

Chapter 20

Configure User Access

To configure user access, you do the following:

- Define Login Classes on page 205

- Configure User Accounts on page 210

Define Login Classes

All users who can log into the router must be in a login class. With login classes, you define the following:

- Access privileges users have when they are logged into the router

- Commands and statements that users can and cannot specify

- How long a login session can be idle before it times out and the user is logged off

You can define any number of login classes. You then apply one login class to an individual user account as described in “Configure User Accounts” on page 210.

To define a login class and its access privileges, include the class statement at the [edit system login] hierarchy level:

```
[edit system]
login {
  class class-name {
    allow-commands "regular-expression";
    deny-commands "regular-expression";
    idle-timeout minutes;
    permissions [ permissions ];
  }
}
```

In *class-name*, you name the login class. The software contains a few predefined login classes, which are listed in Table 8 on page 207. The predefined login classes cannot be modified.

For each login class, you can do the following:

- Configure Access Privilege Levels on page 206

- Deny or Allow Individual Commands on page 208

- Configure the Timeout Value for Idle Login Sessions on page 210

Configure Access Privilege Levels

Each top-level CLI command and each configuration statement has an access privilege level associated with it. Users can execute only those commands and configure and view only those statements for which they have access privileges. The privilege level for each command and statement is listed in the summary chapter of the part in which that command or statement is described. The access privileges for each login class are defined by one or more *permission bits*.

To configure access privilege levels, include the permissions statement at the [edit system login class] hierarchy level:

```
[edit system login class]
permissions [ permissions ];
```

In *permissions*, specify one or more of the permission bits listed in Table 7. Permission bits are not cumulative, so for each class list all the bits needed, including view to display information and configure to enter configuration mode. For the permissions that control the individual parts of the configuration, there are two forms of the permission:

- “Plain” form—Provides read-only capability for that permission type. An example is interface.

- Form that ends in -control—Provides read and write capability for that permission type. An example is interface-control.

Table 7: Login Class Permission Bits

Permission Bit	Description
admin	Can view user account information in configuration mode and with the show configuration command.
admin-control	Can view user accounts and configure them (at the [edit system login] hierarchy level).
all	Has all permissions.
clear	Can clear (delete) information learned from the network that is stored in various network databases (using the clear commands).
configure	Can enter configuration mode (using the configure command) and commit configurations (using the commit command).
control	Can perform all control-level operations (all operations configured with the -control permission bits).
edit	Can edit all portions of a configuration, can load a configuration from an ASCII file, and can commit new and modified configurations (using all the commands in configuration mode).
field	Reserved for field (debugging) support.
firewall	Can view the firewall filter configuration in configuration mode.

Permission Bit	Description
firewall-control	Can view and configure firewall filter information (at the [edit firewall] hierarchy level).
floppy	Can read from and write to the removable media.
interface	Can view the interface configuration in configuration mode and with the show configuration operational mode command.
interface-control	Can view interface configuration information and configure interfaces (at the [edit interfaces] hierarchy level).
maintenance	Can perform system maintenance, including starting a local shell on the router and becoming the superuser in the shell (by issuing the su root command), and can halt and reboot the router (using the request system commands).
network	Can access the network by entering the ping, ssh, telnet, and traceroute commands.
reset	Can restart software processes using the restart command and can configure whether software processes are enabled or disabled (at the [edit system processes] hierarchy level).
rollback	Can use the rollback command to return to a previously committed configuration other than the most recently committed one.
routing	Can view general routing, routing protocol, and routing policy configuration information in configuration and operational modes.
routing-control	Can view general routing, routing protocol, and routing policy configuration information and configure general routing (at the [edit routing-options] hierarchy level), routing protocols (at the [edit protocols] hierarchy level), and routing policy (at the [edit policy-options] hierarchy level).
secret	Can view passwords and other authentication keys in the configuration.
secret-control	Can view passwords and other authentication keys in the configuration and can modify them in configuration mode.
shell	Can start a local shell on the router by entering the start shell command.
snmp	Can view SNMP configuration information in configuration and operational modes.
snmp-control	Can view SNMP configuration information and configure SNMP (at the [edit snmp] hierarchy level).
system	Can view system-level information in configuration and operational modes.
system-control	Can view system-level configuration information and configure it (at the [edit system] hierarchy level).
trace	Can view trace file settings in configuration and operational modes.
trace-control	Can view trace file settings and configure trace file properties.
view	Can use various commands to display current systemwide, routing table, and protocol-specific values and statistics.

