SNMP

Simple Network Management Protocol

27th November 2017

Senevi Herath

What is SNMP

- SNMP
  - Simple Network Management Protocol
    - Protocol for network monitoring and management
    - Structured protocol and structured information
    - For querying network device state and receiving notifications
    - Can be used to change state
    - Industry standard, hundreds of tools uses it
    - Supported on any decent network devices
    - Transport: UDP ports 161 and 162

Uses of SNMP

- Uses
  - Typical queries
    - Bytes In/Out on an interface, errors
    - CPU load
    - Uptime
    - Temperature or other vendor specific OIDs
      - *snmpget -Os -c public -v 2c 192.248.1.1 system.sysName.0*
      - *snmpget -Os -c public -v 2c 192.248.1.1 ifOutOctets.1*
  - In case of hosts (servers)
    - Disk space
    - Installed software
    - Running process
    - Load average, etc
      - *snmpget -Os -c public -v 2c 192.248.1.165 system.sysName.0*
      - *snmpget -Os -c public -v 2c 192.248.1.165 hrStorageDescr.34*
      - *snmpget -Os -c public -v 2c 192.248.1.165 hrStorageSize.34*
      - *snmpget -Os -c public -v 2c 192.248.1.165 hrStorageUsed.34*

SNMP History

- 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 (simple community string model)
    - Historic
- v3 (1998) Robust security

SNMP roles

- Manger
  - It is the monitoring station, sometime known as the SNMP client
  - SNMPv3 calls it the Command Generator and Notification Receiver
- Agent
  - It is running on the equipment/server, sometimes know as the SNMP server
  - SNMPv3 calls it the Command Responder and Notification Originator

How does SNMP work

- 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 the threshold

How does SNMP work

- Query/response based
  - Monitors generally uses **get, getnext, getbulk**
    - e.g. monitors: linux snmp-tools, nagios, cacti, etc
  - Change state uses set
  - Response is always a **getresponse**
  - **getbulk** requires v2c or v3

- Notification are delivered as **traps**
  - **traps** are unacknowledged
  - **informs** are acknowledged (v2c, v3)

The SNMP database

- Then 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
  - OID are organized into a tree structure that is the MIB
  - MIB files documents parts of the MIB on a device
- OID
  - 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!.
    - e.g.
      - 1.3.6.1.2.1.1.3
  - Allocated hierarchically in a tree to ensure uniqueness (similar to DNS)

