Network Monitoring and Management

Introduction to

SNMP

INNOG 6

March 22 - 25, 2023

Whate is SNMP?

- Simple Network Management Protocol (SNMP)
  - Structured protocol, structured information
  - For querying network device state and receiving notifications
  - Also can be used to change state
  - Industry standard, hundreds of tools exist that use it
  - Supported on any decent network equipment
  - Transport : UDP ports 161 and 162 (notifications)

Uses for SNMP

- Typical queries
  - Bytes In/Out on an interface, errors
  - CPU load
  - Uptime
  - Temperature
  - …
- For hosts (servers or workstations)
  - Disk space
  - Installed software
  - Running processes
  - …

SNMP Versions

- v1 (1988) Original specification
  - Historic
- v2 (1996) Failed Standard
  - Security + new data types + new operators
  - 64-bit counters, get-bulk, v2 notifications
  - View-based access control model (VACM) introduced
  - Historic, no current implementations left
- v2c (1996) De facto standard
  - v2 data types and operators
  - v1 security (community string) (simple security model)
  - Historic
- v3 (1998) Robust security
  - User/view based security (USM/VACM)
  - Full Internet Standard

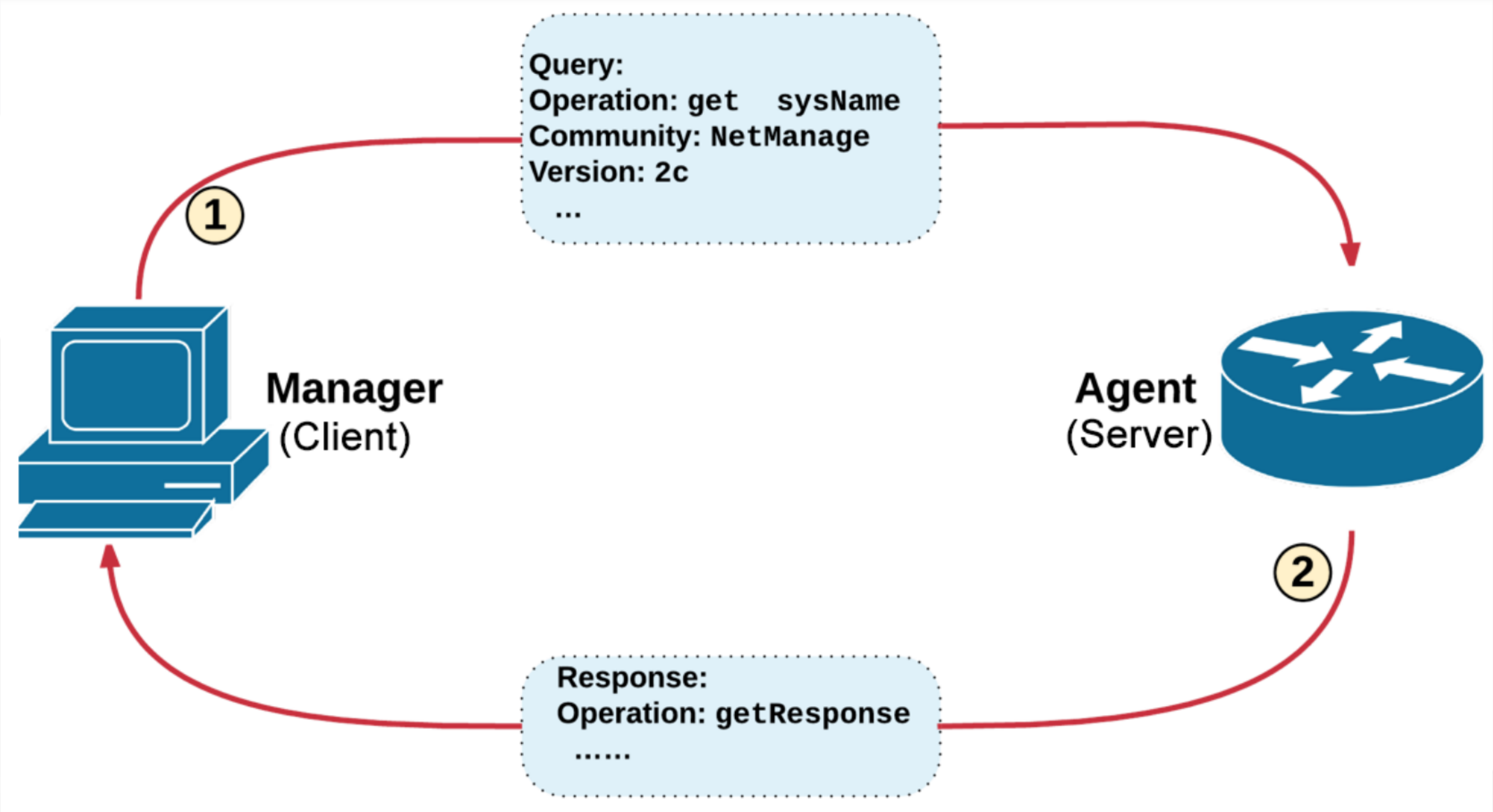
We will use SNMP v2c and v3 in this class

