Communicating with the Remediation Subsystem

Your remediation module must receive information from the Firepower Management Center remediation subsystem to successfully perform its function. You configure the information that your module receives in an XML file called `module.template`. Without it, the remediation subsystem cannot interact with your remediation module.

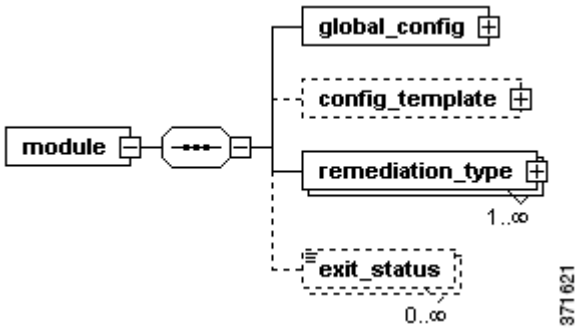
The `module.template` XML file allows you to specify:

- a set of module-level declarations such as the name and version of your remediation module, a short descriptive text, and the name of the binary file for your remediation program
- the information the module requires from the user when the user configures remediation instances in the Firepower Management Center user interface
- the specific remediation actions, known as remediation types, that the module can perform and the correlation event data each remediation type requires
- any custom return codes and exit status messages that your remediation program returns to the Firepower Management Center

Before writing a `module.template` for your remediation module, you should understand the `module.template` schema (`module.template.xsd`). The schema defines the elements (or tags used to contain data) and attributes (or data used to modify the data contained in an element) you can use to provide information to the remediation subsystem. The `module.template` schema is located on the DC at `/etc/sf/remediation/module.template.vsd`.

The top-level element in `module.template` is `module`, in which you specify the name of the remediation module using the `name` attribute. The `name` attribute is required and accepts a string value between 1 and 64 alphabetic characters.

Caution: You cannot use white space in the module's `name` attribute value. In addition, you cannot use punctuation marks except for underscore (_) or dash (-).



Some XML editors can read the `module.template` schema and automatically generate a `module.template` file with a namespace and schema declaration, with the top level element and child elements and attributes. If you choose not to use such an editor, you must include the child elements manually.

Caution: If you set your XML editor to auto-generate the namespace and schema location, you must delete those lines before including the final version of `module.template` in your installation package.

The following example illustrates the `module` element with only the `name` attribute defined.

```
<module name="example_module">
<global_config>
    <display_name/>
    <version/>
    <binary/>
</global_config>
<remediation_type name="">
    <display_name/>
</remediation_type>
</module>
```

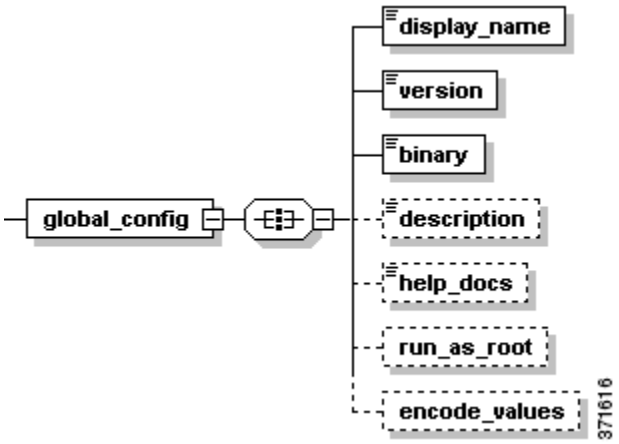
See the following sections for details about writing the rest of `module.template`:

- Defining the Global Configuration, page 3-2 explains how to use the `global_config` element to define the name that appears for your module on the Modules page, as well as the module's version, binary location, and its description.
- Defining the Configuration Template, page 3-4 explains how to use the `config_template` element to define the configuration information that your module requires the user to specify from the web interface.
- Defining the Global Configuration, page 3-2 explains how to use the `remediation_type` element to define the remediations the module can launch and the correlation event data that each remediation requires.
- Defining Exit Statuses, page 3-22 explains how to use the `exit_status` element define the custom exit statuses your module returns to the remediation subsystem.

Defining the Global Configuration

The first required section of `module.template` uses the `global_config` element to define global configuration information. These attributes include the module's name and description, which appear in the list of remediation modules displayed on the Modules page of the Firepower Management Center user interface. The global information also includes the module's version and the location of the executable program that runs when a remediation is triggered.

The following portion of the `module.template` schema diagram illustrates the child elements of the `global_config` element.



The following table describes the child elements available to the `global_config` element.