How does SNMP work

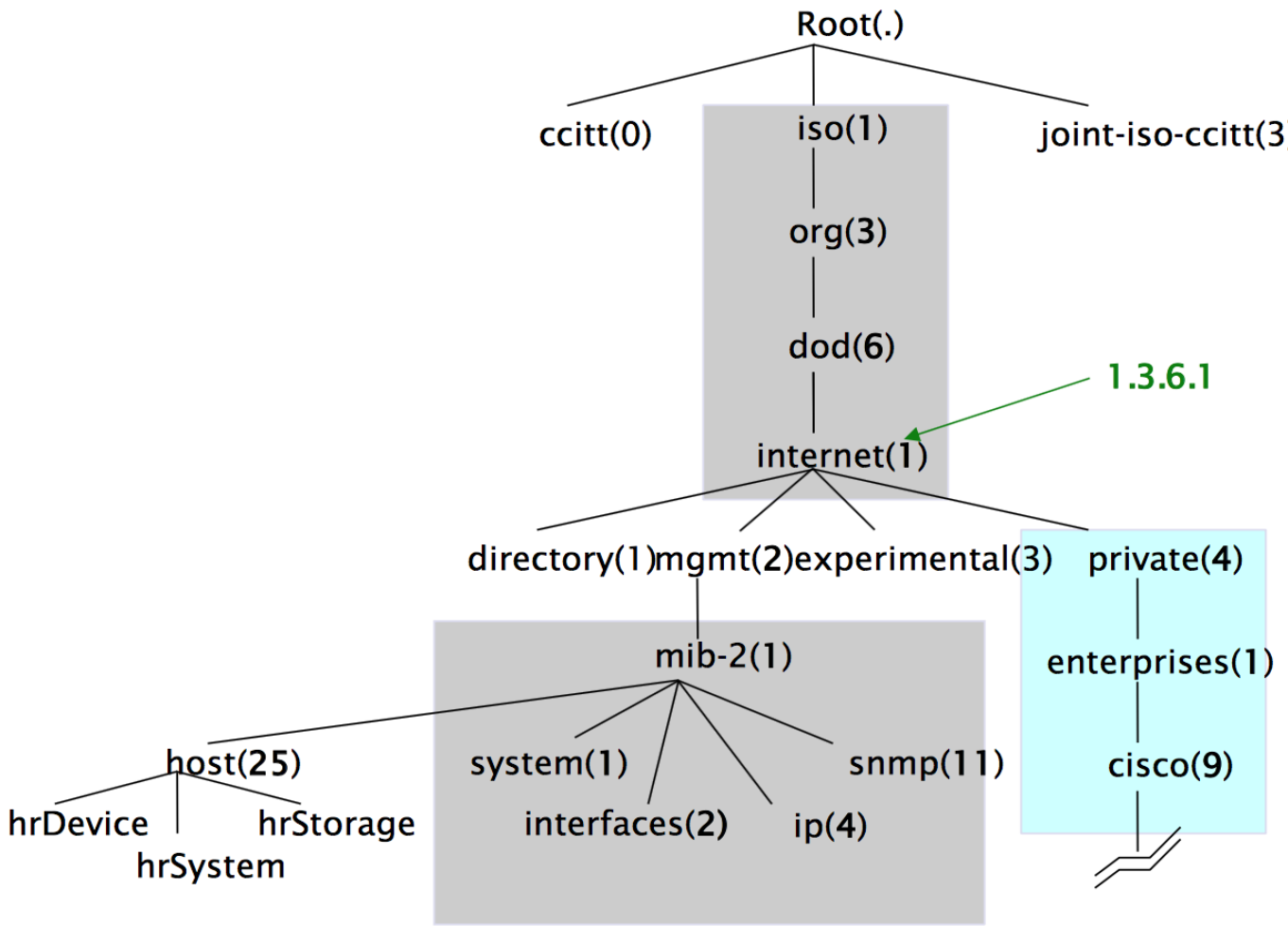
- Query/response based
  - Monitors generally uses **get, getnext, getbulk**
    - e.g. monitors: linux snmp-tools, nagios, cacti, etc
  - Change state uses set
  - Response is always a **getresponse**
  - **getbulk** requires v2c or v3

- Notification are delivered as **traps**
  - **traps** are unacknowledged
  - **informs** are acknowledged (v2c, v3)