Table 8: Default System Login Classes

Login Class	Permission Bits Set
operator	clear, network, reset, trace, view
read-only	view
super-user	all
unauthorized	None

Example: Configure Access Privilege Levels

Create two access privilege classes on the router, one for configuring and viewing user accounts only and the second for configuring and viewing SNMP parameters only:

```
[edit]
system {
  login {
    class user-accounts {
      permissions [ configure admin admin-control ];
    }
    class network-mgmt {
      permissions [ configure snmp snmp-control ];
    }
  }
}
```

Deny or Allow Individual Commands

By default, each top-level CLI command and each configuration statement has an access privilege level associated with it. Users can execute only those commands and configure and view only those statements for which they have access privileges. For each login class, you can explicitly deny or allow commands that would otherwise be permitted or disallowed by a privilege level specified in the permissions statement.

To explicitly deny a command that would otherwise be permitted, include the deny-commands statement at the [edit system login class *class-name*] hierarchy level:

```
[edit system login class class-name]
deny-commands regular-expression;
```

To explicitly allow additional commands that would otherwise be denied, include the allow-commands statement at the [edit system login class *class-name*] hierarchy level:

```
[edit system login class class-name]
allow-commands regular-expression;
```

You can include one deny-commands and one allow-commands statement in each login class.

If the *regular-expression* contains any spaces, operators, or wildcard characters, enclose it in quotation marks. Regular expressions are not case-sensitive.

You use extended regular expressions to specify which commands are denied or allowed. You specify these regular expressions in the allow-commands and deny-commands statements at the [edit system login class] hierarchy level or by specifying JUNOS-specific attributes in your RADIUS authentication server's configuration. If regular expressions are received during RADIUS authentication, they override any regular expressions configured on the local router.

Command regular expressions implement the extended (modern) regular expressions as defined in POSIX 1003.2. Table 9 lists common regular expression operators.

Table 9: Common Regular Expression Operators

Operator	Match...
	One of the two terms on either side of the pipe.
^	At the beginning of an expression, used to denote where the command begins, where there might be some ambiguity.
\$	Character at the end of a command. Used to denote a command that must be matched exactly up to that point. For example, allow-commands "show interfaces \$" means that the user cannot issue show interfaces detail or show interfaces extensive.
[]	Range of letters or digits. To separate the start and end of a range, use a hyphen (-).
()	A group of commands, indicating an expression to be evaluated and the result is then evaluated as part of the overall expression.

If a regular expression contains a syntax error, user authentication fails, and the user cannot login. If a regular expression does not contain any operators, all varieties of the command are allowed. For example, if the following statement is included in the configuration, then the user can issue the commands show interfaces detail and show interfaces extensive in addition to showing an individual interface:

```
allow-commands "show interfaces"
```

Example: Deny or Allow Individual Commands

Configure permissions for individual commands:

```
[edit]
system {
  login {
    /*
    * This login class has operator privileges and the additional ability to reboot the router.
    */
    class operator-and-boot {
      permissions [ clear network reset trace view ];
      allow-commands "request system reboot";
    }
    /*
    * This login class has operator privileges but can't use any commands beginning with "set".
    */
    class operator-no-set {
      permissions [ clear network reset trace view ];
      deny-commands "^set";
    }
    /*
    * This login class has operator privileges and can install software but not view bgp
    information.
    */
    class operator-and-install-but-no-bgp {
      permissions [ clear network reset trace view ];
      allow-commands "request system software add";
      deny-commands "show bgp";
    }
  }
}
```

Configure the Timeout Value for Idle Login Sessions

An idle login session is one in which the CLI operational mode prompt is displayed but there is no input from the keyboard. By default, a login session remains established until a user logs out of the router, even if that session is idle. To close idle sessions automatically, you configure a time limit for each login class. If a session established by a user in that class remains idle for the configured time limit, the session automatically closes.

To define the timeout value for idle login sessions, include the `idle-timeout` statement at the [edit system login class] hierarchy level:

```
[edit system login class class-name]
idle-timeout minutes;
```

Specify the number of minutes that a session can be idle before it is automatically closed.

