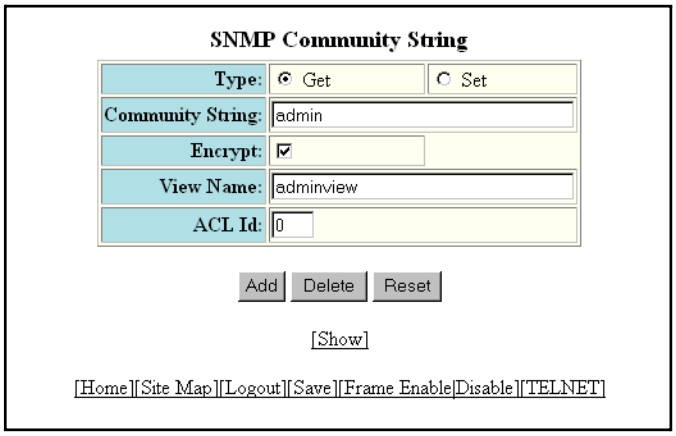


4. Click Add Community String to display the SNMP Community String fields.
5. Select the type of community string you are adding by clicking the "Get" or "Set" button. "Get" provides read-only access, while "Set" provides read-write access.
6. Enter the name of the community string.
7. Encryption is enabled by default. Remove the checkmark from the Encrypt box if you want to disable encryption of the string display. If you disable encryption, other users can view the community string.

   To re-enable encryption, place a checkmark in the Encrypt box.
8. Enter a name for the view that will be assigned to the community string.
9. Enter the number of the ACL that will be used to filter SNMP packets for this community string.

NOTE: In this release, ACL by name is not supported in the Web Interface.

Here is an example of a completed form.



10. Click Add to apply the change to the device's running-config file.
11. Select the Save link at the bottom of the panel. Select Yes when prompted to save the configuration change to the startup-config file on the device's flash memory.

Displaying the SNMP Community Strings

To display the SNMP community strings, use one of the following methods.

USING THE CLI

To display the configured community strings, enter the following command at any CLI level:

```
ProCurveRS(config)# show snmp server
```

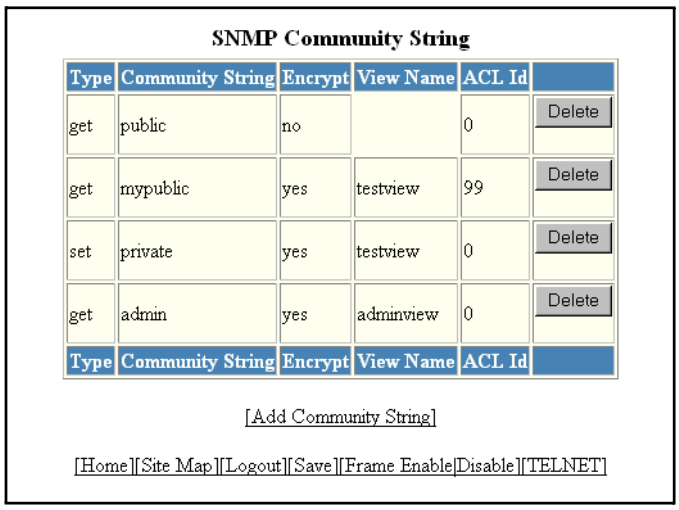
Syntax: show snmp server

See the *Command Line Interface Reference for ProCurve 9300/9400 Series Routing Switches* for an example of the information displayed by the command.

NOTE: If display of the strings is encrypted, the strings are not displayed. Encryption is enabled by default.

USING THE WEB MANAGEMENT INTERFACE

1. Log on to the device using a valid user name and password for read-write access.
2. Select the Management link from the System configuration panel to display the Management configuration panel.
3. Select the Community String link to display the SNMP Community String panel, as shown in the following example.



Using the User-Based Security Model

SNMP version 3 (RFC 2570 through 2575) introduces a User-Based Security model (RFC 2574) for authentication and privacy services.

SNMP version 1 and version 2 use community strings to authenticate SNMP access to management modules. This method can still be used for authentication. In SNMP version 3, the User-Based Security model of SNMP can be used to secure against the following threats:

- Modification of information
- Masquerading the identity of an authorized entity

- Message stream modification
- Disclosure of information

Furthermore, SNMP version 3 supports View-Based Access Control Mechanism (RFC 2575) to control access at the PDU level. It defines mechanisms for determining whether or not access to a managed object in a local MIB by a remote principal should be allowed. (See the section "Defining SNMP Views" on page 10-10.)

