Configuring iWAF

Avi Technical Reference (v17.2)

Configuring iWAF

view online

This document discusses intelligent web application firewall (iWAF) configuration on Avi Vantage and covers the following sections: * WAF Policy * WAF Profile * Enabling WAF on Virtual Server * WAF Mode * Paranoia Mode * WAF Administrator Role

WAF Policy

WAF policy is a specific set of rules that protects the application. This policy is enabled by associating it with a virtual service. Navigate to Templates > WAF > WAF Policy to locate the default policy. *System-WAF-Policy* is the default policy in Avi Vantage that contains OWASP CRS 3.0 rules.

> *Note: For customizing a policy, it is highly recommended to create a new policy instead of editing the default policy (System-WAF-Policy).*

Configuring WAF Policy

To create a new policy, navigate to Templates > WAF > WAF Policy and click on Create.

Settings Tab

Provide the following details to configure the WAF policy:

```
<th width="20%"> <center>Field</center> </th>
<th width="40%"> <center>Description</center> </th>
<th width="40%"> <center>Additional Information</center> </th>
```

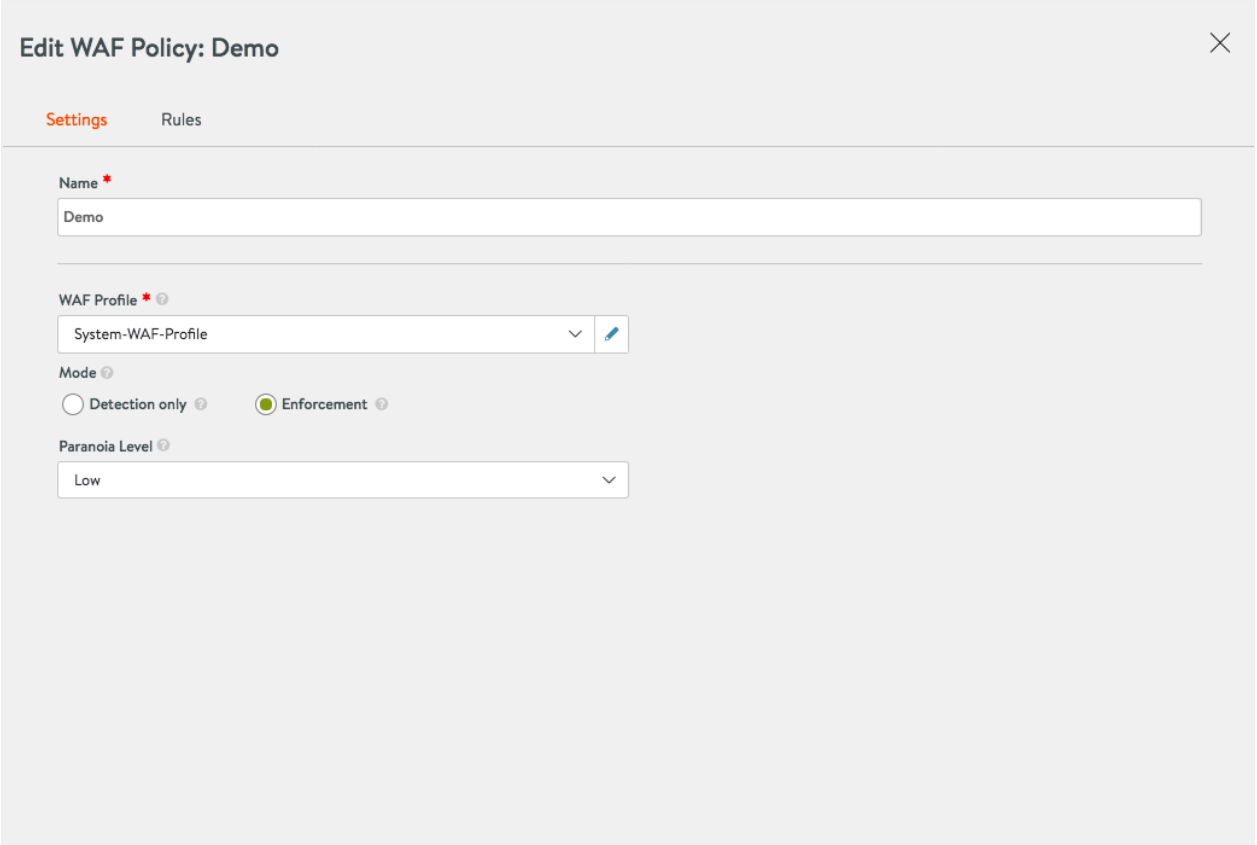
```
<td><b>Name</b></td>
<td>Enter a relevant name for the policy.</td>
```

```
<td><b>WAF Profile</b></td>
 <td>Enter the WAF profile that should be attached to this policy. The profile contains common reusable settings that c
 <td><a href="https://kb.avinetworks.com/docs/17.2/waf-configuring/#waf-profile">WAF Profile</a></td>
```

```
<td><b>Mode</b></td>
<td>Click on the required mode. The two supported modes are:
    <ul>
     <li>Detection only &mdash; In this mode, WAF policy will evaluate the incoming request. A log entry is created whe
    <li> Enforcement &mdash; In this mode, WAF policy will evaluate and block the request based on the defined default
     </ul></td>
 <td> It is recommended to use detection only mode in the beginning. For more details, refer to <a href="https://kb.avi
```

```
<td><b>Paranoia Level</b></td>
   <td>Set the paranoia level for the WAF policy. This is used to determine the rigidity of the policy and has a direct
    <td><a href="https://kb.avinetworks.com/docs/17.2/waf-configuring/#paranoia-mode">Paranoia Mode</a></td>
  </tr>
```