SNMP roles

Terminology - We will be using Manager and Agent

- Manager (the monitoring station)
  - Sometimes known as the SNMP client
  - SNMPv3 calls it the Command Generator and Notification Receiver
- Agent (running on the equipment/server)
  - Sometimes known as the SNMP server
  - SNMPv3 calls it the Command Responder and Notification Originator
  - Windows and UNIX have SNMP agents

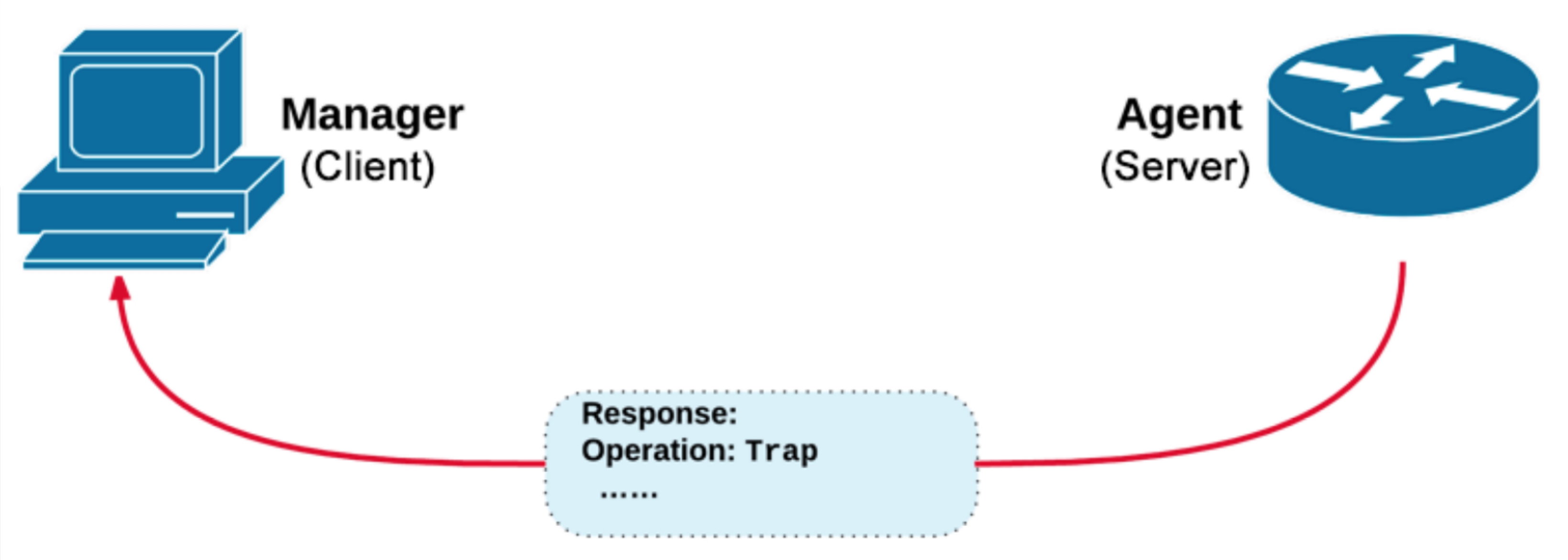
How SNMP works: Query / Response



Query:
Operation: get sysName
Community: NetManage
Version: 2c
...

Response:
Operation: getResponse
......

How SNMP works: Trap / Inform



How SNMP works

Basic operators:

- **get** (manager -> agent)
  - Query for a value
- **getnext** (manager -> agent)
  - Get next value (e.g. list of values for a table)
- **getresponse** (agent -> manager)
  - Response to get, getnext, or set, includes error returns
- **set** (manager -> agent)
  - Set a value, or perform an action
- **trap** (agent -> manager)
  - Spontaneous notification from equipment (line down, temperature above threshold, ...)

How SNMP works (Contd.)

