
Traffic Rewrite

<https://campus.barracuda.com/doc/92767501/>

With Traffic Rewrite, you can insert, remove, or rewrite headers before the request is forwarded to the backend server or the response is sent to a customer. You can also rewrite the URL to map to a different resource. These options can conceal information from external users. With Body Rewrite there is the ability to do a search and replace to content within a body response.

URL Translation

When a web application returns a URL, sensitive information about the web server may be revealed. Rewriting or redirecting modifies the prefix, domain, and response body of an internal URL to an externally viewable URL.

URL Translation can show a customer that they are on a public facing domain and URL such as `www.example.com/pictures`, but link to internal servers with a completely different name space, such as `example1.net/images`. The internal name space remains private and without the need for DNS records.

At this time URL Translation rules can only be configured using the **JSON Editor**. A visual editor will be available in the near future. If you are not familiar with JSON, see [Using The JSON Editor](#) below.

URL Translation rule parameters:

- **name** – The name for this rule.
- **inside_prefix** – The portion of the URL used by your internal servers that follows the **inside_domain**.
- **outside_prefix** – This is what the **inside_prefix** value will be changed to before being sent on to your customers.
- **inside_domain** – The domain name used by your internal servers.
- **outside_domain** – This is what the **inside_domain** value will be changed to before being sent on to your customers.
- **comments** – You can leave a note for anyone managing this rule.

Request Rewriting

Request Rewriting allows incoming requests to be rewritten or redirected. Headers can be added, removed, or edited at WAF-as-a-Service before the request is forwarded to the backend server. The URL can be rewritten to map to a different resource. A redirect response can also be issued to the

clients to point them to an updated location or resource. For example, Request Rewriting is used by default to relay the client IP address to the backend server by inserting the header X-Forwarded-For with the value of the client IP. The backend server can extract and use this value. Similarly, details such as client certificate parameters can be forwarded by inserting request headers using [rewrite macros](#).

Request Rewriting rules can be configured using a visual form editor (**Visual Editor**) or the **JSON Editor**.

The JSON editor is very powerful and can configure parameter values that the Visual Editor and rules table cannot display.

Using the Visual Editor

1. If the Visual Editor is not already displayed, click the **Visual Editor** button.
2. Set **Enable Request Rewriting** to **On** if it is not already in that state.
3. Click the **Add Rule** button. The Add Rule modal will open.
4. Configure the following fields in the Add Rule modal form.
 1. **Rule Name** – Enter a name for the Response Rewrite rule.
 2. **Action**– Set the action to:
 - *Insert Header* - Inserts a header to the request.
 - *Remove Header* - Removes the header from the request.
 - *Rewrite Header* - Rewrites the value of the existing header in the request.
 - *Rewrite URL* - Rewrites the value of the existing URL in the request.
 - *Redirect URL* - Redirects the value of the existing URL in the request.
 3. **Header Name** – Enter the relevant header name, for example X-Forwarded-For. This field is not available for *Rewrite URL* and *Redirect URL*.
 4. **Rewrite From** – Enter the initial request header, or initial URL. An asterisk (*) rewrites or redirects all named headers or URLs. You can also specify the value or expression to be rewritten or redirected. This field is not available for *Remove Header* and *Insert Header*.
 5. **Rewrite To** – Specify the new value of the header or URL to be rewritten or redirected. Click the **Constant Value** radio button to enter any constant value (not a macro) or click **Request Value** to choose from a list request elements. The list elements are *Client IP Address*, *Endpoint IP Address*, *URL*, and *SSL Protocol Version*. You can also use [rewrite macros](#) here using the JSON editor.
 6. **Extended Match Condition** – Set the condition under which a rewrite should occur. An asterisk (*) indicates there are no conditions (applies to all). This field can only be edited by using the JSON editor.
 7. **Comments** – You can make a note about this rule.
5. Click **Add**.

Once added, rules appear in the table. To edit or delete a rule, click the three dots on the right side of the table in the **More** column.

The arrows in the **Priority** column will move rules up or down within the table. This sets the order in which the rules will be acted upon. The rule at the top of the table is first, followed by each below it in descending order. Note: you can configure a rule to stop all processing of other rules once it has been matched and processed. To do so, use the JSON editor and set **continue_processing** to false.