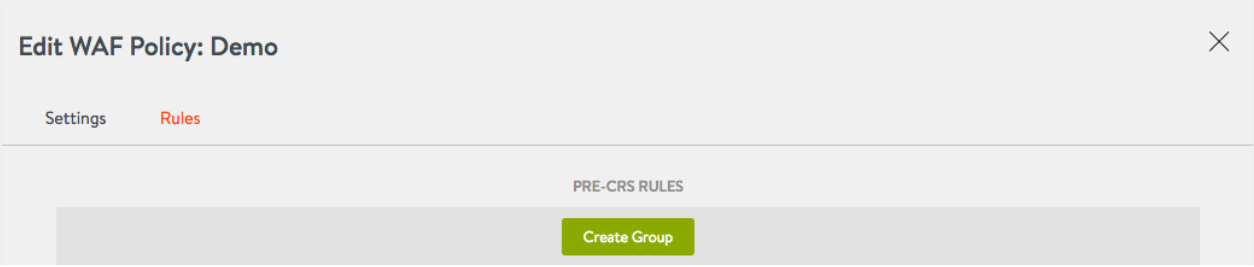
The following screenshot displays a sample configuration:

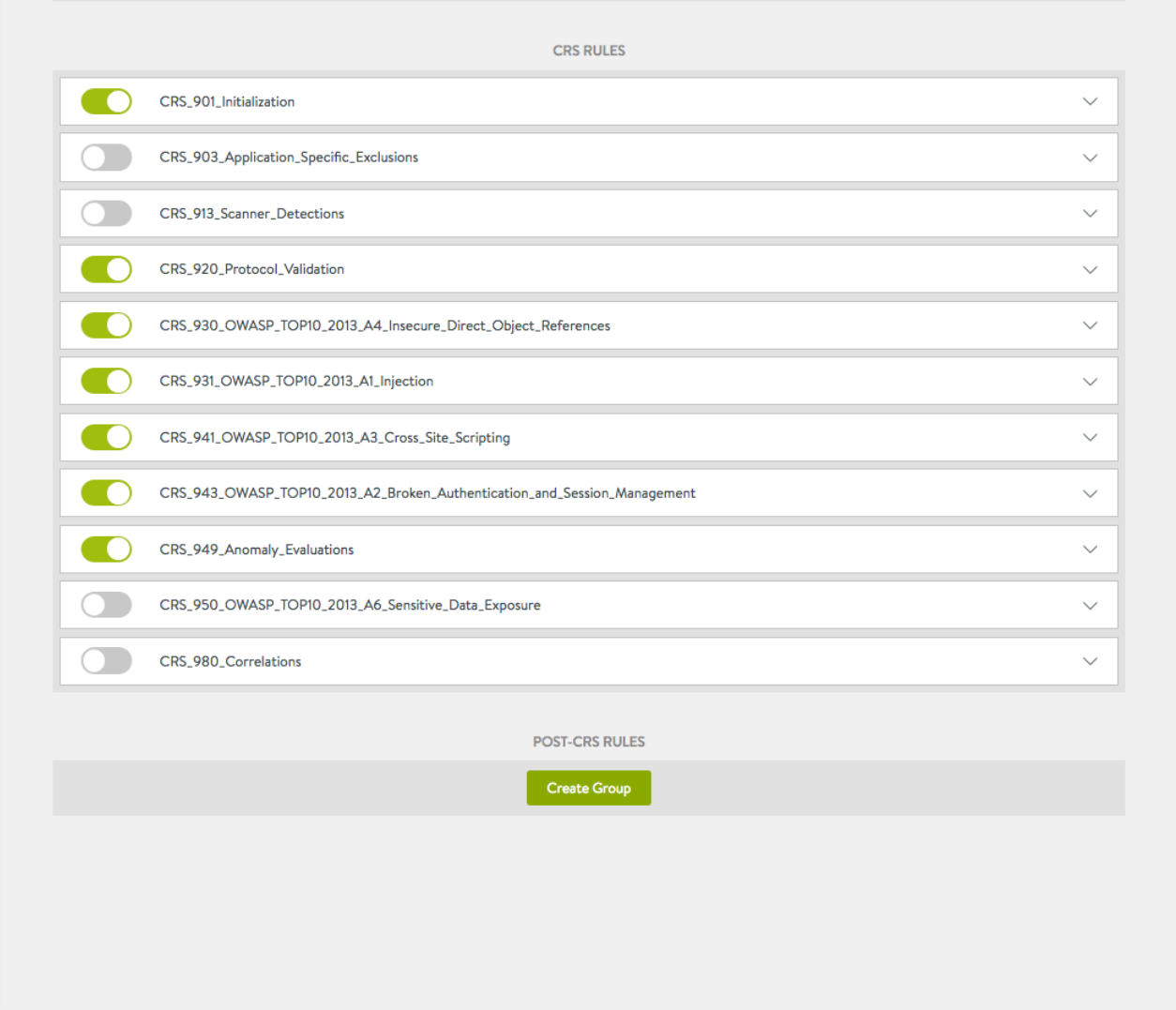


Rules Tab

The rules enabled under this tab will be applied on enabling the specific WAF policy. The following are the defined categories:

- PRE-CRS RULES: The custom rules that are applied before the supplied OWASP Core Rule Set (CRS).
- CRS RULES: The Avi Vantage supplied OWASP CRS policy that covers OWASP top ten attack protection.
- POST-CRS RULES: The custom rules that are applied after the supplied OWASP Core Rule Set (CRS).