- Query/response based
  - Monitoring generally uses **get**, **getnext**, **getbulk**
  - Changing state uses **set**
  - Response is always a **getresponse**
  - **getbulk** requires v2c or v3
- Notifications are delivered as **traps** or **informs**
  - **traps** are unacknowledged
  - **informs** are acknowledged (v2c, v3)
  - Use v2c format **traps**
  - No one uses **informs**

SNMP TRAPS

- A way for an Agent to notify the Manager without getting a Query from the Manager
- Agent is configured to send TRAP messages when an event occurs
    - *coldStart*
    - *warmStart*
    - *ifDown*
    - *authenticationFailure*
- After the manager receives the TRAP message, it can take further action if necessary

SNMP database

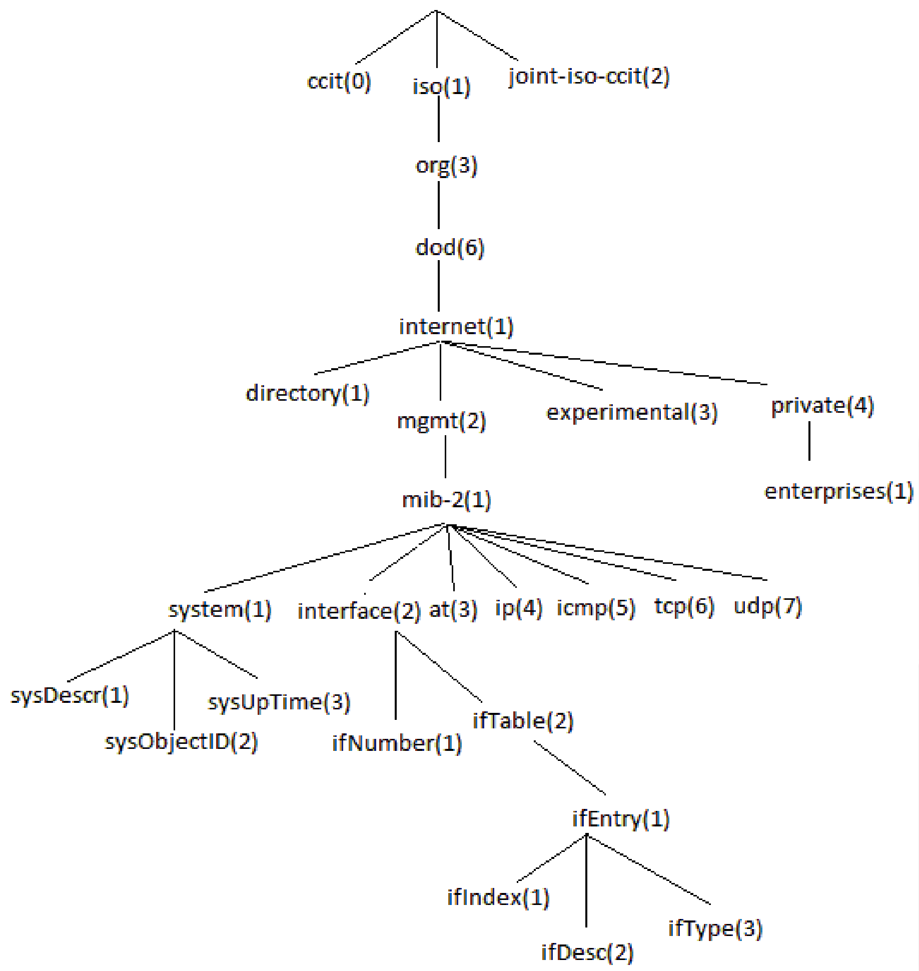
- The information offered by a device is available in its Management Information Base (MIB)
  - SNMP uses Object Identifiers (OIDs) to organize this information
  - OIDs are keys to identifying each piece of data
  - OIDs are organized into a tree structure that is the MIB
  - MIB files document parts of the MIB on a device

