Chapter 7

Configuring Remote Access to a C-series Platform

This chapter describes how to configure access to a C-series platform. Topics include:

Configuring External Interfaces on a C-series Platform

The C-series platform provides the following interfaces:

- Serial port—9600 baud

  The serial port is enabled by default. You can use the serial port to connect to a console terminal and perform initial configuration as well as configuration updates.

- Two external Gigabit Ethernet interfaces—eth0 and eth1

  The eth0 interface is designed to provide access from a network that is behind a firewall. This interface accepts connections from protocols supported by the SRC software. When you configure an SRC component, the specified port is opened on this interface.

  The eth1 interface is designed to provide access for applications on an external network, such as the Internet. You can configure a limited number of ports on this interface. By default, no inbound ports are open.

- Optional two additional Gigabit Ethernet interfaces—eth2 and eth3

  These interfaces require an additional input/output module. You can obtain a module to support either RJ-45 or optical connections.

- Two USB interfaces

Configuring Gigabit Ethernet Interfaces

Configure the Gigabit Ethernet interfaces to allow remote access to the C-series platform. You can specify an IP address with mask or a broadcast address with mask for an interface.

Use the following configuration statements to configure Gigabit Ethernet interfaces and the [edit] hierarchy level:

```
interfaces name unit unit-number

interfaces name unit unit-number family inet {
    address address;
    broadcast broadcast;
}
```

To configure a Gigabit Ethernet interface:

1. From configuration mode, access the configuration statement that configures the interface.

   ```
   [edit]
   user@host# edit interfaces name unit unit-number
   ```

   where *unit-number* is a number that you can assign for a logical interface identifier.

   For example:

   ```
   [edit]
   user@host# edit interfaces eth0
   ```

2. Specify the unit, family, and IP address for the interface.

   ```
   [edit interfaces eth0]
   user@host# set unit number family inet address address
   ```

   For example, to configure an interface with only an IP address:

   ```
   [edit interfaces eth0]
   user@host# set unit 0 family inet address 192.2.0.10/24
   ```

3. (Optional) Specify the unit, family, and broadcast address for the interface.

   ```
   [edit interfaces eth0]
   user@host# set unit number family inet broadcast broadcast
   ```

   For example, to configure an interface with a broadcast IP address:

   ```
   [edit interfaces eth0]
   user@host# set unit 0 family inet address 192.2.0.20/24
   ```

4. Verify the interface configuration.

   ```
   [edit interfaces eth0]
   user@host# show
   unit 0 {
     family {
       inet {
         address 192.2.0.10/24;
       }
   ```

Configuring Tunnel Interfaces

A tunnel allows direct connection between a remote location and an application running on the C-series platform; a tunnel lets you use the redirect server in deployments where the JUNOSe router does not have a direct connection to the C-series platform.

The C-series platform supports two types of tunnel interfaces:

- GRE—Encapsulates traffic that can use various network protocols within IP. For C-series platforms, the tunnel interface encapsulates IP packets.
- IP-over-IP—Encapsulates IP packets within IP packets.

The other endpoint for the tunnel on a JUNOS or JUNOSE router must be configured for the tunnel to be operational.

Use the following configuration statements to configure tunnel interfaces at the [edit] hierarchy level:

```
interfaces name unit unit-number tunnel {
    mode (ipip | gre);
    destination destination;
    source source;
    key key;
    interface interface;
    ttl ttl;
}

interfaces name unit unit-number family inet {
    address address;
}
```

To configure a tunnel interface on a C-series platform:

1. From configuration mode, access the configuration statement that configures tunnel interfaces.

   ```
   [edit]
   user@host# edit interfaces name unit unit-number tunnel
   ```

   For example:

   ```
   [edit]
   user@host# edit interfaces ip-tunnel unit t0 tunnel
   ```

2. Configure the type of tunnel, IP-over-IP or GRE.

   ```
   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# set mode ipip
   ```

   or

   ```
   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# set mode gre
   ```

3. Specify the IP address of the remote end of the tunnel.

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set destination** *destination*

   For example:

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set destination 192.0.2.20**

4. (Optional) Specify an IP address that will not change to receive tunneled packets.

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set source** *source*

   For example:

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set source 192.20.10.5**

   If you specify a source address, Step 6 is required.

5. (Optional) For a GRE tunnel, specify a key.

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set key** *key*

   For example:

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set key 250**

6. (Optional. Required if you specify a source address.) Specify an existing physical interface on the C-series platform.

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set interface** *interface*

   For example:

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set interface eth0**

7. (Optional) Specify the lifetime of tunneled packets.

