

# Chapter 22

## Configure User Access

To configure user access, you do the following:

Define Login Classes on page 291

Configure User Accounts on page 302

For information about how to configure user access by means of ssh, see “Configure ssh Service” on page 328.

### 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 302.

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";
    allow-configuration "regular-expression";
    deny-commands "regular-expression";
    deny-configuration "regular-expression";
    idle-timeout minutes;
    permissions [ permissions ];
  }
}
```

Use *class-name* to name the login class. The software contains a few predefined login classes, which are listed in Table 12, “Default System Login Classes” on page 294. The predefined login classes cannot be modified.

**Note**

You cannot modify a predefined login class name. If you issue the set command on a predefined class name, the JUNOS software will append `-local` to the login class name. The following message also appears:

```
warning: '<classname>' is a predefined class name;
changing to '<classname>-local'
```

**Note**

You cannot issue the rename or copy command on a predefined login class. Doing so results in the following error message:

```
error: target '<classname>' is a predefined class
```

For each login class, you can do the following:

Configure Access Privilege Levels on page 292

Deny or Allow Individual Commands on page 295

Configure the Timeout Value for Idle Login Sessions on page 301

## 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 11. 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. Two forms for the permissions control the individual parts of the configuration:

“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 11: 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).
access	Can view the access configuration in configuration mode and with the show configuration operational mode command.
access-control	Can view and configure access information (at the [edit access] 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.
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 chassis, class of service, groups, forwarding options, and interfaces configuration information. Can configure chassis, class of service, groups, forwarding options, and interfaces (at the [edit] hierarchy).
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.

Permission Bit	Description
secret-control	Can view passwords and other authentication keys in the configuration and can modify them in configuration mode.
security	Can view security configuration in configuration mode and with the show configuration operational mode command.
security-control	Can view and configure security information (at the [edit security] hierarchy level).
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 12: 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, all top-level CLI commands have associated access privilege levels. Users can execute only those commands and view only those statements for which they have access privileges. For each login class, you can explicitly deny or allow the use of operational and configuration mode commands that would otherwise be permitted or not allowed by a privilege level specified in the permissions statement. For information about CLI commands, see “Command-Line Interface Overview” on page 127.



### Note

The all login class permission bits take precedence over extended regular expressions when a user issues the rollback command.

Users cannot issue the load override command when specifying an extended regular expression. Users can only issue the merge, replace, and patch configuration commands.

This section describes how to define a user’s access privileges to individual operational and configuration mode commands. It contains the following topics:

Operational Mode Commands on page 295

Configuration Mode Commands on page 298

## Operational Mode Commands

You can specify extended regular expressions with the `allow-commands` and `deny-commands` attributes to define a user’s access privileges to individual operational commands. Doing so takes precedence over login class permission bits set for a user. You can include one `deny-commands` and one `allow-commands` statement in each login class.

To explicitly allow an individual operational mode command 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";
```

To explicitly deny an individual operational mode command that would otherwise be allowed, 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";
```

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

Use extended regular expressions to specify which operational mode 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 TACACS+ or RADIUS authentication server's configuration. You must specify that these regular expressions are sent as the value of Juniper vendor-specific attributes. If regular expressions are received during TACACS+ or RADIUS authentication, they override any regular expressions configured on the local router. For information about TACACS+ or RADIUS authentication, see "Configure User Access" on page 291.

Command regular expressions implement the extended (modern) regular expressions as defined in POSIX 1003.2. Table 13 lists common regular expression operators.

**Table 13: Operational Mode Commands—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, <code>allow-commands "show interfaces \$"</code> means that the user cannot issue <code>show interfaces detail</code> or <code>show interfaces extensive</code> .
[ ]	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; the result is then evaluated as part of the overall expression.

If a regular expression contains a syntax error, authentication fails and the user cannot log in. 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, 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 1: Define Access Privileges to Individual Operational Mode Commands

The following examples define user access privileges to individual operational mode commands.

If the following statement is included in the configuration and the user does not have the `configure login class permission bit`, the user can enter configuration mode

```
[edit system login class class-name]
user@host# set allow-commands configure
```

If the following statement is included in the configuration and the user does not have the `configure login class permission bit`, the user can enter configuration exclusive mode.

```
[edit system login class class-name]
user@host# set allow-commands "configure exclusive"
```

**Note**

You cannot use runtime variables. In the following example the runtime variable 1.2.3.4 cannot be used.

