Network Monitoring and Management

Cisco Configuration

Topics

- CLI modes
- Accessing the configuration
- Basic configuration (hostname and DNS)
- Authentication and authorization (AAA)
- Log collection
- Time Synchronization (date/timezone)
- SNMP configuration
- Cisco Discovery Protocol (CDP)

CLI Modes

- User EXEC
  - Limited access to the router
  - Can show some information but cannot view nor change configuration

```
rtr>
```

- Privileged EXEC
  - Full view of the router's status, troubleshooting, manipulate config, etc.

```
rtr> enable
rtr#
```

Accessing the router

- Before setting up SSH
  - telnet 10.10.x.254
  - login “cisco” and “cisco” (user and password)
- Privileged user can go to privileged mode:
  - `rtr> enable (default password is “cisco”)`
  - `rtr# configure terminal`
  - `rtr(config)#`
- Type in configuration commands
- Exit and save the new configuration
  - `rtr(config)# exit`
  - `rtr# write memory`

Accessing the configuration

- There are two configurations:
  - *Running config* is the actual configuration that is active on the router and Stored in RAM (will be gone if router is rebooted):

```
rtr# configure terminal     (conf t)
rtr(config)# end
rtr# show running-config
```

  - *Startup config*

    Stored in NVRAM (Non-Volatile RAM):

```
rtr# copy running-config startup-config    (or)
rtr# write memory                  (wr mem)
rtr# show startup-config           (sh start)
```

Basic configuration (hostname and DNS)

- Assign a name
  - `rtr(config)# hostname rtrX`
- Assign a domain
  - `rtr(config)# ip domain-name ws.nsrc.org`
- Assign a DNS server
  - `rtr(config)# ip name-server 10.10.0.250`
- Or, disable DNS resolution
  - `rtr(config)# no ip domain-lookup`

  if no dns this is *very useful* to avoid long waits

Authentication and autorization

Configure passwords in the most secure manner.

- Use the improved method which uses hash function

  Example:

```
#enable secret 0 cisco
#user admin secret 0 cisco
```

Authentication and autorization

Configuring SSH with a 2048 byte key (at least 768 for OpenSSH clients)

```
rtr(config)# aaa new-model
rtr(config)# crypto key generate rsa  (key size prompt)
```

Verify key creation:

```
rtr# show crypto key mypubkey rsa
```

Restrict to only use SSH version 2. Optionally register events:

```
rtr(config)# ip ssh logging events
rtr(config)# ip ssh version 2
```

Use SSH, disable *telnet* (only use telnet if no other option*)*

```
rtr(config)# line vty 0 4
rtr(config)# transport input ssh
```

Log collection (syslog*)

Send logs to the *syslog* server:

```
rtr# logging 10.10.x.x
```

Identify what channel will be used (local0 to local7):

```
rtr# logging facility local5
```

Up to what priority level do you wish to record?

```
rtr# logging trap <logging_level>
```

<0-7>	**Logging severity level**	
emergencies	System is unusable	(severity=0)
alerts	Immediate action needed	(severity=1)
critical	Critical conditions	(severity=2)
errors	Error conditions	(severity=3)
warnings	Warning conditions	(severity=4)
notifications	Normal but significant conditions	(severity=5)
informational	Informational messages	(severity=6)
debugging	Debugging messages	(severity=7)

*syslog, syslog-ng, ryslog

Time synchronization

It is essential that all devices in our network are time-synchronized

In config mode:

```
rtr# ntp server pool.ntp.org
rtr# clock timezone <timezone>
```

To use UTC time

```
rtr# no clock timezone
```

If your site observes daylight savings time you can do:

```
rtr# clock summer-time recurring last Sun Mar 2:00 last Sun Oct  3:00
```

Verify

```
rtr# show clock
      22:30:27.598 UTC Tue Feb 15 2011
rtr# show ntp status
       Clock is synchronized, stratum 3, reference is 4.79.132.217
       nominal freq is 250.0000 Hz, actual freq is 249.9999 Hz, precision is 2**18
       reference time is D002CE85.D35E87B9 (11:21:09.825 CMT Tue Aug 3 2010)
       clock offset is 2.5939 msec, root delay is 109.73 msec
       root dispersion is 39.40 msec, peer dispersion is 2.20 msec
```

SNMP configuration

Start with SNMP version 2

- It's easier to configure and understand
- Example:

```
rtr(config)# snmp-server community public ro 99
r10(config)# access-list 99 permit 10.10.0.0 0.0.0.255
```

SNMP configuration

From a Linux machine (once snmp utils are installed), try:

```
snmpwalk -v2c -c public 10.10.x254 sysDescr
```

Cisco Discovery Protocol (CDP)

Enabled by default in most modern routers

If it's not enabled:

```
rtr# cdp enable
rtr# cdp run          (in older CISCO IOS versions)
```

To see existing neighbors:

```
rtr# show cdp neighbors
```

Tools to visualize/view CDP announcements:

- *tcpdump*
- *cdpr*
- *wireshark*
- *tshark*

Enabling NetFlow (traffic flow export)

Configure FastExthernet 0/0 to generate netflow and export flows to 10.10.0.250 on port 9996:

```
rtr# configure terminal
rtr# interface FastEthernet 0/0
rtr(config-if)# ip flow ingress
rtr(config-if)# ip flow egress
rtr(config-if)# exit
rtr(config-if)# ip flow-export destination 10.10.1.1 9996
rtr(config-if)# ip flow-export version 5
rtr(config-if)# ip flow-cache timeout active 5
```

This breaks up long-lived flows into 5-minute fragments. You can choose any number of minutes between 1 and 60. If you leave it at the default of 30 minutes your traffic reports will have spikes.

Enabling NetFlow cont.

```
rtr(config)# snmp-server ifindex persist
```

This enables ifIndex persistence globally. This ensures that the ifIndex values are persisted during router reboots.

Now configure how you want the ip flow top-talkers to work:

```
rtr(config)#ip flow-top-talkers
rtr(config-flow-top-talkers)#top 20
rtr(config-flow-top-talkers)#sort-by bytes
rtr(config-flow-top-talkers)#end
```

Now we'll verify what we've done

```
rtr# show ip flow export
rt# show ip cache flow
```

See your "top talkers" across your router interfaces

```
rtr# show ip flow top-talkers
```

Questions?

?