Table 3-1 global_config Child Elements

Name	Description	Required?
`display_name`	Specifies the name that appears for this remediation module on the Modules page. The display name can contain only alphanumeric characters and white spaces and must be between 1 and 127 characters long. It must be unique across remediation modules.	yes
`version`	Specifies the version of the remediation module. This value appears on the Modules page. The value for the version element must begin and end with numeric characters, but may contain period (.) characters. Note: The combination of the `name` attribute of the `module` element and the data in the `version` element must be unique across remediation modules.	yes
`binary`	Specifies the UNIX filename of the binary that makes up your remediation module.	yes
`description`	Provides a description of the remediation module and its available remediations. The `description` element appears on the Modules page. Descriptions with more than 255 characters are truncated.	yes
`run_as_root`	Sets a flag that allows the remediation module to run as root on the Cisco appliance where it is installed. Caution: Cisco recommends that you use this element only if absolutely necessary.	no
`encode_values`	Sets a flag that HTML-encodes user input. This allows users to enter input that might otherwise be unintentionally interpreted by the XML processor. Note: If you use this element, your remediation module **must** handle HTML decoding as part of its input handling.	no

Consider the following XML code, which illustrates the global configuration portion of a `module.template` file.

```
<global_config>
<display_name>My Firewall</display_name>
<binary>firewall_block.pl</binary>
<description>Dynamically apply firewall rules to my firewall.</description>
<version>1.0</version>
<run_as_root/>
</global_config>
```

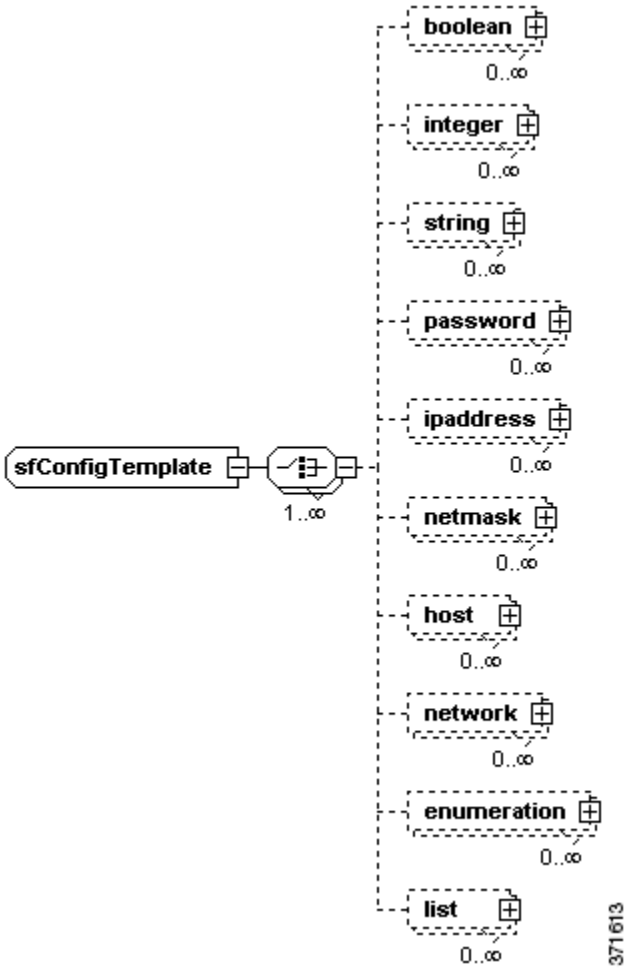
In this example, the remediation module is represented by the name My Firewall in the web interface. It runs version 1.0 of a program called `firewall_block.pl`, which you install using the Firepower Management Center (see Packaging and Installing Your Module, page 2-12 for more information). The program dynamically applies firewall rules to a specific firewall and runs as root on the Firepower Management Center.

Defining the Configuration Template

The `config_template` child element of the `module` element specifies the types of information the user must provide when configuring the instances that this remediation module executes (see Instance Configuration Data, page 2-8). The user provides the information specified in this element via the Firepower Management Center user interface. Each `module` element may contain only one direct child `config_template` element and this element applies to all instances that are configured.

Note, however, that each `remediation_type` element in `module.template` can also contain a child `config_template` element. The `config_template` child element under `remediation_type` allows you to define information that the user must provide for each of the different remediation types. So a user will have to configure general instance-level fields using the `config_template` element in the `module` portion, and then, optionally, an additional set of `config_template` fields specific to the remediation type being executed by the instance. For more information, see Defining Remediation Types, page 3-20.

The following diagram illustrates the child elements available to the `config_template` element.



The `config_template` element allows you to render several basic field types in the web interface. You choose which `config_template` child elements to use depending on the data you need to collect from the user for the remediation module. All child elements of `config_template` are optional and can be used as many times as needed within a `config_template` element. Fields are rendered on the web interface in the order in which they are included in the `config_template` element.

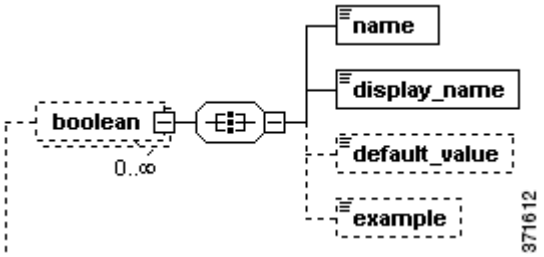
See the following sections for more information on the child elements that represent the fields you can use to collect configuration information on the instance configuration and remediation configuration pages in the web interface:

- The boolean Element, page 3-6
- The integer Element, page 3-7
- The string Element, page 3-8
- The password Element, page 3-9
- The ipaddress Element, page 3-11
- The netmask Element, page 3-12
- The host Element, page 3-13
- The network Element, page 3-14

The boolean Element

Each `boolean` element you use in a `config_template` represents a true/false choice, which appears as a set of radio buttons labeled **On** or **Off**, users can make in the web interface. If you set the element's `required` attribute to false, an additional radio button is available, labeled **Not Selected**.