Each of these categories can have pre-defined set of rules that can be enabled or disabled using the toggle button. If a rule is disabled, then it will not be evaluated during the request check.

Custom Rules

Custom ModSecurity rules can be created under the PRE-CRS RULES and POST-CRS RULES categories.

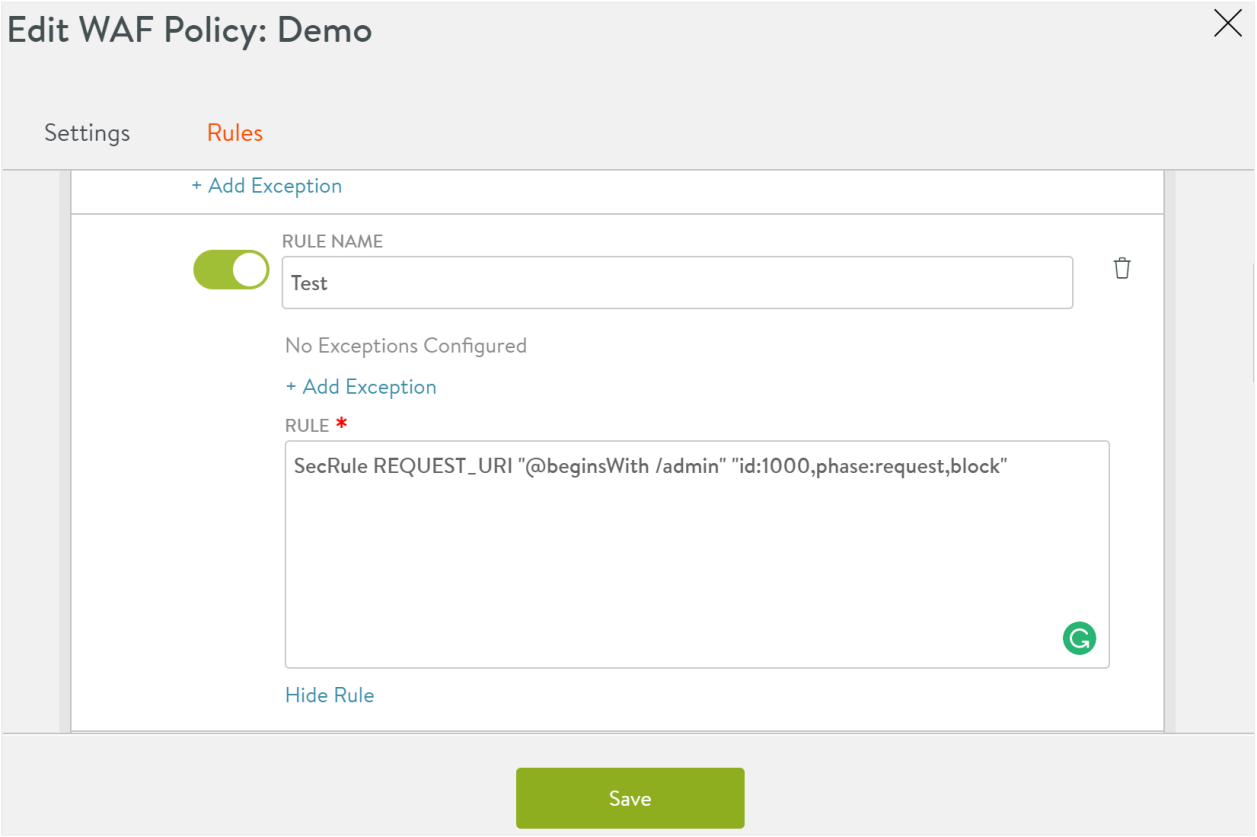
The three parts of ModSecurity rules are the variable to be examined, the test to submit the variable to, and the action to carry out if the test evaluates to be true for the selected variable.

