

## Use Case Scenarios

<https://campus.barracuda.com/doc/76284990/>

These examples draw on the information presented in the [Barracuda Email Security Gateway API Guide](#) for using various APIs to configure the Barracuda Email Security Gateway for common use cases. Some use cases address domain-level settings, and some address global settings.

### Use Case - Adding an Allow List Entry to a User Account

Use the config.add method to add any email senders to the Allow List for a particular user account. This list of senders are not blocked even if the message matches spam rules. Virus scanning is still applied based on the policy set by the administrator. Allow Listing may be performed by full email address ("user@domain.com") or domain only ("domain.com").

**Important:** Per-User Quarantine must be enabled for the domain via the web interface BEFORE you attempt to add per-user Allow List entries. To do so, first, from the **DOMAINS > Domain Manager** page, click **Manage Domain** for the particular domain. For example, if the user account is *cuda\_user@barracuda.com*, click on **Manage Domain** for barracuda.com. At the domain level, navigate to the **BASIC > Quarantine** page and set **Quarantine Type** to **Per-User**. Finally, set **Enable User Features** to Yes.

#### Arguments:

- my \$value1 = 'user1@mymail.net';
- my \$value2 = 'user2@mymail.net';
- my \$user\_account = 'cuda\_user@mymail.net';

#### Sample Request:

```
<?xml version="1.0" encoding="UTF8"?>
<methodCall>
  <methodName>config.add</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>parent_type</name>
            <value>
              <string>
                <![CDATA[user]]>
              </string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```
</value>
</member>
<member>
<name>variable</name>
<value>
<string>
<![CDATA[user_scana_sender_allow]]>
</string>
</value>
</member>
<member>
<name>values</name>
<value>
<array>
<data>
<value>
<string>
<![CDATA[$value1]]>
</string>
</value>
<value>
<string>
<![CDATA[$value2]]>
</string>
</value>
</data>
</array>
</value>
</member>
<member>
<name>parent_path</name>
<value>
<string><![CDATA[$user_account]]></string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

## Use Case - Adding a Block List Entry for a Domain

Use the config.create method to add any IP addresses or networks to the Block List for a particular domain. This example adds an IP address to the Block List for the specified domain and adds values to the per-domain tied variables listed below. The mta\_acl\_ip\_block\_action is set to 'quarantine' for mail from the IP address added to the Block List, and the mta\_acl\_ip\_block\_netmask is set to 255.255.255.0 since we're adding an individual IP address. A comment of 'Blocked IP address' is added as well.

- # Add values to per domain tied variable
- # Domain - xyz.mydomain.net
- # Variable - mta\_acl\_ip\_block\_address (domain scope): 10.5.36.59
- # Tied variables - mta\_acl\_ip\_block\_netmask, mta\_acl\_ip\_block\_action, mta\_acl\_ip\_block\_comment.

**Arguments:**

- type: mta\_acl\_ip\_block\_address
- parent\_path: xyz.mydomain.net
- mta\_acl\_ip\_block\_netmask: 255.255.255.0
- mta\_acl\_ip\_block\_action: 'Quarantine'
- mta\_acl\_ip\_block\_comment: 'Blocked IP Address'

**Sample Request:**

```
<?xml version="1.0" encoding="UTF8"?>
<methodCall>
  <methodName>config.create</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>parent_type</name>
            <value>
              <string><![CDATA[domain]]></string>
            </value>
          </member>
          <member>
            <name>name</name>
            <value>
              <string><![CDATA[10.5.36.59]]></string>
            </value>
          </member>
          <member>
            <name>type</name>
            <value>
              <string>
```

```
<! [CDATA[mta_acl_ip_block_address]]>
</string>
</value>
</member>
<member>
<name>parent_path</name>
<value>
<string><! [CDATA[xyz.mydomain.net]]></string>
</value>
</member>
<member>
<name>mta_acl_ip_block_netmask</name>
<value>
<string><! [CDATA[255.255.255.0]]></string>
</value>
</member>
<member>
<name>mta_acl_ip_block_action</name>
<value>
<string><! [CDATA[Quarantine]]></string>
</value>
</member>
<member>
<name>mta_acl_ip_block_comment</name>
<value>
<string><! [CDATA[Blocked IP address]]></string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

## Use Case - Adding a Regular Expression to a Content Filter

This example uses the config.create method described in the previous section. Using config.create you can add regular expressions to a content filter, which is a global setting. For more details about using regular expressions and content filtering, see the **BLOCK/ACCEPT > Content Filtering** page. The output of a successful call is a simple '200 OK'.