Using the JSON Editor

If you are not familiar with JSON, see [Using The JSON Editor](#) below. The rule parameters for Request Rewriting are:

- **name** – The name for this rule.
- **sequence_number** – This is the order in which rules will be considered. The range is 1 - 1500 and the lowest number is run first. Once a rule match is found, that one is acted upon and no others are considered.
- **action** – Set the action to one of these:
 - *Insert Header* - Inserts a header to the request.
 - *Remove Header* - Removes the header from the request.
 - *Rewrite Header* - Rewrites the value of the existing header in the request.
 - *Rewrite URL* - Rewrites the value of the existing URL in the request.
 - *Redirect URL* - Redirects the value of the existing URL in the request.
- **header** – Leave empty if rewriting or redirecting a URL.
- **rewrite_value_type** – Enter constant to add any constant value to the **rewrite_value** parameter, or request if using a [rewrite macro](#).
- **old_value** – This is the original header or URL value that will be changed to before being sent on to your customers.
- **rewrite_value** – What the header or URL (**old_value**) will be changed to. If the **rewrite_value_type** above is set to request you can use [rewrite macros](#).
- **condition** – An asterisk (*) indicates there are no conditions (applies to all).
- **continue_processing** – Set to false to stop all processing of other rules once this rule has been matched and processed. Set to true to continue applying other rules. Example: If redirecting a customer to a different URL, there may be no reason to continue processing other rules. In that scenario, set to false.
- **comments** – You can leave a note about this rule.

Response Rewrite

This policy sets rewrite rules for outbound responses. It allows you to add, delete, or rewrite headers. Response Rewrites are used for many purposes. For example, if a response includes a header listing the source IP address, Response Rewrite can delete that header, thereby preventing external users from seeing the actual IP address of the server.

At this time Response Rewrite rules can only be configured using the **JSON Editor**. A visual editor will

be available in the near future. If you are not familiar with JSON, see [Using The JSON Editor](#) below.

Response Rewrite rule parameters:

- **name** – The name for this rule.
- **sequence_number** – This is the order in which rules will be considered. The range is 1 - 1500 and the lowest number is run first.
- **action** – Set the action to one of these:
 - *Insert Header* - Inserts a header to the request.
 - *Remove Header* - Removes the header from the request.
 - *Rewrite Header* - Rewrites the value of the existing header in the request.
- **header** – The name of the header to be acted upon.
- **old_value** – Original value that will be changed before being sent on to your customers.
- **rewrite_value** – What the header will be changed to. You can also use [rewrite macros](#) here.
- **condition** – An asterisk (*) indicates there are no conditions (applies to all).
- **continue_processing** – Set to `false` to stop all processing of other rules once this rule has been matched and processed. Set to `true` to continue applying other rules.
- **comments** – You can leave a note about this rule.

Body Rewrite

This policy sets the rule for searching and replacing any text string in the response body. Only responses whose content type begins with `text/` can be searched, including `text/html`, `text/plain`, `text/javascript`, `text/css`, `text/xml`. Binary content types (`image/gif`, etc.) cannot be searched. We can only search text. The search and replace strings should be text rather than regular expressions. Meta-characters cannot be used, such as `\r` or `\n` in either search or replace, which means you cannot search and replace any multi-byte charset strings.

At this time Body Rewrite rules can only be configured using the **JSON Editor**. A visual editor will be available in the near future. If you are not familiar with JSON, see [Using The JSON Editor](#) below.

Body Rewrite rule parameters:

- **name** – The name for this rule.
- **sequence_number** – This is the order in which rules will be considered. The range is 1 - 1500 and the lowest number is run first.
- **url** – The rule will only be applied if the URL portion after the domain matches this value. Example: `/documents`. You can use a wild card here as well. Example: `/documents/*` will search every page under `/documents/`
- **replace_string** – This is the string that will replace the **search_string** anywhere it is found in the body. You can use [rewrite macros](#) in this field.
- **host** – The rule will only be applied if the domain matches this value. Example: `customer.com`

- **comments** – You can leave a note for anyone managing this rule.
- **search_string** – This is the string that will be searched out and replaced. All instances of it will be replaced by the **replace_string**.

Using the JSON Editor

JSON is a format for storing and transporting data, often from a server to a web page. If you are not familiar with it, JSON may appear daunting. However, there are only a few rules that make up this powerful tool.

- Data is in name/value pairs. Example: name: Joe Friday or address: 123 Fourth Street
- Data is separated by commas.

Those are the main ones to keep in mind and enough to get you working with JSON on Traffic Rewrite. There are two more rules for those who are interested.

- Curley brackets { } hold objects. (Don't get hung up on the word "object". For the Traffic Rewrite functionality, you can think of an object as a "traffic rewrite rule and its parameters.")
- Square brackets hold arrays. (Think of an array as a set of WAF-as-a-Service traffic rewrite rules.)

The JSON editor can be a very useful way to create, edit or delete rules. On any of the **Traffic Rewrite** pages, click on **JSON Editor** to bring it up. You can then make additions or edits directly in the JSON window.