The following portion of the `module.template` schema diagram illustrates the `boolean` element's child elements.



When configuring child elements for an occurrence of a `boolean` element, you may only use each available child element once. The following table describes the child elements available to the `boolean` element.

Table 2 boolean Attributes and Child Elements

Name	Type	Description	Required?
required	attribute	Indicates whether specifying a value in the field is optional. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must select either **On** or **Off**. If you set the value of the attribute to `false`, the web interface indicates that the choice is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. Names should be unique within a module.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field. If the web interface user does not specify a value, the remediation program uses this value by default.	no

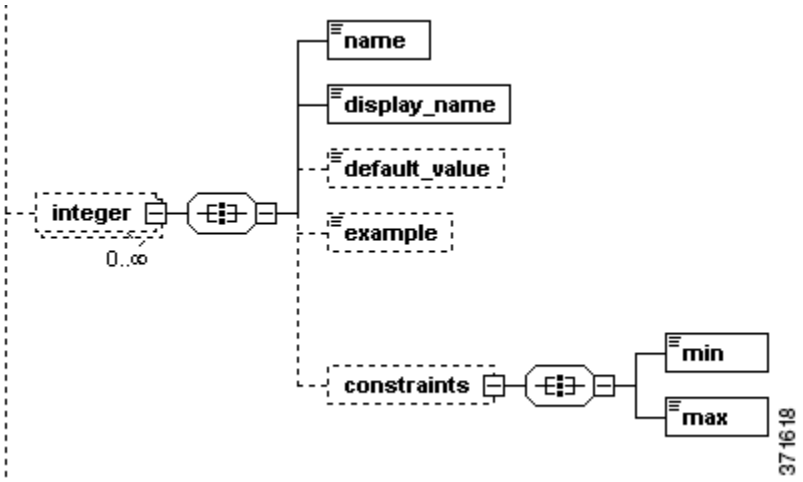
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled "Enabled?" that provides user with two choices: **On** or **Off**. The choice defaults to `true`, that is, the radio button labeled **On** is preselected.

```
<boolean>
<name>process_enabled</name>
<display_name>Enabled?</display_name>
<default_value>true</default_value>
</boolean>
```

The integer Element

Each `integer` element you use in a `config_template` represents a field in the web interface that accepts an integer value.

The following diagram illustrates the child and grandchild elements of the `integer` element.



The following table describes the child elements available to the `integer` element.

Table 3 integer Attributes, Child Elements, and Grandchild Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character.Names should be unique within a module.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field. If the web interface user does not specify a value, the remediation program uses this value by default.	no
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no
`constraints`	element	Constrains the values that the user can enter in this field to fall between specified minimum and maximum values, inclusive. The `constraints` element has two child elements: `min` and `max`. Each is an optional, single-occurrence child element that accepts an integer value.	no

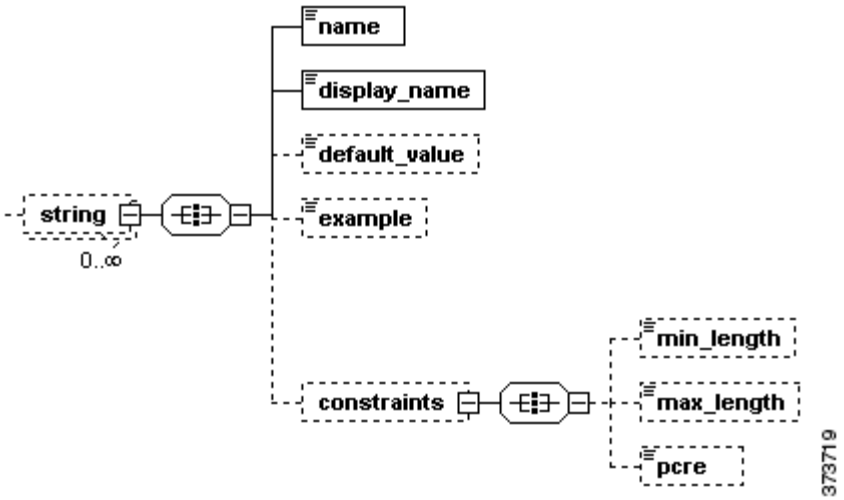
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled “Rate”, which accepts an integer value between 0 and 500 but defaults to 430.

```
<integer>
<name>rate</name>
<display_name>Rate</display_name>
<default_value>430</default_value>
<constraints>
    <min>0</min>
    <max>500</max>
</constraints>
</integer>
```

The string Element

Each `string` element you use in a `config_template` represents a field in the web interface that accepts a string value.

The following diagram illustrates the child elements of the `string` element instance.



The following table describes child elements available to the `string` element.

