Cisco Configuration

Network Startup Resource Center

www.ws.nsrc.org

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Topics

- CLI modes
- Accessing the configuration
- Basic configuration (hostname and DNS)
- Authentication and authorization (AAA)
- SSH

CLI Modes

User EXEC

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

```
bdr1.campusY>
```

Privileged EXEC

- Full view of the router’s status, troubleshooting, manipulate config, etc.

```
bdr1.campusY> enable
bdr1.campusY#
```

Accessing the router (first time)

Before setting up SSH

- telnet to a Cisco network device, or use its console
- *(You will be given <USER> and <PASS> for class)*

Privileged user can go to privileged mode:

```
bdr1.campusY> enable  (enter <PASS> default is “cisco”)
bdr1.campusY# configure terminal
bdr1.campusY(config)#
```

Accessing the router (first time)

Now that you are in “config” mode you can adjust router settings. When done:

Exit and save the new configuration

```
bdr1.campusY(config)# end
bdr1.campusY# write memory
```

- If you don’t “`wr  mem`” (write memory) changes are lost if router reboots.
- We have added a space between “#” and commands for clarity. On the router there is no space.

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        (show run)
```

- ***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)
```

*****For simplicity we use “`rtr`” for the remainder of this presentation vs. “`bdr1.campus`”

Basic configuration (hostname and DNS)

- **Assign a name**

```
rtr(config)# hostname bdr1.campusY
```

or (for example)

```
rtr(config)# hostname core1.campusY
```

- **Assign a domain**

```
rtr(config)# ip domain-name ws.nsrc.org
```

- **Assign a DNS server**

```
rtr(config)# ip name-server 192.168.122.1
```

- Or, **disable DNS resolution**

```
rtr(config)# no ip domain-lookup
```

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

Authentication & authorization

Configuring passwords:

- Passwords stored as a hash

Example:

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

In class we use different user names and passwords.

Enabling SSH access

Configuring SSH with a 2048 bit host 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
```

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

```
    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
```

Questions?

?

End of presentation!

- Now do the Cisco configuration lab

Following slides for reference only

- You'll be introduced to these commands in later presentations and labs

Log collection (syslog*)

Send logs to the *syslog* server:

```
      rtr(config)# logging 100.68.Y.130.  (example)
```

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

```
      rtr(config)# logging facility local5
```

Up to what priority level do you wish to record?

```
     rtr(config)# 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, rsyslog

Time synchronization

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

In config mode:

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

To use UTC time:

```
rtr(config)# no clock timezone
```

If your site observes daylight savings time you can do:

```
rtr(config)# 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…
```

SNMP configuration

Start with SNMP version 2

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

```
rtr(config)# snmp-server community NetManage ro 99
rtr(config)# access-list 99 permit 100.68.Y.128 0.0.0.15
rtr(config)# access-list 99 permit 100.64.0.0 0.0.3.255
```

Note the Cisco subnet mask inversion:

```
0.0.3.255       == 255.255.252.0     == /22  (1022 hosts)
0.0.0.15  == 255.255.255.240   == /28  (14 hosts)
```

SNMP configuration

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

```
snmpwalk –v2c –c NetManage bdrX.campusY.ws.nsrc.org sysDescr
```

Cisco Discovery Protocol (CDP)*

Enabled by default in most modern routers -

If it’s not enabled (don’t!):

```
rtr(config)# cdp run
rtr(config-if)# cdp enable(per-interface)
```

To see existing neighbors:

```
rtr# show cdp neighbors
```

Tools to visualize/view CDP announcements:

tcpdump

cdpr

wireshark

tshark

* As discovered in February of 2020, CDP is a serious security risk due to CDPwn. Patch your switches and routers first.

Enabling NetFlow flows version 5

Configure version 5 NetFlow flows on GigabithEthernet interface 0/0 and export them to 100.68.Y.130 on port 9996:

```
rtr# configure terminal
rtr(config)# interface GigabithEthernet0/0
rtr(config-if)# ip flow ingress
rtr(config-if)# ip flow egress
rtr(config-if)# exit
rtr(config-if)# ip flow-export destination 100.68.Y.130 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.

Note: Newer version of Cisco IOS have changed this syntax.

Enabling top-talkers NetFlow
(version 5)

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

Ensures that the ifIndex values are retained over router reboots or if you add/remove interface modules.

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
```

Verify what we've done

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

See your "top talkers" across your router interfaces:

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

Enabling NetFlow IPv4 flows
(version 9)

Configure version 9 NetFlow flows for IPv4 on GigabitEthernet interface 0/0 and export them to 100.68.Y.130 on port 9996:

```
rtr# configure terminal
rtr(config)# flow exporter EXPORTER-1
rtr(config-flow-exporter)# description Export to DB Server CampusY
rtr(config-flow-exporter)# destination 100.68.Y.130
rtr(config-flow-exporter)# transport udp 9996
rtr(config-flow-exporter)# template data timeout 300
rtr(config-flow-exporter)# flow monitor FLOW-MONITOR-V4
rtr(config-flow-monitor)# exporter EXPORTER-1
rtr(config-flow-monitor)# record netflow ipv4 original-input
rtr(config-flow-monitor)# cache timeout active 300
rtr(config)# snmp-server ifindex persist
rtr(config)# interface GigabitEthernet0/0
rtr(config-if)# ip flow monitor FLOW-MONITOR-V4 input
rtr(config-if)# ip flow monitor FLOW-MONITOR-V4 output
rtr(config-if)# exit
rtr# write memory
```

Enabling NetFlow IPv6 flows
(version 9)

Configure version 9 NetFlow flows for IPv6:

To monitor IPv6 flows you would have to create a new flow monitor for IPv6 and attach it to the interface and the existing exporters.

```
rtr(config-flow-exporter)# flow monitor FLOW-MONITOR-V6
rtr(config-flow-monitor)# exporter EXPORTER-1
rtr(config-flow-monitor)# record netflow ipv6 original-input
rtr(config-flow-monitor)# cache timeout active 300
rtr(config)# interface GigabitEthernet0/0
rtr(config-if)# ipv6 flow monitor FLOW-MONITOR-V6 input
rtr(config-if)# ipv6 flow monitor FLOW-MONITOR-V6 output
rtr(config-if)# exit
rtr# write memory
```

Viewing NetFlow flows
(version 9)

These are not configuration directives, just a few samples of viewing flow information directly on your router.

To view your current configuration:

```
rtr# show flow exporter EXPORTER-1
rtr# show flow monitor FLOW-MONITOR-V4
```

It’s possible to see active individual flows on the device:

```
rtr# show flow monitor FLOW-MONITOR-V4 cache
```

Will display too many flows. Press ‘q’ to exit display. Group flows so you can see your “Top Talkers” by traffic destinations and sources. This is one long command:

```
rtr# show flow monitor FLOW-MONITOR-V4 cache aggregate ipv4 \
     source address ipv4 destination address sort counter   \
     bytes top 20
```