SNMP (Simple Network Management Protocol)—Application layer (L7) protocol to remotely query, monitor, or even change a collection of variables on TCP/IP devices.

Agent—Software running on the managed device, e.g. a router.

NMS (Network Management Station) AKA Manager—Application on a PC to collect data and perform statistics, monitor thresholds for alerts, etc., e.g. "Cisco Prime."

MIB (Management Information Base)—The agent's database of variables comprising the device's parameters.

OID (Object ID)—A variable in the MIB, denoted by name or number. These variables are organized by the MIB into a hierarchy. Some public vars are defined by RFC. Vendors can have private branches of variables.

Variable—e.g. the load on one interfaces on one router.

Managers can remotely query devices with "get" messages or remotely configure the device with "set" messages.

Traps—A device can notify the NMS with the state of a variable. Not acknowledged, therefore "unreliable." Version 3 adds "inform" messages that are acknowledged & reliable.

SECURITY

ACLs can be used to limit access to known admins (NMS workstations). You may need to filter both IPv4 and IPv6.

Community Strings—like a shared-secret password. They're sent in plain-text with each get or set message. You can have separate ones for read-only and read-write access.

SNMP Versions—2c adds messaging improvements for large quantities of statistics, but still has cleartext passwords (community strings).

Version 3 eliminates community strings and improves security with

- Message integrity (no tampering)
- Authentication
- Encryption

SNMP Version 3 Security Levels—message integrity is always guaranteed. Notice that "noauth" has username authentication; it just doesn't have a password for that username.

Name	Keyword for snmp-server Cmd	Authentication	Encryption
noAuthNoPriv	`noauth`	Username	None
authNoPriv	`auth`	MD5 or SHA	None
authPriv	`priv`	MD5 or SHA	DES, 3DES, or AES

Enabling SNMP

All SNMP commands are global (no subcommand mode). By default, the agent is disabled until the first "snmp-server" command is typed, then on. Only way to turn back off is to eliminate all "snmp-server" commands from the config ("no…") and reload.

```
R7(config)# access-list 99 permit host 172.28.0.5
R7(config)# snmp-server community mySNMPpassword RO 99
R7(config)# snmp-server community myOtherPassword RW 99
                    Community string is the cleartext password used by versions 1 and 2c
                    RO = Read-only access (safer), RW = Read-Write
R7(config)# snmp-server location San Jose
R7(config)# snmp-server contact Ben Steel
                    The location and contact are optional.
```

Enabling Traps

Note: In the official cert guide, Mr. Odom shows an example [Example 26-2, page 702] of "inform" being used with version 2c of SNMP. Actually, inform is a version 3 feature.

```
R7(config)# snmp-server host 172.28.0.5 version 2c myTrapPassword
                    You can use DNS hostnames for the NMS (172.28.0.5)
R7(config)# snmp-server enable traps
```

Configuration Verification

Configuration settings are retrieved one at a time, with a separate "show" command for each.

```
1  R7# show snmp community
2
3  Community name: ILMI
4  Community Index: cisco0
5  Community SecurityName: ILMI
6  storage-type: read-only  active
7
8  Community name: mySNMPpassword
9  Community Index: cisco1
10 Community SecurityName: mySNMPpassword
11 storage-type: nonvolatile        active access-list: 99
12
13 Community name: myOtherPassword
14 Community Index: cisco2
15 Community SecurityName: myOtherPassword
16 storage-type: nonvolatile        active access-list: 99
17
18 Community name: myTrapPassword
19 Community Index: cisco3
20 Community SecurityName: myTrapPassword
21 storage-type: nonvolatile        active
```

ILMI [Lines 4-7] is a pre-configured community present by default. As for security, it's a good reminder of the need for an ACL when enabling SNMP.

```
R7# show snmp location
San Jose

R7# show snmp contact
Ben Steel

R7# show snmp host
Notification host: 172.28.0.5   udp-port: 162   type: trap
user: myTrapPassword    security model: v2c
```

Operation Verification

```
R7# show snmp
Chassis: FTX1211A1BH
Contact: Ben Steel
Location: San Jose
0 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    0 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    0 Get-next PDUs
    0 Set-request PDUs
    0 Input queue packet drops (Maximum queue size 1000)
0 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    0 Response PDUs
    0 Trap PDUs
SNMP Dispatcher:
   queue 0/75 (current/max), 0 dropped
SNMP Engine:
   queue 0/1000 (current/max), 0 dropped

SNMP logging: enabled
    Logging to 172.28.0.5.162, 0/10, 0 sent, 0 dropped.
```

Use Example—Out of Scope Bonus

snmpget—freeware utility to query the MIB of a remote device

```
HostComputer:~$ snmpget -v2c -c mySnmpPassword 192.168.2.210 1.3.6.1.4.1.9.2.1.58.0
SNMPv2-SMI::enterprises.9.2.1.58.0 = INTEGER: 0
```

Response is the router's 5-minute exponential moving average of the CPU busy percentage
-v the version of SNMP in use (2c)
-c the password (AKA Community String)

Communities are completely removed from version 3 in favor of group and user security. SNMP groups and users are local to the machine they're declared on, but since the whole point of SNMP is to centralize the monitoring and control of many devices, some consistency is wise.

