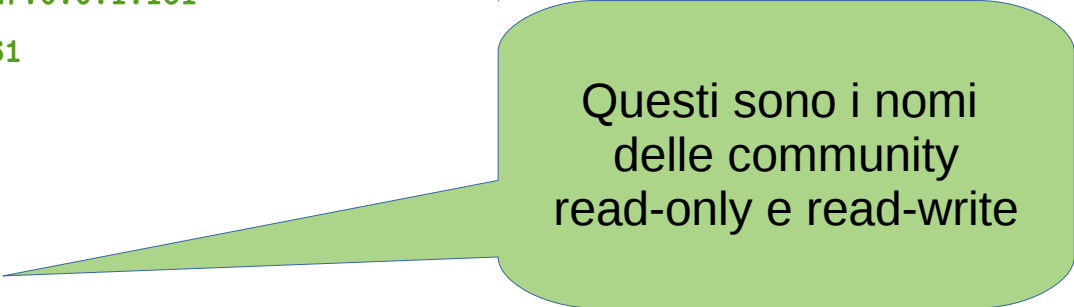


# Lato manager – Linux CLI

- Per interrogare agent SNMP da riga di comando:
  - **snmpget** recupero di un singolo oggetto
  - **snmpset** impostazione del valore di un oggetto
  - **snmpwalk** utilizza ricorsivamente la PDU *getNext* per navigare un intero sottoalbero del MIB
- Questi comandi hanno moltissimi parametri in comune
  - le man page di ognuno documentano solo le opzioni specifiche
  - la man page **snmpcmd** documenta quelle comuni; essenziali in ogni invocazione:
    - v versione
    - c community
    - indirizzo del network element
    - OID del managed object
- Nel file **/etc/snmp/snmp.conf** (utilizziamo la **VM Client**) si possono inserire i valori di alcuni parametri per configurare il comportamento di default dei tool
  - per mostrare i nomi simbolici degli OID, commentare la riga **mibs : "**

# Lato agent – Linux daemon

- L'agent è il demone `snmpd` (**utilizziamo la VM Server**)
- Il file `/etc/snmp/snmpd.conf` configura l'agent; la sezione iniziale definisce
  - socket
    - sostituire `agentAddress udp:127.0.0.1:161`  
con `agentAddress udp:161`
  - controllo dell'accesso
    - abilitare le community per SNMPv1
      - `rocommunity public`
      - `rwcommunity supercom`
  - alcuni managed object di sistema
    - es. `sysLocation`, `sysContact`
    - NOTA: questi sono oggetti RW (verificate sugli standard) solo se non settati nel file di configurazione
      - **commentiamo la definizione di `sysLocation`**
- Dopo la configurazione
  - avviarlo con `systemctl start snmpd`
  - verificare con `ps -fC snmpd` i dettagli del processo
  - cercare con `ss` o `lsof` le socket di rete utilizzate



Questi sono i nomi  
delle community  
read-only e read-write

# Esempi (da Client)

- Es. 1

- `snmpget -v 1 -c public 192.168.56.203 .1.3.6.1.2.1.1.4.0`  
(vediamo il valore impostato nel file di configurazione?)

- Es. 2

- `snmpwalk -On -v 1 -c public 192.168.56.203 .1.3.6.1.2.1.1`
- se togliamo `-On` l'output è più leggibile (ma meno processabile)

- Es. 3

- avviamo in un altro terminale `tcpdump -nlp -i lo udp port 161`
- ripetiamo `snmpwalk`

- Es. 4

- `snmpget -v 1 -c public 192.168.56.203 .1.3.6.1.2.1.1.6.0`  
(avendolo commentato, non ha valore)
- `snmpset -v 1 -c supercom 192.168.56.203 .1.3.6.1.2.1.1.6.0 s "proprio qui"`
- ripetiamo `snmpget`

# Misura di parametri di sistema

- ancora in `/etc/snmp/snmpd.conf`, si possono attivare direttive di monitoraggio dei parametri base del sistema
- estensione UCD-SNMP
  - `load [max-1] [max-5] [max-15]`
    - tabella .1.3.6.1.4.1.2021.10
    - tre righe (carico negli ultimi 1-5-15 minuti)
    - colonne: carico effettivo, flag di superamento delle rispettive soglie
  - `disk [partizione] [minfree|minfree%]`
    - tabella .1.3.6.1.4.1.2021.9
    - una riga per ogni partizione messa sotto controllo da una direttiva disk
    - colonne: tutti i dettagli della partizione e flag di spazio sotto il minimo
  - `proc [nomeprocesso] [maxnum [minnum]]`
    - tabella .1.3.6.1.4.1.2021.2
    - una riga per ogni processo messo sotto controllo da una direttiva proc
    - colonne: numero di istanze, flag di superamento delle soglie
  - vedere la documentazione completa sul sito del corso

# Esecuzione di codice remoto

- ancora in `/etc/snmp/snmpd.conf`, si possono inserire direttive per eseguire codice il cui output è reso accessibile come managed object
- estensione NET-EXTEND

- tabella `NET-SNMP-EXTEND-MIB::`
- righe con nome = etichetta della direttiva extend-sh
- diverse colonne, la più comune: `nsExtendOutputFull`
- vedere la documentazione sito del corso
- es:

```
extend-sh test1 echo HelloWorld
```

OID corrispondente: `NET-SNMP-EXTEND-MIB::nsExtendOutputFull."test1"`

(notare i doppi apici, indicano all'agent che è un nome da risolvere, non un segmento di OID standard – attenzione all'espansione bash, devono arrivare al comando snmp!)

- quando l'agent riceve una getRequest per l'OID corrispondente
  - esegue il comando
  - restituisce l'output nella Response

# Esercizi

- **[superagent.sh]** Supponiamo di voler recuperare via SNMP l'elenco delle regole di iptables di un host Linux
  - che ostacoli incontro?
  - qual è il modo corretto di superarli?
- Modificare gli esercizi fatti in precedenza per usare SNMP invece del lancio di un comando remoto via SSH:
  - **sshnum.sh**
  - **log\_user.sh** (dell'esercizio "netmon")
- **[pivot.sh]** Supponiamo di avere un agent con un numero imprecisato di processi sorvegliati da direttive proc
  - come mi procuro via SNMP il numero di istanze di un processo di cui conosco il nome?
  - come verifico rapidamente, ai fini di un test in uno script (es. if o while) se il numero di istanze attive è entro i limiti prestabiliti?