NOTE: SNMP version 3 Notification is not supported at this time. The system will generate traps in SNMP version 1 format, just as in earlier releases.

Configuring Your NMS

To be able to use the SNMP version 3 features:

1. Make sure that your Network Manager System (NMS) supports SNMP version 3.
2. Configure your NMS agent with the necessary users.
3. Configure the SNMP version 3 features in HP devices.

Configuring SNMP Version 3 on HP Devices

To configure SNMP version 3 on HP devices, do the following:

1. Enter an engine ID for the management module using the **snmp-server engineid** command if you will not use the default engine ID. See "Defining the Engine ID" on page 10-6.
2. Create views that will be assigned to SNMP user groups using the **snmp-server view** command. See the "Defining SNMP Views" on page 10-10 for details.
3. Create ACL groups that will be assigned to SNMP user groups using the **access-list** command. Refer to the *Command Line Interface Reference for ProCurve 9300/9400 Series Routing Switches* for details.
4. Create user groups using the **snmp-server group** command. See "Defining an SNMP Group" on page 10-7.
5. Create user accounts and associate these accounts to user groups using the **snmp-server user** command. See "Defining an SNMP User Account" on page 10-8.

NOTE: In this release, configuration of SNMP version 3 features is done using the CLI. No Web Interface or SNMP interface is available.

If SNMP version 3 is not configured, then community strings by default are used to authenticate access.

Defining the Engine ID

A default engine ID is generated during system start up. To determine what the default engine ID of the device is, enter the **show snmp engineid** command and find the following line.

```
Local SNMP Engine ID: 800007c70300e05290ab60
```

See the section "Displaying the Engine ID" on page 10-9 for details.

The default engine ID guarantees the uniqueness of the engine ID for SNMP version 3. If you want to change the default engine ID, enter a command such as the following:

```
ProCurveRS(config)# snmp-server engineid local 800007c70300e05290ab60
```

Syntax: [no] snmp-server engineid local <hex-string>

The **local** parameter indicates that engine ID to be entered is the ID of this device, representing an SNMP management entity.

NOTE: Since the current implementation of SNMP version 3 does not support Notification, remote engine IDs cannot be configured at this time.

The <hex-string> variable consists of 11 octets, entered as hexadecimal values. There are two hexadecimal characters in each octet. There should be an even number of hexadecimal characters in an engine ID.

The default engine ID has a maximum of 11 octets:

- Octets 1 through 4 represent the agent's SNMP management private enterprise number as assigned by the Internet Assigned Numbers Authority (IANA). The most significant bit of Octet 1 is "1".
- Octet 5 is always 03 in hexadecimal and indicates that the next set of values represent a MAC address.
- Octets 6 through 11 form the MAC address of the lowest port in the management module.

NOTE: Engine ID must be a unique number among the various SNMP engines in the management domain. Using the default engine ID ensures the uniqueness of the numbers.

Defining an SNMP Group

SNMP groups map SNMP users to SNMP views. For each SNMP group, you can configure a read view, a write view, or both. Users who are mapped to a group will use its views for access control.

To configure an SNMP user group, enter a command such as the following:

```
ProCurveRS(config)# snmp-server group admin v3 auth read v1default write v1default
```

Syntax: [no] snmp-server group <groupname>
v1 | v2 | v3
auth | noauth | priv
[access <standard-acl-id>] [read <viewstring> | write <viewstring>]

NOTE: This command is not used for SNMP version 1 and SNMP version 2. In these versions, groups and group views are created internally using community strings. (See “Establishing SNMP Community Strings” on page 10-1.) When a community string is created, two groups are created, based on the community string name. One group is for SNMP version 1 packets, while the other is for SNMP version 2 packets.

The **group** <groupname> parameter defines the name of the SNMP group to be created.

The **v1**, **v2**, or **v3** parameter indicates which version of SNMP is used. In most cases, you will be using v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

The **auth** | **noauth** parameter determines whether or not authentication will be required to access the supported views. If auth is selected, then only authenticated packets are allowed to access the view specified for the user group. Selecting **noauth** means that no authentication is required to access the specified view. Selecting **priv** means that an authentication password will be required from the users.