OIDs

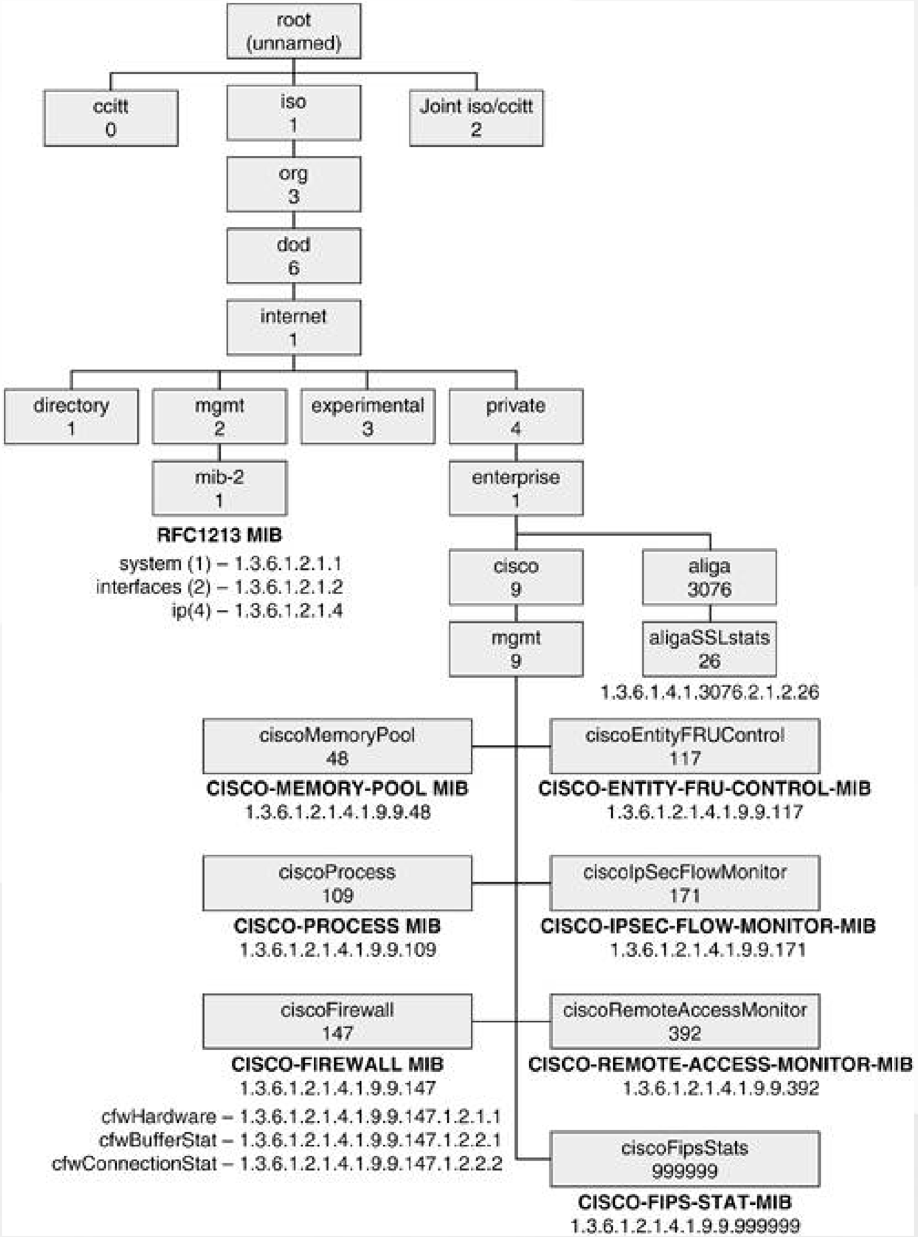
OID: Object Identifier

- A unique key to select a particular item of data in the device
- The same piece of information is always found at the same OID. That's simple!
- An OID is a variable-length string of numbers, e.g.
  - .1.3.6.1.2.1.1.3
- Allocated hierarchically in a tree to ensure uniqueness (similar to DNS)

The MIB Tree



ccit(0)
iso(1)
joint-iso-ccit(2)
org(3)
dod(6)
internet(1)
directory(1)
mgmt(2)
experimental(3)
private(4)
enterprises(1)
mib-2(1)
system(1)
interface(2)
at(3)
ip(4)
icmp(5)
tcp(6)
udp(7)
sysDescr(1)
sysObjectID(2)
sysUpTime(3)
ifNumber(1)
ifTable(2)
ifEntry(1)
ifIndex(1)
ifDesc(2)
ifType(3)



root (unnamed)
ccitt 0
iso 1
Joint iso/ccitt 2
org 3
dod 6
internet 1
directory 1
mgmt 2
experimental 3
private 4
mib-2 1
enterprise 1

RFC1213 MIB
system (1) – 1.3.6.1.2.1.1
interfaces (2) – 1.3.6.1.2.1.2
ip(4) – 1.3.6.1.2.1.4

cisco 9
mgmt 9
aliga 3076
aligaSSLstats 26
1.3.6.1.4.1.3076.2.1.2.26

ciscoMemoryPool 48
CISCO-MEMORY-POOL MIB
1.3.6.1.2.1.4.1.9.9.48

ciscoEntityFRUControl 117
CISCO-ENTITY-FRU-CONTROL-MIB
1.3.6.1.2.1.4.1.9.9.117