For more information on ModSecurity Language refer to the ModSecurity Handbook.

To create custom rules, click on Create group and provide the following details:

- Enter the required group name. Every rule needs to be configured within a group.
- Rule Name:
- Provide a name for the rule.
- Specify the rule text in ModSecurity language.
  As displayed in the example below, *SecRule REQUEST_URI "@beginsWith /admin" "id:1000,phase:request,block"* is the

rule text. On detecting external access on admin interface, this rule is configured to blacklist the URL. Post rule configuration, the allowed IP range for the internal IT department can be added as an exception rule, so that they can use the URL for admin access.



- Any group or rule can have one or more exceptions.

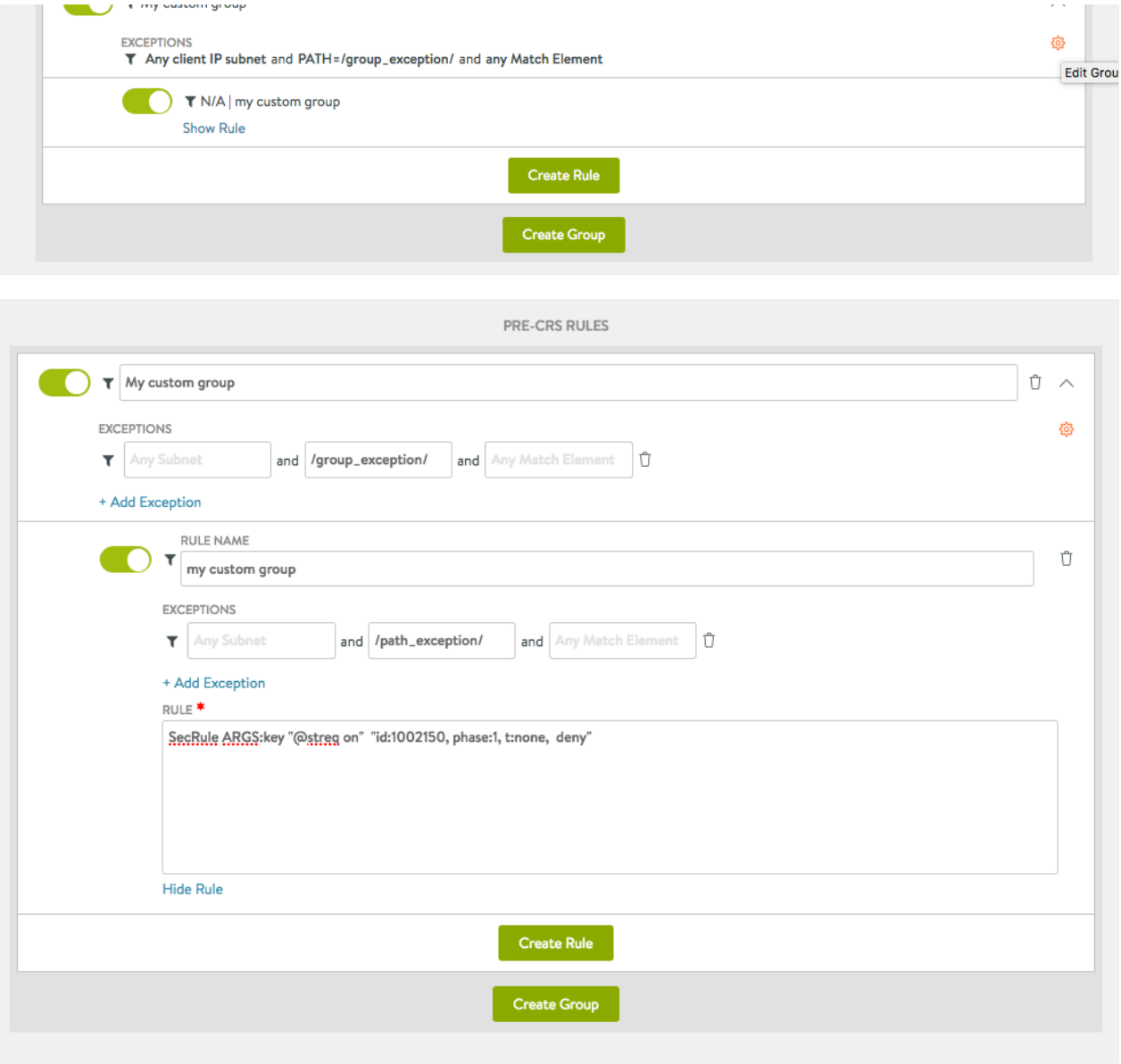
Exceptions

Exceptions are a common way of tuning a WAF policy to work with an application. These are normally created when an application's regular traffic matches specific WAF rules. The following are a few other reasons for creating exceptions: * For false-positive mitigation. * For applications that do not conform with the *System-WAF-Policy*. * For applications transmitting data that might appear like an attack. For instance, transferring HTML content in query parameters. * For applications with special requirements that are not allowed in the policy. For instance, accessing application on its direct IP address.

You can use Avi Vantage's recommendation system to create exceptions or you can even add them manually.

Manually Configuring Exceptions

- Click on + Add Exception to configure exceptions.
- These exceptions can be created on a group or rule level.
- With exceptions, a group or a rule check can be bypassed for the defined criteria.
- Exceptions can be configured for IP address/subnet, path, or any match element. For example, Subnet- *10.0.0.0/8*, Path- */admin/* , Match Element - *REQUEST_HEADERS*.