The **access** <standard-acl-id> parameter is optional. It allows incoming SNMP packets to be filtered based on the standard ACL attached to the group.

The **read** <viewstring> | **write** <viewstring> parameter is optional. It indicates that users who belong to this group have either read or write access to the MIB.

The <viewstring> variable is the name of the view to which the SNMP group members have access. If no view is specified, then the group has no access to the MIB.

The value of <viewstring> is defined using the **snmp-server view** command. The SNMP agent comes with the "v1default" view, the default view that provides access to the entire MIB; however, it must be specified when creating the group. The "v1default" view also allows SNMP version 3 to be backwards compatibility with SNMP version 1 and version 2.

NOTE: If you will be using a view other than the "v1default" view, that view must be configured before creating the user group. See the section “Defining SNMP Views” on page 10-10, especially for details on the include | exclude parameters.

Defining an SNMP User Account

The **snmp-server user** command does the following:

- Creates an SNMP user.
- Defines the group to which the user will be associated.
- Defines the type of authentication to be used for SNMP access by this user.

Here is an example of how to create the account:

```
ProCurveRS(config)# snmp-s user bob admin v3 access 2 auth md5 bobmd5 priv des
bobdes
```

The CLI for creating SNMP version 3 users has been updated as follows.

Syntax: [no] snmp-server user <name> <groupname> v3
[[access <standard-acl-id>] [encrypted] [auth md5 <md5-password> | sha <sha-password>]
[priv [encrypted] des <des-password>]]

The <name> parameter defines the SNMP user name or security name used to access the management module.

The <groupname> parameter identifies the SNMP group to which this user is associated or mapped. All users must be mapped to an SNMP group. Groups are defined using the **snmp-server group** command.

NOTE: The SNMP group to which the user account will be mapped should be configured before creating the user accounts; otherwise, the group will be created without any views. Also, ACL groups must be configured before configuring user accounts.

The **v3** parameter is required.

The **access** <standard-acl-id> parameter is optional. It indicates that incoming SNMP packets are filtered based on the ACL attached to the user account.

NOTE: The ACL specified in a user account overrides the ACL assigned to the group to which the user is mapped. If no ACL is entered for the user account, then the ACL configured for the group will be used to filter packets.

The **encrypted** parameter means that the MD5 or SHA password will be a digest value. MD5 has 16 octets in the digest. SHA has 20. The digest string has to be entered as a hexadecimal string. In this case, the agent need not generate any explicit digest. If the **encrypted** parameter is not used, the user is expected to enter the authentication password string for MD5 or SHA. The agent will convert the password string to a digest, as described in RFC 2574.

The **auth md5 | sha** parameter is optional. It defines the type of encryption that the user must have to be authenticated. Choose between MD5 or SHA encryption. MD5 and SHA are two authentication protocols used in SNMP version 3.

The <md5-password> and <sha-password> define the password the user must use to be authenticated. These password must have a minimum of 8 characters. If the encrypted parameter is used, then the digest has 16 octets for MD5 or 20 octets for SHA.

NOTE: Once a password string is entered, the generated configuration displays the digest (for security reasons), not the actual password.

The **priv [encrypted] des** <des-password> parameter is optional. It defines the type of encryption that will be used to encrypt the privacy password. If the "encryption" keyword is used, enter a 16-octet DES key in

hexadecimal format for the des-password. If the "encryption" keyword is not used enter a password string. The agent will generate a suitable 16-octet DES key from the password string.

Currently, DES is the only encryption type supported for priv password.

Displaying the Engine ID

To display the engine ID of a management module, enter a command such as the following:

```
ProCurveRS(config)# show snmp engineid
Local SNMP Engine ID: 800007c70300e05290ab60
Engine Boots: 3
Engine time: 5
```

Syntax: show snmp engineid

The engine ID identifies the source or destination of the packet.

The engine boots represents the number of times that the SNMP engine reinitialized itself with the same engine ID. If the engineID is modified, the boot count is reset to 0.

The engine time represents the current time with the SNMP agent.