Table 4 string Attributes, Child Elements, and Grandchild Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character.Names should be unique within a module.	yes
`display_name`	element	Specifies the web interface label for this field.	yes

Table 4 string Attributes, Child Elements, and Grandchild Elements (continued)

Name	Type	Description	Required?
`default_value`	element	Specifies the default value for this field.If the web interface user does not specify a value, the remediation program uses this value by default.	no
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no
`constraints`	element	Constrains the values that the user can enter in this field. The `constraints` element has three child elements: `min_length`, `max_length` and `pcre`. The `min_length` and `max_length` elements are optional, single-occurrence child elements that accept integer values and specify a range for the acceptable length of string values. The `pcre` element is optional; use it to specify a Perl-compatible regular expression that provides additional constraints.	no

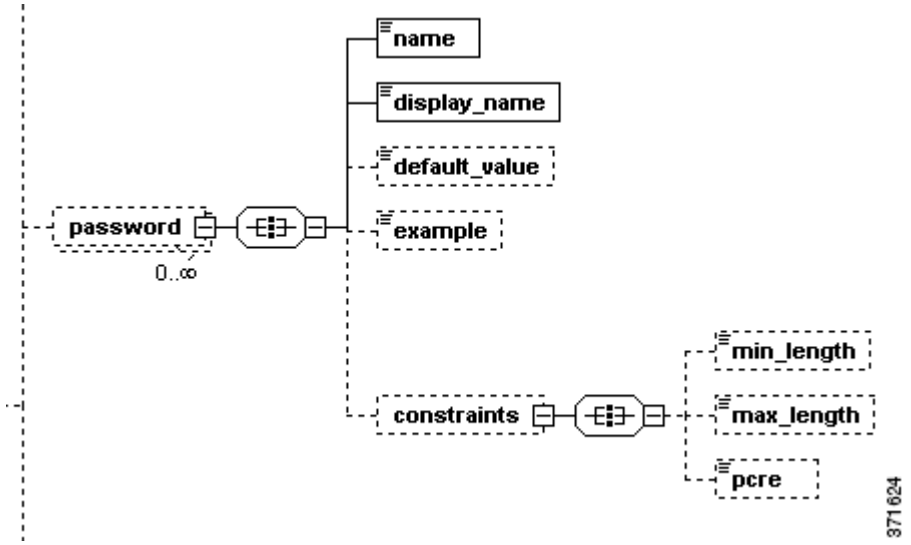
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled “Username”, which accepts a string value that is at least eight characters long and does not use white spaces.

```
<string>
    <name>user_name</name>
    <display_name>Username</display_name>
    <constraints>
        <min_length>8</min_length>
        <pcre>\S+</pcre>
    </constraints
</string>
```

The password Element

Each `password` element you use in a `config_template` represents a field in the web interface that accepts a string comprised of alphanumeric characters.

The following diagram illustrates the child and grandchild elements of the `password` element instance.



The following table describes the child elements available to the `password` element.

Table 5 password Attributes, Child Elements, and Grandchild Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. Names should be unique within modules.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field. If the web interface user does not specify a value, the remediation program uses this value by default.	no
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no
`constraints`	element	Constrains the values that the user can enter in this field. The `constraints` element has three child elements: `min_length`, `max_length` and `pcre`. The `min_length` and `max_length` elements are optional, single-occurrence child elements that accept integer values and specify a range for the acceptable length of password values. The `pcre` element is optional; use it to specify a Perl-compatible regular expression that provides additional constraints.	no

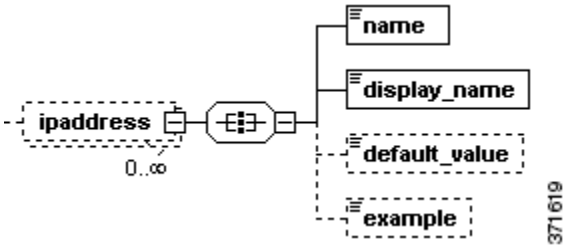
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled “Login Password”, which accepts an alphanumeric string between 6 and 12 characters long.

```
<password>
    <name>login_password</name>
    <display_name>Login Password</display_name>
    <constraints>
        <min_length>6</min_length>
        <max_length>12</max_length>
    </constraints>
</password>
```

The ipaddress Element

Each `ipaddress` element you use in a `config_template` represents a field in the web interface that accepts a single IP address. IP addresses may be entered in the form of a fully formed dotted quad (for example, 1.1.1.1).

The following diagram illustrates the child elements of the `ipaddress` element.



When configuring child elements for an occurrence of an `ipaddress` element, you may only use each available child element once. The following table describes the child elements available to the `ipaddress` element.

Table 6 ipaddress Attributes and Child Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. Names should be unique within modules.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field.	no
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no

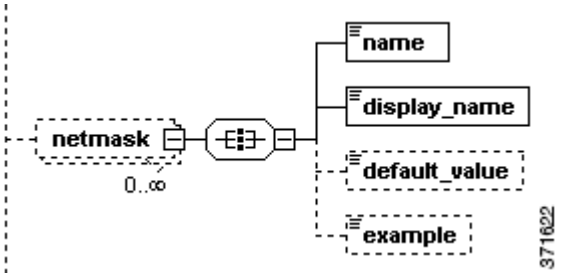
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled “Mail Server,” which accepts a single IP address.

```
<ipaddress>
<name>mail_server</name>
<display_name>Mail Server</display_name>
</ipaddress>
```

The netmask Element

Each `netmask` element you use in a `config_template` represents a field in the web interface that accepts netmask values. Netmask values can be denoted by a dotted quad (255.255.255.255) or a CIDR mask (/8).