```
[edit system login class class-name
user@host set deny "show bgp neighbor 1.2.3.4"
```

### Example 2: Define Access Privileges to Individual Operational Mode Commands

Configure permissions for individual operational mode 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";
    }
  }
}
```

## Configuration Mode Commands

You can specify extended regular expressions with the `allow-configuration` and `deny-configuration` attributes to define user access privileges to parts of the configuration hierarchy or individual configuration mode commands. Doing so overrides login class permission bits set for a user. You can also use wildcards to restrict access. When you define access privileges to parts of the configuration hierarchy or individual configuration mode commands, do the following:

Specify the full paths in the extended regular expressions with the `allow-configuration` and `deny-configuration` attributes.

Enclose parentheses around an extended regular expression that connects two or more terms with the pipe (`|`) symbol. For example:

```
[edit system login class class-name]
user@host# set deny-configuration "(system login class)|(system services)"
```



**Note**

Do not use spaces between regular expressions separated with parentheses and connected with the pipe (`|`) symbol.

You cannot define access to keywords such as `set`, `edit`, or `activate`.

For more information about how to use wildcards, see Table 14, “Configuration Mode Commands—Common Regular Expression Operators” on page 299.

To explicitly allow an individual configuration mode command that would otherwise be denied, include the `allow-configuration` statement at the `[edit system login class class-name]` hierarchy level:

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

To explicitly deny an individual configuration mode command that would otherwise be allowed, include the `deny-configuration` statement at the `[edit system login class class-name]` hierarchy level:

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

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

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

Use extended regular expressions to specify which configuration mode commands are denied or allowed. You specify these regular expressions in the allow-configuration and deny-configuration statements at the [edit system login class] hierarchy level, or by specifying JUNOS-specific attributes in your TACACS+ or RADIUS authentication server's configuration. You must specify that these regular expressions are sent as the value of Juniper vendor-specific attributes. If regular expressions are received during TACACS+ or RADIUS authentication, they override any regular expressions configured on the local router. For information about TACACS+ or RADIUS authentication, see "Configure User Access" on page 291.

Command regular expressions implement the extended (modern) regular expressions, as defined in POSIX 1003.2. Table 14 lists common regular expression operators.

**Table 14: Configuration Mode Commands—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; the result is then evaluated as part of the overall expression.
*	0 or more terms.
+	One or more terms.
.	Any character except for a space " ".

### Example 3: Define Access Privileges to Individual Configuration Mode Commands

The following examples show how to configure access privileges to individual configuration mode commands.

If the following statement is included in the configuration and the user's login class permission bit is set to all, the user cannot configure telnet parameters:

```
[edit system login class class-name]
user@host# set deny-configuration "system services telnet"
```

If the following statement is included in the configuration and the user's login class permission bit is set to all, the user cannot issue login class commands within any login class whose name begins with "m".

```
[edit system login class class-name]
user@host# set deny-configuration "system login class m.*"
```

If the following statement is included in the configuration and the user's login class permission bit is set to all, the user cannot issue configuration mode commands at the login class or system services hierarchy levels.

```
[edit system login class class-name]
user@host# set deny-configuration "(system login class) | (system services)"
```

If the following statement is included in the configuration and the user's login class permission bit is set to protocols, the user cannot issue login class commands within any login class whose name begins with "m".

```
[edit system login class class-name]
user@host# set deny-configuration "system login class m.*"
```

#### Example 4: Configure Access Privileges to Individual Configuration Mode Commands

Configure permissions for individual configuration mode commands:

```
[edit]
system {
  login {
    /*
     * This login class has operator privileges and the additional ability to issue commands
     * at the system services hierarchy.
     */
    class only-system-services {
      permissions [ configure ];
      allow-configuration "system services";
    }
    /*
     * This login class has operator privileges but can't issue any system services
     * commands.
     */
    class all-except-system-services {
      permissions [ all ];
      deny-configuration "system services";
    }
  }
}
```

## **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

User accounts provide one way for users to access the router. (Users can access the router without accounts if you configured RADIUS or TACACS+ servers, as described in “User Authentication” on page 263.) 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";
      ssh-dsa "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 12, “Default System Login Classes” on page 294.

**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 information about how to create plain-text passwords, see “Plain-Text Passwords” on page 21.

For ssh authentication, you can copy the contents of an ssh keys file into the configuration. For information about how to specify filenames, see “Specify Filenames and URLs” on page 260.

To load an ssh key file, use the `load-key-file` command. This command loads RSA (ssh version 1) and DSA (ssh version 2) public keys. You can also configure a user to use `ssh-rsa` and `ssh-dsa` keys.

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 9727638204084251055468226757249864241630322207404
962528390382038690141584534964170019610608358722961563475784918
273603361276441874265946893207739108344810126831259577226254616
679992783161235004386609158662838224897467326056611921814895398
139655615637862119403276878065381696020274916416373591326939634
4008443 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 275.

## 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-dsa "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;  
}  
}  
}
```