If you have configured a timeout value, the CLI displays messages similar to the following when timing out an idle user. It starts displaying these messages 5 minutes before timing out the user.

```
user@host# Session will be closed in 5 minutes if there is no activity.
Warning: session will be closed in 1 minute if there is no activity
Warning: session will be closed in 10 seconds if there is no activity
Idle timeout exceeded: closing session
```

If you configure a timeout value, the session closes after the specified time has elapsed except if the user is running Telnet or monitoring interfaces using the `monitor interface` or `monitor traffic` command.

Configure User Accounts

One way for users to access the router CLI is if they have user accounts. (Users can access the router without accounts if you have configured RADIUS or TACACS+ servers as described in “User Authentication” on page 186.) For each account, you define the login name for the user and, optionally, information that identifies the user. After you have created an account, the software creates a home directory for the user.

To create user accounts, include the `user` statement at the [edit system login] hierarchy level:

```
[edit system]
login {
  user user-name {
    full-name complete-name;
    uid uid-value;
    class class-name;
    authentication {
      (encrypted-password "password" | plain-text-password);
      ssh-rsa "public-key";
    }
  }
}
```

For each user account, you can define the following:

User name—(Optional) Name that identifies the user. It must be unique within the router. Do not include spaces, colons, or commas in the user name.

User's full name—(Optional) If the full name contains spaces, enclose it in quotation marks. Do not include colons or commas.

User identifier (UID)—(Optional) Numeric identifier that is associated with the user account name. The identifier must be in the range 100 through 64000 and must be unique within the router. If you do not assign a UID to a user name, the software assigns one when you commit the configuration, preferring the lowest available number.

You must ensure that the UID is unique. However, it is possible to assign the same UID to different users. If you do this, the CLI displays a warning when you commit the configuration, then assigns the duplicate UID.

User's access privilege—(Required) One of the login classes you defined in the class statement at the [edit system login] hierarchy level or one of the default classes listed in Table 8 on page 207.

Authentication method or methods and passwords that the user can use to access the router—(Optional) You can use SSH or an MD5 password, or you can enter a plain-text password that the JUNOS software encrypts using MD5-style encryption before entering it in the password database. For each method, you can specify the user's password. If you configure the plain-text-password option, you are prompted to enter and confirm the password:

```
[edit system]
user@host# set root-authentication plain-text-password
New password: type password here
Retype new password: retype password here
```

For SSH authentication, you can copy the contents of an SSH keys file into the configuration. For information about how to specify filenames, see “How to Specify Filenames” on page 184. To load an SSH key file, use the load-key-file command. If you load the SSH keys file, the contents of the file are copied into the configuration immediately after you enter the load-key-file statement. To view the SSH keys entries, use the configuration mode show command. For example:

```
[edit system]
user@host# set root-authentication load-key-file my-host:.ssh/identity.pub
.file.19692 | 0 KB | 0.3 kB/s | ETA: 00:00:00 | 100%
[edit system]
user@host# show
root-authentication {
ssh-rsa "1024 35 97276382040842510554682267572498642416303222074049625
2839038203869014158453496417001961060835872296156347578491827360336
1276441874265946893207739108344810126831259577226254616679992783161
2350043866091586628382248974673260566119218148953981396556156378621
194032768780653816960202749164163735913269396344008443
boojum@juniper.net"; # SECRET-DATA
}
```

An account for the user root is always present in the configuration. You configure the password for root using the root-authentication statement as described in “Configure the Root Password” on page 196.

Example: Configure User Accounts

Create accounts for four router users, and create an account for the template user “remote.” All users use one of the default system login classes.

```
[edit]
system {
  login {
    user philip {
      full-name "Philip of Macedonia";
      uid 1001;
      class superuser;
      authentication {
        encrypted-password "$1$poPpeY";
      }
    }
    user alexander {
      full-name "Alexander the Great";
      uid 1002;
      class view;
      authentication {
        encrypted-password "$1$14c5.$sBopasdFFdssdffFFdsdfs0";
        ssh-rsa "8924 37 5678 5678@gaugamela.per";
      }
    }
    user darius {
      full-name "Darius King of Persia";
      uid 1003;
      class operator;
      authentication {
        ssh-rsa "1024 37 12341234@ecbatana.per";
      }
    }
    user anonymous {
      class unauthorized;
    }
    user remote {
      full-name "All remote users";
      uid 9999;
      class read-only;
    }
  }
}
```