The diagram illustrates the child elements of the `netmask` element.



When configuring child elements for an occurrence of a `netmask` element, you may only use each available child element once. The following table describes the child elements available to the `netmask` element.

Table 7 netmask Attributes and Child Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. Names should be unique within modules.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field. If the web interface user does not specify a value, the remediation program uses this value by default.	no
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no

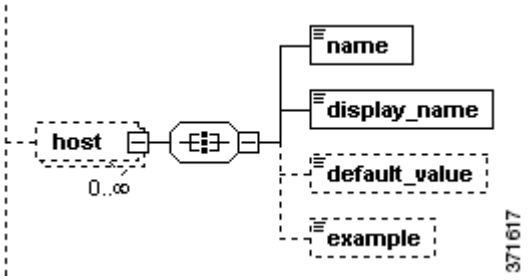
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled “Netmask”, which accepts netmask values denoted by a dotted quad or CIDR mask and defaults to 255.255.255.255.

```
<netmask>
<name>netmask</name>
<display_name>Netmask</display_name>
<default_value>255.255.255.0</default_value>
</netmask>
```

The host Element

Each `host` element you use in a `config_template` represents a field in the web interface that accepts a single IP address or string.

The following diagram illustrates the child elements of the `host` element.



When configuring child elements for an occurrence of a `host` element, you may only use each available child element once. The following table describes the child elements and attributes available to the `host` element.

Table 8 host Attributes and Child Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. Names should be unique within modules.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field.If the web interface user does not specify a value, the remediation program uses this value by default.	no
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no

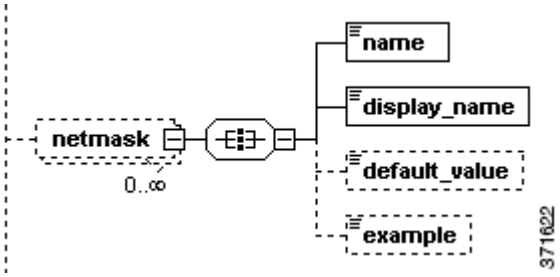
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled “Host Name”, which accepts an IP address or string. The web interface also provides example text of “192.10.1.3.”

```
<host>
<name>hostname</name>
<display_name>Host Name</display_name>
<example>192.10.1.3</example>
</host>
```

The network Element

Each `network` element you use within a `config_template` represents a field in the web interface. A network field accepts an IP address (assumed to be a single IP address, that is, an IP address with /32 netmask) or a CIDR block.

The following diagram illustrates the child elements of the `network` element.



When configuring child elements for an occurrence of a `network` element, you may only use each available child element once. The following table describes the child elements and attributes available to the `network` element.

Table 9 network Attributes and Child Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. Names should be unique within modules.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field. If the web interface user does not specify a value, the remediation program uses this value by default.	no
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no

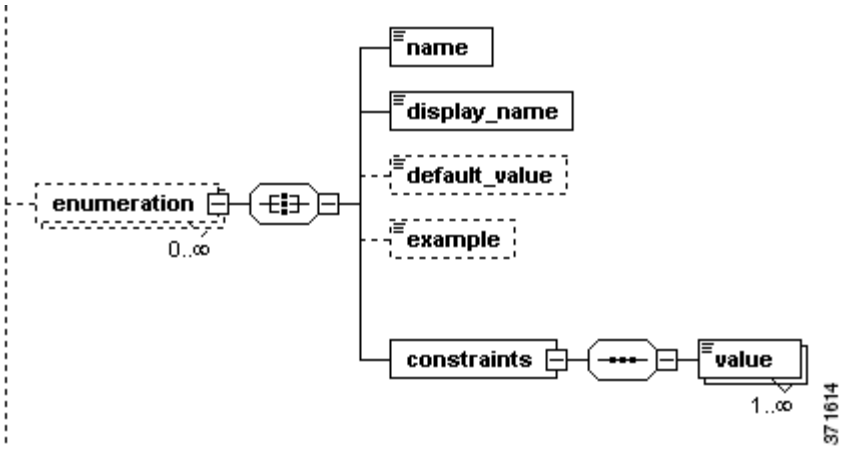
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled “Monitored Network”, which accepts either a /32 IP address or an IP address and netmask value, and which has a default value of `192.168.1.0/24`.

```
<network>
<name>monitored_network</name>
<display_name>Monitored Network</display_name>
<default_value>192.168.1.0/24</default_value>
</network>
```

The enumeration Element

Each `enumeration` element you use in a `config_template` represents a drop-down list of strings displayed in the web interface. Users can select a single value from this list.

The following diagram illustrates the child and grandchild elements of the `enumeration` element.



The following table describes the child elements and attributes available to the `enumeration` element.