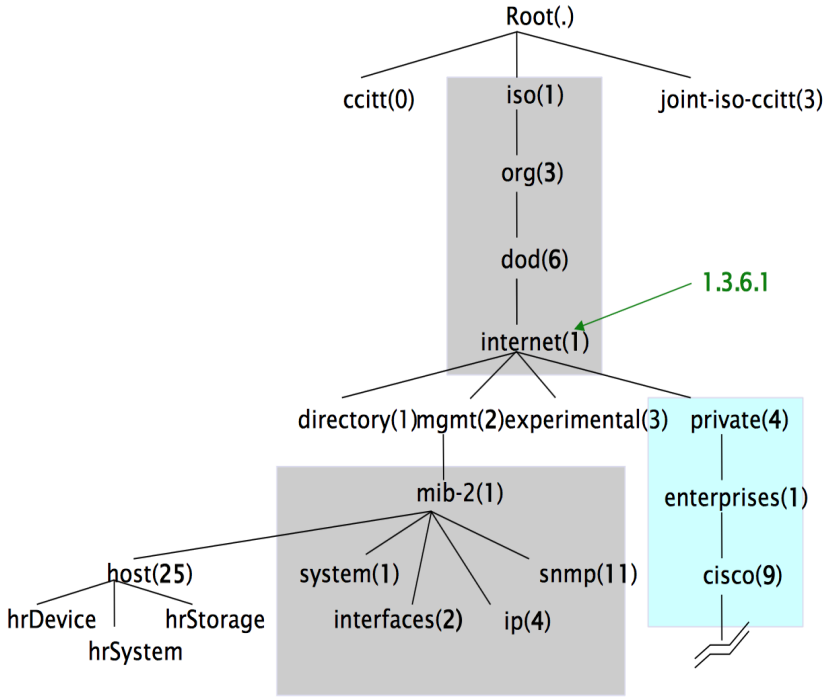
OIDs and MIB files

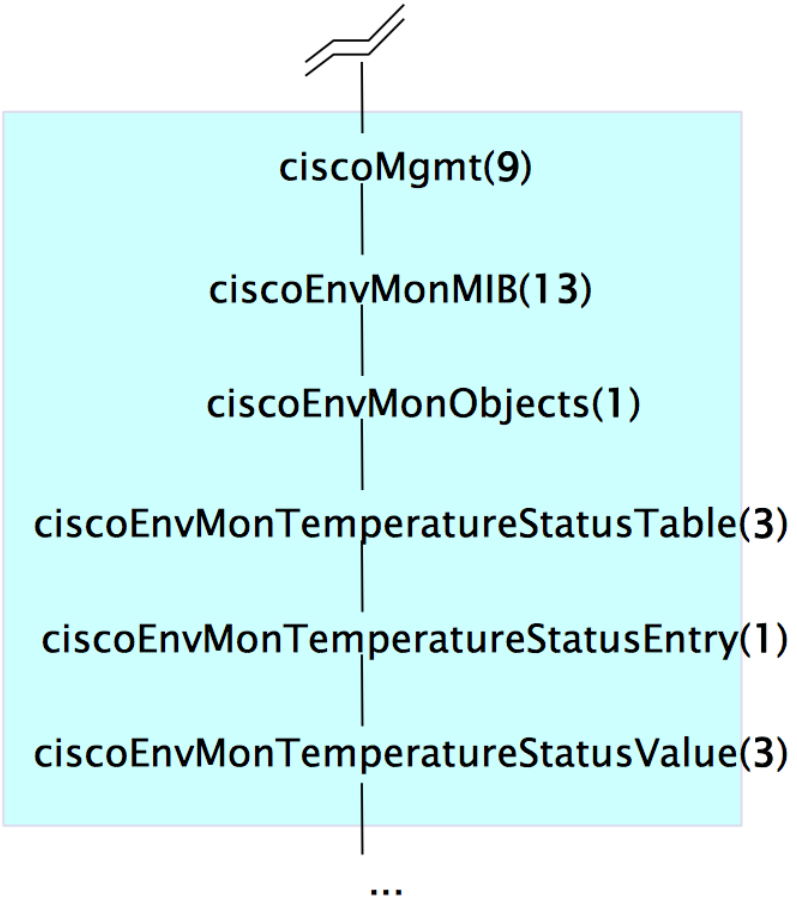
- For example: user@learn.ac.lk
  - would have been something like
    - user@learn.enterprises.private.internet.dod.org.iso
    - user@99988.1.4.1.6.3.1
      - except that we reverse the ordering and putting iso(1) first
    - .1.3.6.1.4.1.99988.117.115.101.114 - It is unique..
  - Read from left to right
  - OID components seperated 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.mib-2.system.sysName
  - How do we convert from OIDs to Lables (and vice versa) ?
    - Use of MIB files!

The MIB Tree


Root(.)
ccitt(0)
iso(1)
joint-iso-ccitt(3)
org(3)
dod(6)
1.3.6.1
internet(1)
directory(1)
mgmt(2)
experimental(3)
private(4)
mib-2(1)
enterprises(1)
host(25)
system(1)
snmp(11)
cisco(9)
hrDevice
hrStorage
hrSystem
interfaces(2)
ip(4)

The MIB Tree


Root(.)
ccitt(0)
iso(1)
joint-iso-ccitt(3)
org(3)
dod(6)
1.3.6.1
internet(1)
directory(1)
mgmt(2)
experimental(3)
private(4)
mib-2(1)
enterprises(1)
host(25)
system(1)
snmp(11)
cisco(9)
hrDevice
hrStorage
hrSystem
interfaces(2)
ip(4)


ciscoMgmt(9)
ciscoEnvMonMIB(13)
ciscoEnvMonObjects(1)
ciscoEnvMonTemperatureStatusTable(3)
ciscoEnvMonTemperatureStatusEntry(1)
ciscoEnvMonTemperatureStatusValue(3)
...

Interesting parts of the MIB Tree

- The Internet MIB, .1.3.6.1, really only two branches of intersets
  - Standard MIBs
    - .1.3.6.1.2.1 = .iso.org.dod.internet.mgmnt.mib-2
  - Vendor-specific (proprietary) MIBs
    - .1.3.6.1.4.1. = .iso.org.dod.internet.private.enterprises