Displaying SNMP Groups

To display the definition of an SNMP group, enter a command such as the following:

```
ProCurveRS(config)# show snmp group
groupname = exceptifgrp
security model = v3
security level = authNoPriv
ACL id = 2
readview = exceptif
writeview = <none>
```

Syntax: show snmp group

The value for security level can be one of the following:

Security Level	Authentication
<none>	If the security model shows v1 or v2, then security level is blank. User names are not used to authenticate users; community strings are used instead.
noauthNoPriv	Displays if the security model shows v3 and user authentication is by user name only.
authNoPriv	Displays if the security model shows v3 and user authentication is by user name and the MD5 or SHA algorithm.

Displaying User Information

To display the definition of an SNMP user account, enter a command such as the following:

```
ProCurveRS(config)# show snmp user
username = bob
acl id = 2
group = admin
security model = v3
group acl id = 0
authtype = md5
authkey = 3aca18d90b8d172760e2dd2e8f59b7fe
privtype = des,  privkey = 1088359afb3701730173a6332d406eec
engine ID= 800007c70300e052ab0000
```

Syntax: show snmp user

Interpreting Varbinds in Report Packets

If an SNMP version 3 request packet is to be rejected by an SNMP agent, the agent sends a report packet that contains one or more varbinds. The varbinds contain additional information, showing the cause of failures. An SNMP manager application decodes the description from the varbind. The following table presents a list of varbinds supported by the SNMP agent.

Varbind Object Identifier	Description
1. 3. 6. 1. 6. 3. 11. 2. 1. 3. 0	Unknown packet data unit.
1. 3. 6. 1. 6. 3. 12. 1. 5. 0	The value of the varbind shows the engine ID that needs to be used in the snmp-server engineid command
1. 3. 6. 1. 6. 3. 15. 1. 1. 1. 0	Unsupported security level.
1. 3. 6. 1. 6. 3. 15. 1. 1. 2. 0	Not in time packet.
1. 3. 6. 1. 6. 3. 15. 1. 1. 3. 0	Unknown user name. This varbind may also be generated: • If the configured ACL for this user filters out this packet. • If the group associated with the user is unknown.
1. 3. 6. 1. 6. 3. 15. 1. 1. 4. 0	Unknown engine ID. The value of this varbind would be the correct authoritative engineID that should be used.
1. 3. 6. 1. 6. 3. 15. 1. 1. 5. 0	Wrong digest.
1. 3. 6. 1. 6. 3. 15. 1. 1. 6. 0	Decryption error.

Defining SNMP Views

SNMP views are named groups of MIB objects that can be associated with user accounts to allow limited access for viewing and modification of SNMP statistics and system configuration. SNMP views can also be used with other commands that take SNMP views as an argument. SNMP views reference MIB objects using object names,

numbers, wildcards, or a combination of the three. The numbers represent the hierarchical location of the object in the MIB tree. You can reference individual objects in the MIB tree or a subset of objects from the MIB tree.

To configure the number of SNMP views available on the HP device:

```
ProCurveRS(config)# system-max view 15
```

Syntax: system-max view <number-of-views>

This command specifies the maximum number of SNMPv2 and v3 views that can be configured on a device. The number of views can be from 10 – 65536. The default is 10 views.

To add an SNMP view, enter one of the following commands:

```
ProCurveRS(config)# snmp-server view Maynes system included
ProCurveRS(config)# snmp-server view Maynes system.2 excluded
ProCurveRS(config)# snmp-server view Maynes 2.3.*.6 included
ProCurveRS(config)# write mem
```

NOTE: The **snmp-server view** command supports the MIB objects as defined in RFC 1445.

Syntax: [no] snmp-server view <name> <mib_tree> included | excluded

The <name> parameter can be any alphanumeric name you choose to identify the view. The names cannot contain spaces.

The <mib_tree> parameter is the name of the MIB object or family. MIB objects and MIB sub-trees can be identified by a name or by the numbers called Object Identifiers (OIDs) that represent the position of the object or sub-tree in the MIB hierarchy. You can use a wildcard (*) in the numbers to specify a sub-tree family.

The **included** | **excluded** parameter specifies whether the MIB objects identified by the <mib_family> parameter are included in the view or excluded from the view.

NOTE: All MIB objects are automatically excluded from any view unless they are explicitly included; therefore, when creating views using the **snmp-server view** command, indicate which portion of the MIB you want users to access.

To delete a view, use the no parameter before the command.