Table 10 enumeration Attributes, Child Elements, and Grandchild Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. Names should be unique within modules.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field. If the web interface user does not specify a value, the remediation program uses this value by default.	no

Table 10 enumeration Attributes, Child Elements, and Grandchild Elements (continued)

Name	Type	Description	Required?
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no
`constraints`	element	Specifies the values that the user can enter in this field. The `constraints` element has one required child element, `value`, that accepts a string that represents one choice for the users. Use multiple `value` elements to provide multiple choices to the user.	yes

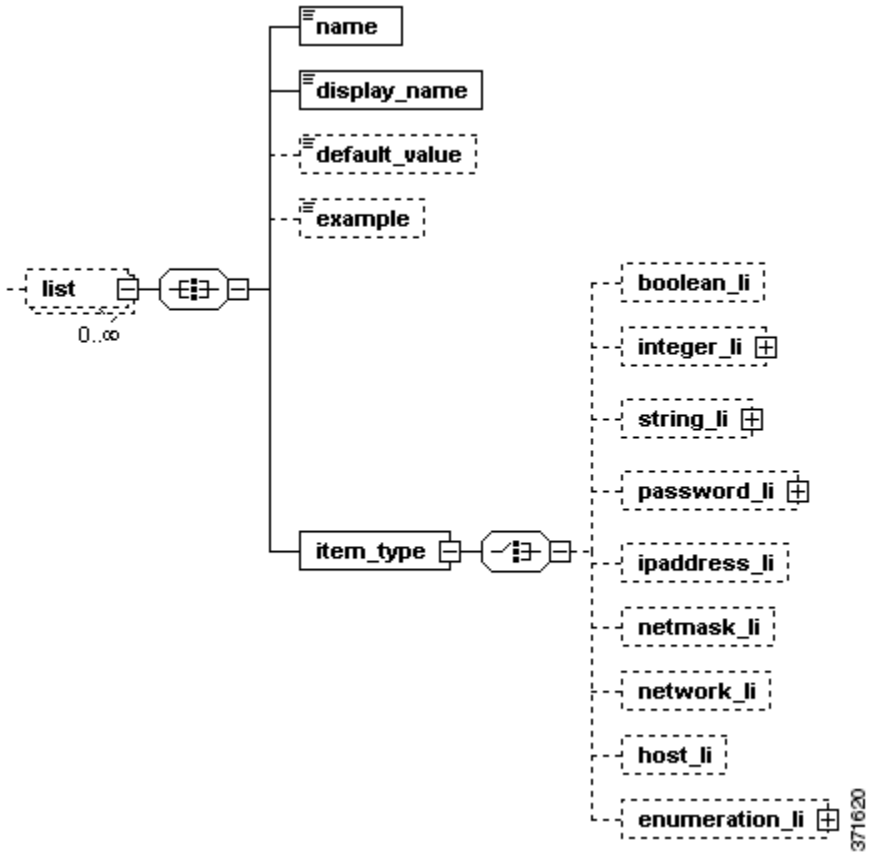
The following portion of a `config_template` element definition indicates that the web interface displays a field labeled “Day”, which allows users to select one of the values provided (`Monday`, `Tuesday`, `Wednesday`, `Thursday`, and `Friday`).

```
<enumeration>
<name>day</name>
<display_name>Day</display_name>
<constraints>
    <value>Monday</value>
    <value>Tuesday</value>
    <value>Wednesday</value>
    <value>Thursday</value>
    <value>Friday/value>
</constraints>
</enumeration>
```

The list Element

Each `list` element you use in a `config_template` represents a field in the web interface that allows users to enter a list of values, one per line, whose type is specified by the required `item_type` child element.

The following diagram illustrates the child and grandchild elements of the `list` element.



The following table describes the child elements available to the `list` element.

Table 11 list Attribute and Child Elements

Name	Type	Description	Required?
required	attribute	Indicates whether users must provide a value in the field. This attribute defaults to `true`. You are not required to use this attribute. Therefore, if you do not use it (or if you explicitly set its value to `true`), users must provide a value. If you set the value of the attribute to `false`, the web interface indicates that providing a value is optional.	no
`name`	element	Provides context to the remediation module for the value entered in the field. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. Names should be unique within modules.	yes
`display_name`	element	Specifies the web interface label for this field.	yes
`default_value`	element	Specifies the default value for this field. If the web interface user does not specify a value, the remediation program uses this value by default.	no

Table 11 list Attribute and Child Elements (continued)

Name	Type	Description	Required?
`example`	element	Provides an example of the input that the remediation module expects to receive. Note: This value is not displayed in the web interface.	no
`item_type`	element	Specifies the type of value that can appear in this field. The value type is specified by a child element. Valid child elements are listed below.	no

The following list describes the child elements available to the `item_type` element, which are similar to the child elements of the `config_template` element; the only difference is that `item_type` child elements do not use the `required` attribute. Each instance of the `item_type` element can use only one child element:

- `boolean_li` indicates that the list accepts multiple Boolean values (see The boolean Element, page 3-6).
- `integer_li` indicates that the list accepts multiple integer values (see The integer Element, page 3-7).
- `string_li` indicates that the list accepts multiple string values (see The string Element, page 3-8).
- `password_li` indicates that the list accepts multiple password values (see The password Element, page 3-9).
- `ipaddress_li` indicates that the list accepts multiple ipaddress values (see The ipaddress Element, page 3-11).
- `network_li` indicates that the list accepts multiple network values (see The network Element, page 3-14).
- `netmask_li` indicates that the list accepts multiple netmask values (see The netmask Element, page 3-12).
- `host_li` indicates that the list accepts multiple host values (see The host Element, page 3-13).
- `enumeration_li` indicates that the list accepts multiple values as defined by the `value` child elements of the `enumeration_li` element's `constraints` child element (see The enumeration Element, page 3-15).