Group

Groups are a Ciscoism that allow security settings to be set for entire groups of SNMP users. You'll need a group before you can create users; it's a mandatory field in the user config. The options vary, but a group declaration for version 3 includes the following possibilities.

```
snmp-server group <Group Name> v3
                    {noauth | auth | priv}
                  [ read  <viewName> ]
                  [ write <viewName> ]
                  [ access [ipv6] <ACLname> ]
                  The "access" phrase applies an ACL
```

View—a subset of the MIB that can be assigned to groups to limit the fields those users can access.

V1default—The only predefined group mentioned in the cert guide. It contains most of the MIB. By default a group will have read (get) access to the v1default group and no write (set) access.

User

User accounts for SNMP are separate from the normal IOS username/password command. When you create a user, the SNMP process will start up as a side-effect. A subset of the syntax looks like:

```
snmp-server user <User Name> <Group Name> v3
                 [ auth {md5 | sha} <authenticationPassword>
                   [ priv {  DES | 3DES | AES <keylength> } <encryptionKey> ]
                                                  The keylength for AES can be 128, 192, or 256 bits
                 ]
```

Which sections of the above to use will depend on the group's security setting (noauth, auth, priv).

- With "noauth," the only authentication is knowledge of the username, so the auth (authentication) and priv (privacy) phrases are dropped. Notice that there is no *noauth* keyword; you just stop typing and hit <return>
- Auth adds an authentication password via the *auth* phrase
- Priv (Privacy) adds encryption of entire messages and needs both the *auth* and *priv* phrases, in that order. Cisco doesn't bother giving you the option of privacy without authentication

Leaving off an *auth* or *priv* phrase that the user's group declaration makes necessary won't give an error, but it won't work either; the agent and manager won't communicate [the book says].

No matter what, an "access" phrase can add an ACL, just like in the group declaration.

Host for V3 Informs or Traps

As mentioned at the start, V3 adds the concepts of acknowledged/reliable "inform" messages to the previous concept of "traps" sending unacknowledged status messages from the agent to the NMS. V3 also adds security (noauth/auth/priv and passworded snmp user accounts) to its inform messages, just like it did with gets and sets.

```
snmp-server host <ip address> [informs] version 3 {noauth | auth | priv} <UserName>
                 The User Name matches an "snmp-server user" command and the security level
                 (noauth/auth/priv) matches the setting in that user's group. Curiously, "version 3" is spelled out
                 in this command but simply "v3" in the group and user commands.
```

V3 EXAMPLE AND VERIFICATION

```
1 R7(config)# snmp-server group myGroup v3 auth write v1default
2 R7(config)# snmp-server user myUserName myGroup v3 auth md5 myPassword
3 R7(config)# snmp-server host 172.28.0.5 version 3 auth myUserName
                    Our router sends traps to this host, the auth password "myPassword" is implied and comes
                    from the user declaration on line 2
  R7(config)# snmp-server enable traps
                    Unchanged from version 2c
```

```
1  R7# show snmp group
2                   This command cannot be limited to just one group, but you can pipe it to "begin"
3  groupname: ILMI                                security model:v1
4  contextname: <no context specified>           storage-type: permanent
5  readview : *ilmi                               writeview: *ilmi
6  notifyview: <no notifyview specified>
7  row status: active
8
9  groupname: ILMI                                security model:v2c
10 contextname: <no context specified>           storage-type: permanent
11 readview : *ilmi                               writeview: *ilmi
12 notifyview: <no notifyview specified>
13 row status: active
14
15 groupname: myGroup                             security model:v3 auth
16 contextname: <no context specified>           storage-type: nonvolatile
17 readview : v1default                           writeview: v1default
18                  Every group has a read view of v1default by default and no write view. We added
19                  v1default as the write view for this group when we created it and said nothing about the
20                  read view, so it's there by default.
21 notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F
22 row status: active
```

```
1 R7# show snmp user [ <user name> ]
2                   If you don't limit it by user name, it'll list them all
3 User name: myUserName
4 Engine ID: 800000090300001F6CD43C40
5 storage-type: nonvolatile        active
6 Authentication Protocol: MD5
7 Privacy Protocol: None
8 Group-name: myGroup
```

```
1 R7# show snmp host
2                   Again, this will show all hosts and you can't limit it except by piping the command to an
3                   output modifier. Unfortunately, "section" gives the output you'd expect from "include" (just
4                   one line of output) so you'll need to use "begin."
5 Notification host: 172.28.0.5   udp-port: 162   type: trap
6 user: myUserName         security model: v3 auth
```

Because we didn't specify "informs" or "traps" when we configured "snmp-server host…," it defaulted to traps [Line 5, far right]. In fact, when I showed the syntax for that command, I left off the default "traps" option, which you can explicitly type in place of the "informs" option.