- On adding an exception, a funnel icon will appear which indicates that a rule or group contains an exception.
- Exceptions can be deleted using the trash can icon.



Recommended Assisted Workflow

The following steps are for a recommended workflow to configure exceptions:

1. Using WAF Log Analytics and find possible false-positives.
    - False-positives may occur in large numbers and for different client IP addresses.
    - To understand the context for false-positives, consult the application owner if possible.
2. In the log, choose the WAF Hit entry that you want to add the exception for, and click on + Add Exception.
    - The modal dialog will generate a set of suggested values.
    - These values are precomputed from the log entry and related findings.
3. Save the exception to apply it to the policy.

WAF Profile

A WAF profile contains the basic settings for WAF functionality and is attached to a WAF Policy. As the profile is independent of the policy, it can be easily reused. Generally, WAF profile is defined for a specific set of virtual services and is reused to a feasible extent.

The following are a few examples for WAF profiles: * Application Java profile: Contains all necessary elements for your Java applications. * Application PHP profile: Contains all necessary elements for your PHP applications. * API profile: Contains API specific settings.

Navigate to Templates > WAF > WAF Profile to locate the default profile. *System-WAF-Profile* is the default profile that contains most commonly used web application settings served through a virtual service.

> *Note: For customizing a profile, it is highly recommended to create a new profile instead of editing the default profile (System-WAF-Profile).*

Configuring WAF Profile

To create a new profile, navigate to Templates > WAF > WAF Profile and click on Create icon.