The following portion of a `config_template` element definition indicates that the web interface should allow the user to provide a list of integers between zero and 500 inclusive, one per line, in a field labeled "Integer List".

```
<list>
<name>list_integer</name>
<display_name>Integer List</display_name>
<example>Constrained value [0-500]</example>
<item_type>
    <integer_li>
        <constraints>
            <min>0</min>
            <max>500</max>
        </constraints>
    </integer_li>
</item_type>
```

```
</list>
```

Sample Configuration Template

This section provides a sample `config_template` element definition, which governs both the web interface appearance and the types of information the remediation module must receive from the user.

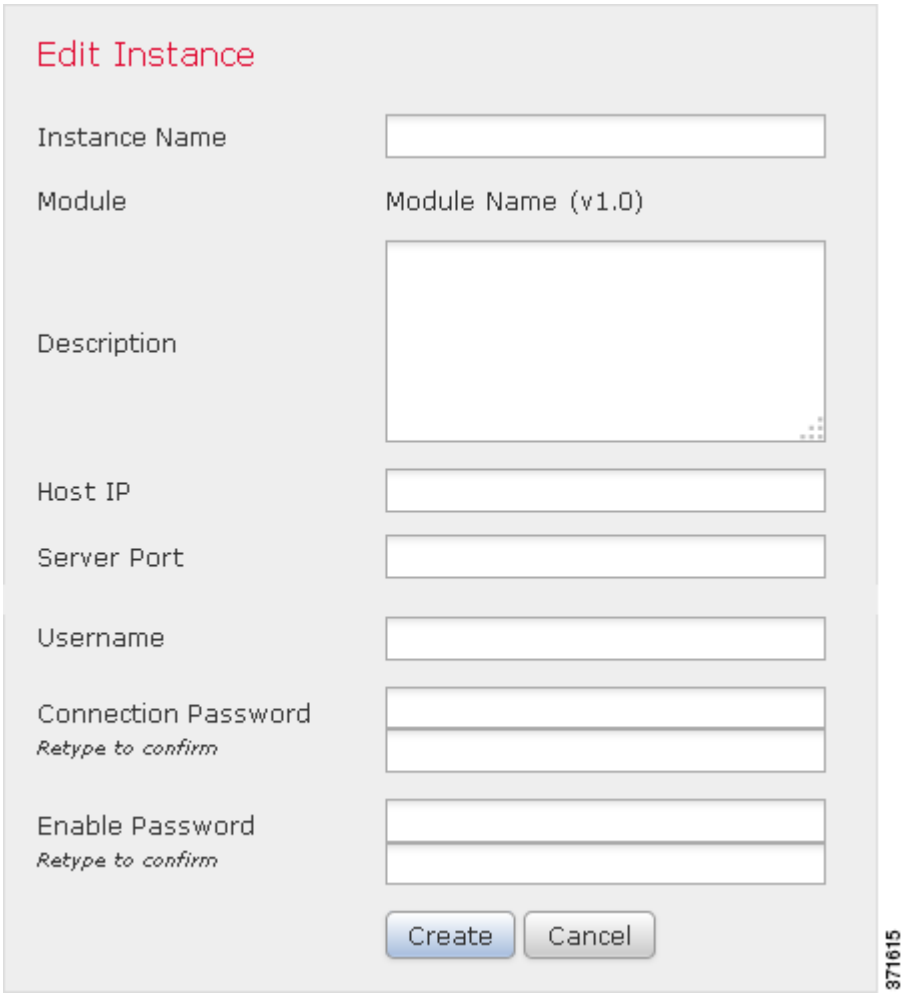
```
<config_template>
    <ipaddress>
        <name>host_ip</name>
        <display_name>Host IP</display_name>
    </ipaddress>
    <string>
        <name>user_name</name>
        <display_name>Username</display_name>
    </string>
    <password>
        <name>login_password</name>
        <display_name>Connection Password</display_name>
    </password>
    <password>
        <name>root_password</name>
        <display_name>Enable Password</display_name>
    </password>
</config_template>
```

The above template renders four fields on the web interface. The following table describes each field.

Table 3-12 Fields Created by the Sample Configuration Template

Field	Description
Host IP	Accepts an IP address that the remediation module identifies as `host_ip`.
Username	Accepts a string that the remediation module identifies as `user_name`.
Connection Password	Accepts an alphanumeric password string that the remediation module identifies as `login_password`.
Enable Password	Accepts an alphanumeric password string that the remediation module identifies as `root_password`.