### Arguments:

Regular Expression: \bvi.gra\b (see [Regular Expressions](#))

**Sample Request:**

```
<?xml version="1.0" encoding="UTF8"?>
<methodCall>
<methodName>config.create</methodName>
<params>
  <param>
    <value>
      <struct>
        <member>
          <name>parent_type</name>
          <value>
            <string><! [CDATA[global]]></string>
          </value>
        </member>
        <member>
          <name>name</name>
          <value>
            <string><! [CDATA[\bvi.gra\b]]></string>
          </value>
        </member>
        <member>
          <name>type</name>
          <value>
            <string>
              <! [CDATA[filter_header_list]]>
            </string>
          </value>
        </member>
        <member>
          <name>parent_path</name>
          <value>
            <string></string>
          </value>
        </member>
        <member>
          <name>filter_header_list_comment</name>
          <value>
            <string><! [CDATA[Filter this content]]></string>
          </value>
        </member>
        <member>
          <name>inbound_filter_header_list_action</name>
```

```
<value>
  <string><! [CDATA[Block]]></string>
</value>
</member>
<member>
  <name>outbound_filter_header_list_action</name>
  <value>
    <string><! [CDATA[Quarantine]]></string>
  </value>
</member>
<member>
  <name>apply_to_subject</name>
  <value>
    <string><! [CDATA[1]]></string>
  </value>
</member>
<member>
  <name>apply_to_header</name>
  <value>
    <string><! [CDATA[0]]></string>
  </value>
</member>
<member>
  <name>apply_to_body</name>
  <value>
    <string><! [CDATA[1]]></string>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

## Use Case - Listing Explicit Users (Valid Recipients) and Aliases at the Global Level

Supported by firmware version 5.1.3.006, 6.x and higher

Use the config.list method to list valid recipients and aliased accounts at the global level - i.e. not domain-specific. Explicit Users and aliased email accounts are added or deleted on the **ADVANCED > Explicit Users** page of the web interface. In this case, the Type, or scope, is blank (empty) to indicate global. Note that the 'variable' 'list\_valid\_recipient\_aliases' is not actually a variable as defined in the configuration; rather, it is an indicator to the API of what is being listed by the config.list

call.

**Sample Request:**

```
<?xml version="1.0" encoding="UTF8"?>
<methodCall>
    <methodName>config.list</methodName>
    <params>
        <param>
            <value>
                <struct>
                    <member>
                        <name>variable</name>
                        <value>
                            <string>
                                <![CDATA[list_valid_recipient_aliases]]>
                            </string>
                        </value>
                    </member>
                    <member>
                        <name>child_type</name>
                        <value>
                            <string>
                                <![CDATA[global]]>
                            </string>
                        </value>
                    </member>
                    <member>
                        <name>path</name>
                        <value>
                            <string>
                            </string>
                        </value>
                    </member>
                    <member>
                        <name>type</name>
                        <value>
                            <string>
                                <![CDATA['']]>
                            </string>
                        </value>
                    </member>
                </struct>
            </value>
        </param>
    </params>
</methodCall>
```

```
</params>
</methodCall>
```

## Use Case - Adding and Configuring Multiple Domains

Use the domain.add method, described in the previous section, in a loop to add multiple domains for which the Barracuda Email Security Gateway should process email. These domains will then be listed in the **DOMAINS > Domain Manager** page of the web interface.

To configure the domains, use the config.set method for each domain. This example configures the 'Spam Score limit block level' to **4** for n domains, by setting the scana\_pd\_block\_level variable, if you put the request in a loop. In the web interface, you'll see this value on the **BASIC > Spam Checking** page after clicking on the **Manage Domain** link for each domain.

### Sample Request (for each domain):

```
<?xml version="1.0" encoding="UTF8"?>
<methodCall>
  <methodName>config.set</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>scana_pd_block_level</name>
            <value>
              <i4>4</i4>
            </value>
          </member>
          <member>
            <name>path</name>
            <value>
              <string>
                <![CDATA[$domain]]>
              </string>
            </value>
          </member>
          <member>
            <name>type</name>
            <value>
              <string>
                <![CDATA[domain]]>
              </string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

## Use Case - Listing Valid Recipients and Aliases for a Domain

Supported by firmware version 5.1.3.006, 6.x and higher

Use the config.list method to list valid recipients and aliased accounts for a domain. Valid Recipients and aliased email accounts are added or deleted on the per-domain **USERS > Valid Recipients** page of the web interface. In this case, the Type, or scope, is 'domain', and this call returns a list of all valid recipients and aliased email accounts for the domain 'mymail.net'. Note that the 'variable' 'list\_valid\_recipient\_aliases' is not actually a variable as defined in the configuration; rather, it is an indicator to the API of what is being listed by the config.list call.