ciscoProcess 109
CISCO-PROCESS MIB
1.3.6.1.2.1.4.1.9.9.109

ciscoIpSecFlowMonitor 171
CISCO-IPSEC-FLOW-MONITOR-MIB
1.3.6.1.2.1.4.1.9.9.171

ciscoFirewall 147
CISCO-FIREWALL MIB
1.3.6.1.2.1.4.1.9.9.147
cfwHardware – 1.3.6.1.2.1.4.1.9.9.147.1.2.1.1
cfwBufferStat – 1.3.6.1.2.1.4.1.9.9.147.1.2.2.1
cfwConnectionStat – 1.3.6.1.2.1.4.1.9.9.147.1.2.2.2

ciscoRemoteAccessMonitor 392
CISCO-REMOTE-ACCESS-MONITOR-MIB
1.3.6.1.2.1.4.1.9.9.392

ciscoFipsStats 999999
CISCO-FIPS-STAT-MIB
1.3.6.1.2.1.4.1.9.9.999999

Interesting parts of the MIB tree

The Internet MIB, **.1.3.6.1**, really only two branches of interest:

- Standard MIBs
    - **.1.3.6.1.2.1 = .iso.org.dod.internet.mgmt.mib-2**
- Vendor-specific (proprietary) MIBs
    - **.1.3.6.1.4.1 = .iso.org.dod.internet.private.enterprises**

OIDs and MIB files

Read from left to right

- OID components separated by '.'
  - .1.3.6.1.4.1.9. ...
- Each OID corresponds to a label
  - .1.3.6.1.2.1.1.5 => sysName
- The complete path:
  - .iso.org.dod.internet.mgmt.mib2.system.sysName
- How do we convert from OIDs to Labels (and vice versa)?
  - Use the MIBs files!