Settings Tab

Provide the following details to configure the WAF profile:

```
<th width="20%"> <center>Field</center> </th>
<th width="40%"> <center>Description</center> </th>
<th width="40%"> <center>Additional Information</center></th>
```

```
<td><b>Name</b></td>
<td>Enter a relevant name for the profile.</td>
<td></td>
```

```
<td><b>Allowed Versions</b></td>
<td>Enter the allowed HTTP versions for the profile.</td>
<td>1.0 and 1.1 are the default entries.</td>
```

```
<td><b>Allowed Methods</b></td>
<td>Enter the allowed HTTP method for the profile. Different applications might need different methods.</td>
<td>Websites might use only the default GET HEAD POST options. APIs might use other HTTP methods such as PUT, DELETE, e
```

```
<td><b>Allowed Content Types</b></td>
<td>Enter the accepted request content types for the profile.</td>
<td>Default entry covers all standard content types.</td>
```

```
```

```
<td><b>Restricted Extensions</b></td>
<td>Enter extensions that should be restricted and blocked.</td>
<td>Generally, these are files that do not reside on a web server. </td>
```

```
<td><b>Restricted Headers</b></td>
<td>Enter headers that will not be allowed by WAF.</td>
<td></td>
```

```
<td><b>Static Extensions</b></td>
<td>Enter the list of static file extensions that will bypass the WAF check.</td>
<td>A GET request without any parameter or dynamic part is classified as a static request. It does not contain any atta
```

```
<td><b>Default Actions</b></td>
<td colspan="2"> Request Header, Request Body, Response Header, and Response Body are the four WAF phases. Each of this
```

```
<td><i>phase</i></td>
<td>Enter the WAF phase.</td>
<td>Allowed values are phase:1, phase:2, phase:3, and phase:4.<br>
<i>Example- phase:1</i></td>
```

```
<td><i>action</i></td>
<td>Enter the action to be executed for that specific phase.</td>
<td>Two options are permit and deny.<br>
<i>Example- deny</i></td>
```

```
<td><i>status code</i></td>
<td>Enter the status code sent to the client.</td>
<td><i>Example- status:403</i></td>
```

```
<td><i>additional logging</i></td>
<td>Enter the additional logging level.</td>
<td><i>Example- log</i></td>
```

```
<td><i>WAF logs</i></td>
 <td>Enter the WAF logging level.</td>
 <td><i>Example- auditlog</i></td>
```

```
<td colspan="3"><b>Other Settings</b></td>
```

```
<td><b>Maximum non-file upload size</b></td>
<td>Enter the maximum file body size in KB examined by WAF. Larger body sized files will be blocked. </td>
<td><i>Example- 128</i></td>
```

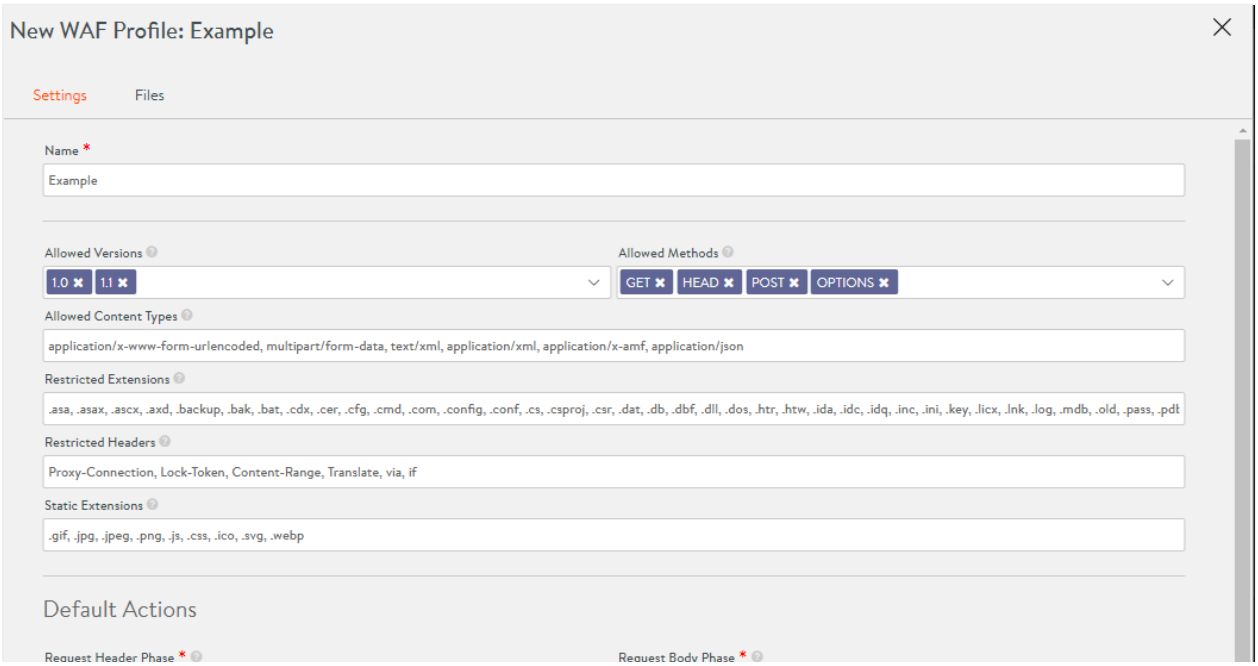
```
<td><b>Maximum file upload size</b></td>
<td>Enter the maximum file upload size in KB allowed by WAF.</td>
<td><i>Example- 1024</i></td>
```