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set ttl** *ttl*

   For example:

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **set ttl 110**

8. Configure an IP address for the tunnel interface. This IP address is used to connect to a device at the other end of the tunnel. For example:

   [edit interfaces ip-tunnel unit t0 tunnel]
   user@host# **up**
   [edit interfaces ip-tunnel unit t0]
   user@host# **edit family inet**
   [edit interfaces ip-tunnel unit t0 family inet]
   user@host# **set address 10.0.1.1/24**

9. Verify the configuration by running the show command. For example:

```
[edit interfaces]
user@host# show
ip-tunnel {
  unit t0 {
    family {
      inet {
        address 10.0.1.1/24;
      }
    }
    tunnel {
      mode ipip;
      destination 192.0.2.20;
      source 192.20.10.5;
      interface eth0;
      ttl 110;
    }
  }
}
```

Configuring a Static Route to Devices on Other Networks

In some instances, the SRC software might need to connect to devices that reside on networks other than the one that the SRC software accesses directly. You can configure a static route for the software to be able to connect devices on other networks.

When you specify IP addresses for a static route, include a network mask.

To configure a static route to another network:

- From configuration mode, enter the following command at the top level of the hierarchy.

  [edit]
  user@host# **set routing-options static route** *destination* **next-hop** *next-hop*

  The next-hop option is required.

You can also specify that packets to the specified destination be dropped and that an ICMP unreachable message be returned.

To specify that packets to a specified network be dropped:

- From configuration mode, enter the following command at the top level of the hierarchy.

  [edit]
  user@host# **set routing-options static route** *destination* **next-hop** *next-hop* **reject**

Securing Connections Between a C-series Platform and Remote Hosts

For security reasons, take care to limit the number of open ports you configure for applications and SRC components on the external interfaces. To review the default port settings for SRC components, see *Chapter 29, Defining an Initial Configuration on a Solaris Platform* which provides information about an initial configuration on a Solaris platform.

By default, SSH for nonwhite users is enabled on C-series platforms. Otherwise, you configure the C-series platform to explicitly allow users on remote systems to access it. Table 8 lists the applications through which remote users can access a C-series platform.

Table 8: Applications to Remotely Access the C-series Platform

Application	Information About Access Configuration
SSH	*Configuring a C-series Platform to Accept SSH Connections* on page 68
Telnet	*Configuring a C-series Platform to Accept Telnet Connections* on page 69
NETCONF	*Configuring a C-series Platform to Accept NETCONF Connections* on page 69
C-Web interface	*Chapter 6, Accessing and Starting the C-Web Interface*
Policies, Services, and Subscribers CLI	*Chapter 5, Accessing and Starting the SRC CLI*

You can also configure security certificates for use by HTTPS connections.

See *Chapter 7, Configuring Remote Access to a C-series Platform*.

You can connect from a C-series platform to remote hosts through:

- SSH
- Telnet
- FTP by means of a file URL

Configuring a C-series Platform to Accept SSH Connections

You can enable SSH to let users who have the appropriate privileges connect to a C-series platform. For security reasons, we recommend that you do not allow remote users to access the CLI as `root`.

Use the following configuration statements to enable SSH access from the [edit] hierarchy level:

```
system services ssh {
    root-login (allow | deny | deny-password);
    protocol-version (v1 | v2);
}
```

To configure the C-series platform to accept SSH connections:

1. From configuration mode, access the [edit system services ssh] hierarchy level.
2. (Optional) Specify that SSH version 1 be used.

   ```
   [edit system services ssh]
   user@host> set protocol-version v1
   ```

   SSH version 2 is enabled by default.
3. (Optional) Specify whether or not to allow root login through SSH:

   ```
   [edit system services ssh]
   user@host> set root-login (allow | deny | deny-password)
   ```

   where:

   - allow—Allow users to log in to the C-series platform as `root` through SSH.
   - deny—Disable users from logging in to the C-series platform as `root` through SSH.
   - deny-password—Allow users to log in to the C-series platform as `root` through SSH when the authentication method (for example, RSA authentication) does not require a password. (Default)

Configuring a C-series Platform to Accept Telnet Connections

You can enable Telnet to let users who have the appropriate privileges connect to a C-series platform. The system does not allow `root` access over a Telnet connection.

Use the following configuration statements to enable Telnet access from the [edit] hierarchy level:

```
system services {
    telnet;
}
```

To configure the C-series platform to accept Telnet connections:

```
[edit]
user@host# set system services telnet
```

Configuring a C-series Platform to Accept NETCONF Connections

Use the following configuration statements to enable NETCONF access from the [edit] hierarchy level:

```
system services netconf {
    ssh;
}
```

To configure the C-series platform to accept NETCONF connections:

1. From configuration mode, access the [edit system services netconf] hierarchy level.

   ```
   [edit]
   user@host# edit system services netconf
   ```

2. (Optional) Enable NETCONF to run over SSH.

   ```
   [edit system services netconf]
   user@host# set ssh
   ```