MIB Files

- MIB files define the objects that can be queried, including:
  - Object name
  - Object description
  - Data type (integer, text, list)
- MIB files are structured text
  - using an ASN.1 subset called the Structure of Management Information (SMI)
- Standard MIB files include:
  - MIB-II - (RFC1213) - a sub-group of MIBs
  - HOST-RESOURCES-MIB (RFC2790)

MIB Sample

```
sysUpTime OBJECT-TYPE
          SYNTAX TimeTicks
          ACCESS read-only
          STATUS mandatory
          DESCRIPTION
          "The time (in hundredths of a second)
since the network management portion of the system
was last reinitialized."
          ::= { system 3 }
```

MIB Sample (Contd.)

- **sysUpTime OBJECT-TYPE**
  - This defines the object called ***sysUpTime***.
- **SYNTAX TimeTicks**
  - This object is of the type ***TimeTicks***. Object types are specified in the SMI we mentioned a moment ago.
- **ACCESS read-only**
  - This object can only be read via SNMP (i.e. **get**, **getnext**); it cannot be changed (i.e. **set**).
- **STATUS mandatory**
  - This object must be implemented in any SNMP agent.
- **DESCRIPTION**
  - A description of the object
- **::= { system 3 }**
  - The ***sysUpTime*** object is the third branch off of the ***system*** object group tree.

MIB Files (Contd.)

- MIB files also make it possible to interpret a returned value from an agent
  - For example, the status for a fan could be:
    - 1, 2, 3, 4, 5, or 6
    - What does it mean?
- Look for the Textual Convention (tc) in the MIB

MIB Sample

```
CiscoEnvMonState ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
           "Represents the state of a device being monitored.
            Valid values are:
            normal(1):         the environment is good, such as low
                               temperature.
            warning(2):        the environment is bad, such as temperature
                               above normal operation range but not too
                               high.
            critical(3):       the environment is very bad, such as
                               temperature much higher than normal
                               operation limit.
            shutdown(4):       the environment is the worst, the system
                               should be shutdown immediately.
            notPresent(5):     the environmental monitor is not present,
                               such as temperature sensors do not exist.
            notFunctioning(6): the environmental monitor does not
                               function properly, such as a temperature
                               sensor generates a abnormal data like
                               1000 C.
```

SNMP and Security

- SNMP versions 1 and 2c are insecure
- SNMP version 3 was created to fix this
- SNMPv3 authentication is based on a "User-based Security Model" (USM):
    - Authenticity and integrity
    - Keys are used for users, and messages have digital signatures generated with a hash function (MD5 or SHA)
    - Privacy
    - Messages can be encrypted with secret-key (private) algorithms (DES or AES)
    - Temporary validity
    - Utilizes a synchronized clock with a 150 second window with sequence checking

SNMPv3 Security Levels

- **noAuthNoPriv**
    - No authentication, no privacy
- **authNoPriv**
    - Authentication with no privacy
- **authPriv**
    - Authentication with privacy

Cisco SNMP Agent Configuration R/O

Read-only

```
# snmp-server community NetManage RO
```

- Enables SNMPv1 and v2c

```
# snmp-server group ReadGroup v3 auth
# snmp-server user admin ReadGroup v3 auth sha NetManage
```

- SNMPv3 authentication, no encryption

Cisco SNMP Agent Configuration R/W

Read-write

```
# snmp-server group WriteGroup v3 auth write v1default
# snmp-server user admin-rw WriteGroup v3
  auth sha NetManage priv aes 128 NetWrite
```