### Sample Request:

```
<?xml version="1.0" encoding="UTF8"?>
<methodCall>
<methodName>config.list</methodName>
<params>
<param>
<value>
<struct>
<member>
<name>variable</name>
<value>
<string>
<![CDATA[list_valid_recipient_aliases]]>
</string>
</value>
</member>
<member>
<name>child_type</name>
<value>
<string>
<![CDATA[global]]>
</string>
</value>

```

```
</member>
<member>
<name>path</name>
<value>
<string> <![CDATA[mymail.net]]>
</string>
</value>
</member>
<member>
<name>type</name>
<value>
<string>
<![CDATA[domain]]>
</string>
</value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

## Use Case - Adding Valid Recipients and Aliases for a Domain

---

Supported by firmware version 5.1.3.006, 6.x and higher

Use the config.set method to add valid recipients and aliases for a domain. This case adds a primary account and two email aliases for the domain 'testqa.com'. Aliased accounts are added or deleted on the per-domain **USERS > Valid Recipients** page of the web interface and are linked to a 'primary account', which receives quarantined mail for the aliased accounts. The primary valid recipient is added first, followed by a number of aliases. See the per-domain **USERS > Valid Recipients** page of the web interface for details about alias linking.

Note that the 'member' name 'new\_valid\_recipient\_aliases' is an indicator to the API of what is being set by the config.set call. Make sure the domain is present in the Barracuda Email Security Gateway before adding recipients and aliases.

### Arguments:

- path: testqa.com
- type: domain
- child\_type: global
- my \$domain = "testqa.com";

- my \$primary\_valid\_recip = 'user1@testqa.com';
- my \$alias = 'user2@testqa.com'.'.'user3@testqa.com';
- my \$primary\_and\_alias = \$primary\_valid\_recip." ".\$alias;
- new\_valid\_recipient\_aliases = 'user1@testqa.com user2@testqa.com user3@testqa.com';

**Sample Request:**

```
<?xml version="1.0" encoding="UTF8"?>
<methodCall>
  <methodName>config.set</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>new_valid_recipient_aliases</name>
            <value>
              <string>$primary_and_alias</string>
            </value>
          </member>
          <member>
            <name>path</name>
            <value>
              <string>
                <![CDATA[$domain]]>
              </string>
            </value>
          </member>
          <member>
            <name>type</name>
            <value>
              <string>
                <![CDATA[domain]]>
              </string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

**Use Case - Deleting Aliases and Valid Recipients for a Domain**

Supported by firmware version 5.1.3.006, 6.x and higher

Use the config.delete method to delete valid recipients and aliases for a domain. This example deletes the valid recipient and aliases for the domain 'testqa.com'. Valid recipients and aliased accounts are added or deleted on the per-domain **USERS > Valid Recipients** page of the web interface. Note that the variable 'delete\_valid\_recipient\_aliases' is not actually a variable as defined in the configuration; rather, it is an indicator to the API of what is being deleted by the config.delete call.

In this example, '[user2@testqa.com](#)', '[user3@testqa.com](#)' are the aliases to be deleted. Make sure the domain for which you are deleting aliased accounts is present in the Barracuda Email Security Gateway. The list of per-domain aliased user accounts to be deleted can be specified in the 'Values' variable in the XML request.

**Arguments:**

- path: testqa.com
- type: domain
- my \$domain = "testqa.com";
- my \$user2 = 'user2@testqa.com';
- my \$user3 = 'user3@testqa.com';

**Sample Request:**

```
<?xml version="1.0" encoding="UTF8"?>
<methodCall>
  <methodName>config.delete</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>path</name>
            <value>
              <string>
                <![CDATA[$domain]]>
              </string>
            </value>
          </member>
          <member>
            <name>type</name>
            <value>
              <string>
                <![CDATA[domain]]>
              </string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```
</member>
<member>
  <name>variable</name>
  <value>
    <string>
      <![CDATA[delete_valid_recipient_aliases]]>
    </string>
  </value>
</member>
<member>
  <name>values</name>
  <value>
    <array>
      <data>
        <value>
          <string>
            <![CDATA[$user2]]>
          </string>
        </value>
        <value>
          <string>
            <![CDATA[$user3]]>
          </string>
        </value>
      </data>
    </array>
  </value>
</member>
</struct>
</value>
</param>
</params>
</methodCall>
```

---

© Barracuda Networks Inc., 2024 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.