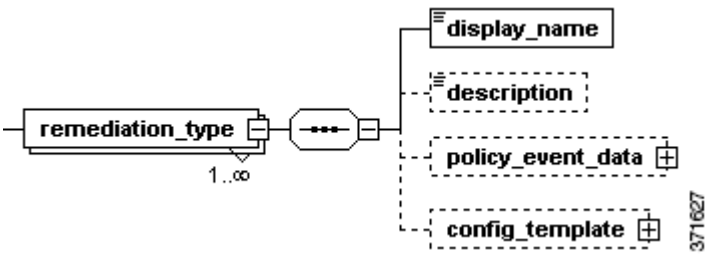
The following screen illustrates how these fields appear on the web interface. You must provide the data requested by these fields to configure the remediation module from the web interface.



Defining Remediation Types

Remediation types describe the actions, or *remediations*, taken by the device that is governed by the remediation module. Each `remediation_type` element you use in `module.template` represents one of those remediations. Remediations are triggered by correlation event data from the remediation subsystem. For more information see Event Data, page 2-2.

The following diagram illustrates the child elements of the `remediation_type` element.



The following table describes the attributes and child elements available to the `remediation_type` element.

Table 13 remediation_type Attributes and Child Elements

Name	Type	Description	Required?
`name`	attribute	Provides context to the remediation module for the remediation type. This attribute is required and accepts a string between 1 and 64 characters, inclusive. Names may not contain white space and may only contain alphanumeric characters and the underscore (_) and dash (-) character. `remediation_type` names must be unique within each module.	yes
`display_name`	element	Labels the remediation type on the web interface.	yes
`policy_event_data`	element	Specifies the correlation event data that the remediation module needs to receive from the remediation subsystem. The `policy_event_data` has one child element, `pe_item`, that represents a specific correlation event data item. Use multiple `pe_item` elements to provide multiple correlation event data items. For more information on appropriate correlation event data values, see Event Data, page 2-2.	no
`config_template`	element	Specifies the information the user must provide when configuring an instance of this remediation module. For more information, see Defining the Configuration Template, page 3-4.	no

The following portion of a `module.template` file illustrates several `remediation_type` element definitions.

```
<remediation_type name="block_src">
<display_name>Block Source</display_name>
<policy_event_data>
    <pe_item>src_ip_addr</pe_item>
    <pe_item>src_port</pe_item>
    <pe_item>src_protocol</pe_item>
</policy_event_data>
</remediation_type>
<remediation_type name="block_dest">
<display_name>Block Destination</display_name>
<policy_event_data>
    <pe_item>dest_ip_addr</pe_item>
    <pe_item>dest_port</pe_item>
    <pe_item>dest_protocol</pe_item>
</policy_event_data>
</remediation_type>
<remediation_type name="acl_insert">
<display_name>ACL Insertion</display_name>
<policy_event_data>
    <pe_item>src_ip_addr</pe_item>
    <pe_item>src_port</pe_item>
    <pe_item>src_protocol</pe_item>
    <pe_item>dest_ip_addr</pe_item>
    <pe_item>dest_port</pe_item>
    <pe_item>dest_protocol</pe_item>
</policy_event_data>
    <config_template>
        <integer>
            <name>acl_num</name>
```

```
            <display_name>ACL Number</display_name>
        </integer>
    </config_template>
</remediation_type>
```

The example above contains 3 remediation types: `block_src`, `block_dest`, and `acl_insert`. Each of these requires specific correlation event (`pe_item`) data. The `acl_insert` remediation type also requires configuration data, which is specified in its `config_template` child element; users must provide an ACL number when they configure instances of that type.

Defining Exit Statuses

The remediation subsystem expects to receive an exit status, or return code, in the form of an integer from your remediation module.

Cisco provides a set of predefined exit status messages your remediation module can return. You can return predefined exit statuses, which correspond to integer values between 1 and 128, inclusive. The following lists and describes these predefined exit status codes.

Table 3-14 Predefined Exit Statuses

Exit Status	Description
0	Successful completion of remediation.
1	Error in the input provided to the remediation module.
2	Error in the remediation module configuration.
3	Error logging into the remote device or server.
4	Unable to gain required privileges on remote device or server.
5	Timeout logging into remote device or server.
6	Timeout executing remote commands or servers.
7	The remote device or server was unreachable.
8	The remediation was attempted but failed.
10	A white-list match was found.
11	Failed to execute remediation program
20	Unknown/unexpected error.

Alternatively, your module may return integers between 129 and 254, inclusive, as custom exit statuses. If your remediation module returns custom exit statuses, you must define the set of exit statuses it can return. Each `exit_status` element you use in `module.template` represents a custom exit status that your remediation module can return. For more information, see Data Returned by Modules, page 2-12.

The `exit_status` element accepts a string that describes a return code. In addition, the element requires an attribute, `value`, that accepts a unique integer between 129 and 255. This attribute associates remediation module return codes with their descriptions, which the user can see in remediation status event views.

The following example illustrates valid custom `exit_status` elements.

```
<exit_status value="138">syslog error</exit_status>
<exit_status value="139">unknown error</exit_status>
```