- Cisco allows authNoPriv and authPriv queries with this user
- You could also define a read-write user without encryption (priv)
- Note that we recommend using SNMP version 3 if you want write access using the set operator

Net-SNMP Agent (snmpd)

- Add a community string by editing `/etc/snmp/snmpd.conf` and adding:

```
rocommunity NetManage 100.64.0.0/16
```

- Add the SNMPv3 user

```
# systemctl stop snmpd
# net-snmp-create-v3-user –X DES -a NetManage admin
# systemctl start snmpd
```

Querying an SNMP agent

Using Net-SNMP command line tools...

- Some typical commands for querying:
  - **snmpget**
  - **snmpwalk**
  - **snmpbulkwalk** (requires v2c or v3)
  - **snmpstatus**
  - **snmptable**
- All commands have same authentication options:

  -
    ```
    snmpXXX -v1 -c<community> host [OID]
    ```

  -
    ```
    snmpXXX -v2c -c<community> host [OID]
    ```

  -
    ```
    snmpXXX -v3 -lauthNoPriv -u<user> -X DES -a<pass>
    host [OID]>
    ```

Querying an SNMP agent (aka server)

Let's look at some examples

```
$ snmpstatus -v2c -c NetManage 100.68.Y.1
$ snmpget -v2c -c NetManage 100.68.Y.1 ifNumber.0
$ snmpwalk -v2c -c NetManage 100.68.Y.1 ifDescr
```

“Y” == your group number

Querying an SNMP agent

- OID
  - A value, for example, .1.3.6.1.2.1.1.5.0
  - or its name equivalent: sysName.0
  - For example:

```
$ snmpget -v2c -c NetManage localhost .1.3.6.1.2.1.1.5.0
$ snmpget -v2c -c NetManage localhost sysName.0
$ snmpget -v2c -c NetManage localhost sysName
```

- Let's ask for the system's name (using the OID above)
  - Why the .0? What do you notice?

Failed Query...Why?

- Two gets:

```
# snmpget -v1 -c NetManage 100.68.1.1 ifHCInOctets.1
Error in packet
Reason: (noSuchName) There is no such variable name in this MIB.
Failed object: IF-MIB::ifHCInOctets.1

# snmpget -v2c -c NetManage 100.68.1.1 ifHCInOctets.1
IF-MIB::ifHCInOctets.1 = Counter64: 144058265
```

- Why? Notice the data type: Counter64. 64-bit counters are only supported in SNMPv2c and v3.
- 64-bit counters are important because 32-bit interface counters (ifInOctets) can wrap in 34 seconds on Gig interfaces. (How fast can it wrap on 10G?)

SNMP failure: no response?

- The device might be offline or unreachable
- The device might not be running an SNMP agent
- The device might be configured with a different community string
- The device might be configured to refuse SNMP queries from your IP address

In all of these cases you will get no response

SNMP Best Practices

- Secure your SNMP access and traffic:
  - Management VLAN
  - Access lists
  - Use SNMPv3 with authentication for queries and sets where possible
- Use SNMPv2c traps
  - Better formatted than v1 traps
  - Accurate timestamps
- Do no harm
  - Only poll as fast as you really need
  - Possible to drive CPU load on devices up and affect other protocol processing
  - It does no good to poll every 5 seconds if the device updates the counter every 10

References

- Essential SNMP (O'Reilly Books) by Douglas Mauro, Kevin Schmidt
  http://www.amazon.com/Essential-Second-Edition-Douglas-Mauro/dp/0596008406
- Wikipedia
  http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol
- MIB/OID Browser
  http://oid-info.com/
- Cisco IOS MIB Tools
  https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index
- Open Source Java MIB Browser
  http://www.dwipal.com/mibbrowser.htm
- SNMP Link – collection of SNMP resources
  http://www.snmplink.org/
- Net-SNMP Open Source SNMP tools
  http://net-snmp.sourceforge.net/