Here is an example using the **Body Rewrite** feature. Go to the JSON editor on the **Body Rewrite** page. If you haven't already added rules, you will see the following:

```
{
  rules: [ ],
  enabled : true
}
```

All rules will go between the "rules" brackets. `enabled: true` means that any such rules will be applied. If you are not ready for your rules to be active, change this to `enabled: false`

Click on the **Insert JSON Sample** button for a little help with rule constructing. You will see the following in the editor window.

```
{
  rules : [
```

```
{
  name : 'rule name',
  sequence_number : '1',
  url : '/url',
  replace_string : 'replace string' ,
  host : 'host.com' ,
  search_string : 'Search string' ,
  comments : 'This is an example rule'
},
enabled : true
}
```

Now we have a rule with sample parameters to configure. Lets say we have a company that sends out an email newsletter to its customers each day. We want to automatically change the words in the body of our message only for the customers in the United Kingdom. In this fictional example we know they are in the UK because the URL they use begins with `example-company.com/uk`. We have also given our rule a name and internal comment for future reference.

```
{
  rules : [
    {
      name : 'Location rule UK',
      url : '/uk',
      replace_string : 'European News Update',
      host : 'example-company.com',
      search_string : 'North American News Update',
      comments : 'Our UK customers read the EU update.'
    }
  ],
  enabled : true
}
```

We will now create a second rule that will apply only to customers in the United Kingdom that visit the `/download` page. Notice that the `sequence_number` of the second rule is 1 and the `sequence_number` of the first rule is 2.

```
{
  rules : [
    {
      name : 'Location rule UK',
      sequence_number : '2',
      url : '/uk',
      replace_string : 'European News Update',
      host : 'example-company.com',
```

```
        search_string : 'North American News Update',
        comments : 'Our UK customers read the EU update.'
    },
    {
        name : 'Location rule UK download',
        sequence_number : '1',
        url : '/uk/download',
        replace_string : 'Click the EU Update icon to begin downloading your
newsletter.',
        host : 'example-company.com',
        search_string : 'Click the US News icon to download your
newsletter.',
        comments : 'Directions should be region specific'
    }
],
enabled : true
}
```

Because the `sequence_number` of the second rule is 1, it will therefore be the first considered. If it is a match, the rule with `sequence_number: 2` will not be applied. Also, a comma was added between the two rule blocks. Without it, our JSON would not be valid.

Traffic Rewrite Macros

You can rewrite requests with a constant value or a request value.

Constant Values

You can choose to use a constant rewrite value, using the following macros.

For Request Rewrites:

The following macros are supported *ONLY* for Request Rewrite Values:

- `$SRC_ADDR` - Inserts the source (client) IP address. You can use it for the new value (Rewrite Value parameter) when inserting or rewriting a header
- `$URI` Should be specified in the new value, if you are rewriting or redirecting the URI. `$URI` specifies the complete request URI including the query string.
- `$X509_VERSION` - The client certificate's X.509 version string.
- `$X509_SERIAL_NUMBER` - The serial number of the client certificate.
- `$X509_SIGNATURE_ALGORITHM` - The signature algorithm used in the client certificate.

- \$X509_ISSUER - The issuer string of the client certificate.
- \$X509_NOT_VALID_BEFORE - Time before which the client certificate is not valid.
- \$X509_NOT_VALID_AFTER - Time after which the client certificate is not valid.
- \$X509_SUBJECT - The subject string of the client certificate.
- \$X509_SUBJECT_PUBLIC_KEY_TYPE - The X.509 Certificate Subject Key Identifier string of the client certificate.
- \$X509_SUBJECT_PUBLIC_KEY - Public key modulus of the client certificate.
- \$X509_SUBJECT_PUBLIC_KEY_RSA_BITS - Size of the client certificate's public key, in bits.
- \$X509_EXTENSIONS - The client certificate's X.509 Extensions string.
- \$X509_HASH - The X.509 Hash string of the client certificate.
- \$X509_WHOLE - The X.509 client certificate, represented as a string in PEM format.
- X509_SAN_EMAIL and X509_IAN_EMAIL - Macros copy the information of an email from the client certificate and send it back to the backend server in HTTP header.
- \$COUNTRY_CODE - The two-letter country code of the location from where the client is sending the request.

For Response Page

- %action-id - The attack ID of the violation that resulted in displaying this response page.
- %host - The host that sent this request.
- %s - The URL of the request that caused this violation.
- %client-ip - The client IP address of the request that caused the violation.
- %attack-time - The time at which the violation occurred.
- %attack-name - The attack name of the violation that resulted in displaying the response page.

Request Values

Request values are available in the menu and include:

- Client IP Address
- Endpoint Address
- URL
- SSL Protocol Version

If using the JSON editor you are not limited to these four values.

© 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.