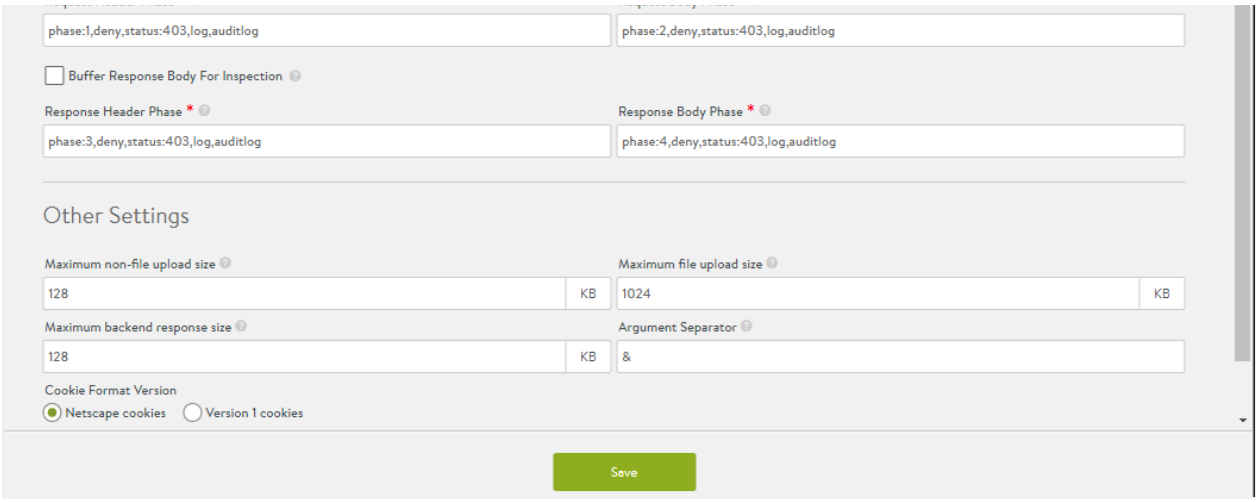
```
<td><b>Maximum backend response size</b></td>
<td>Enter the maximum response size in KB allowed by WAF.</td>
<td><i>Example- 128</i></td>
```

```
<td><b>Argument Separator</b></td>
<td>Enter the separator for specical applications that have different argument separators.</td>
<td><i>Example- &</i></td>
```

```
<td><b>Cookie Format Versions</b></td>
<td>Select the preferred cookie format version.</td>
<td><b>Version 1 cookies</b> have been deprecated and so <b>Netscape cookies</b> are recommended.</td>
```

The following screenshot displays a sample configuration:

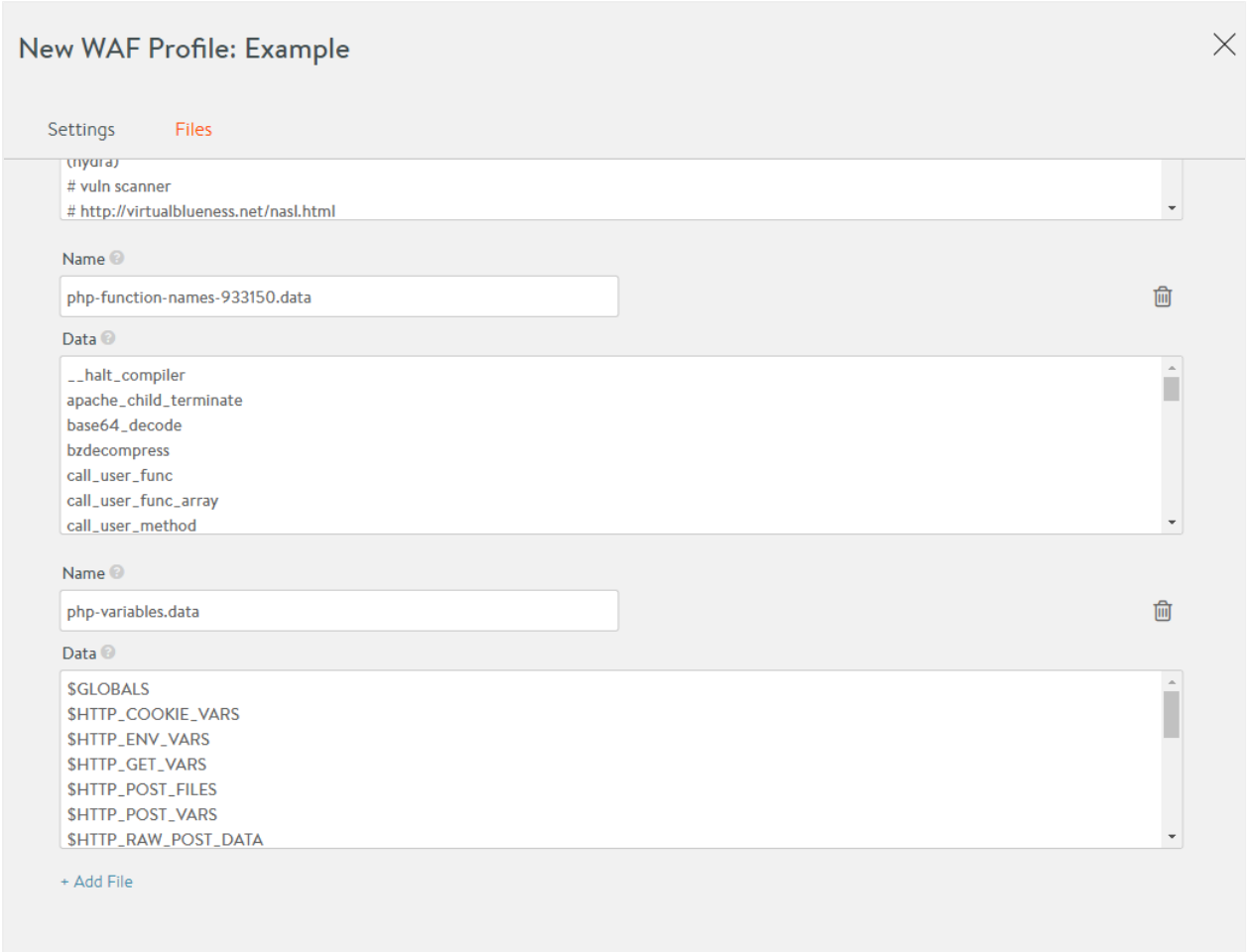


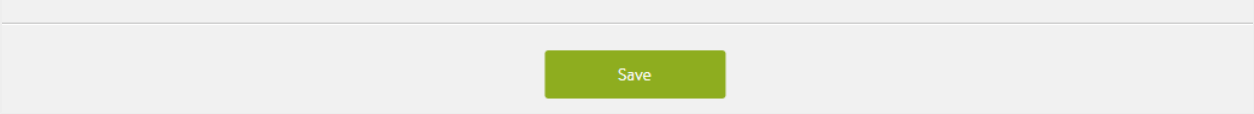


Files Tab

The static input data in a WAF profile that is shared between virtual services is stored here. For instance, the file name sql-errors.data has the default data set which contains strings for examining HTTP responses for data leakage protection.

To create a new file, scroll to the bottom of the page and click on + Add File. Provide a Name and enter the relevant Data. These files can be referred in the custom WAF policy rules.



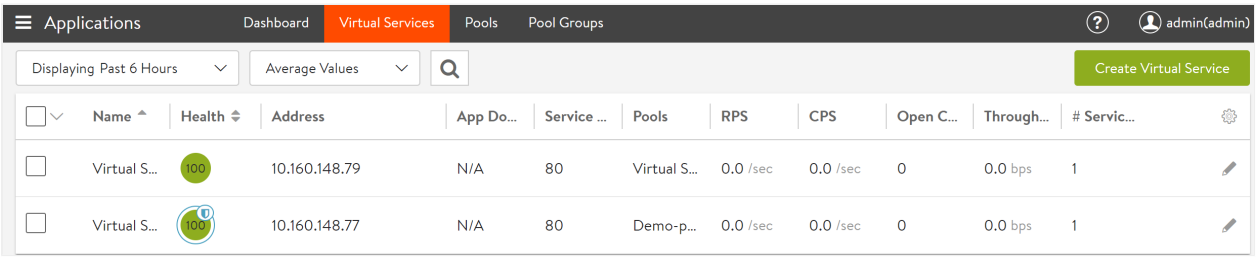


Enabling WAF on Virtual Server

Each virtual service can have one WAF policy attached to it.

To add a WAF policy to an existing virtual service, follow the given steps:

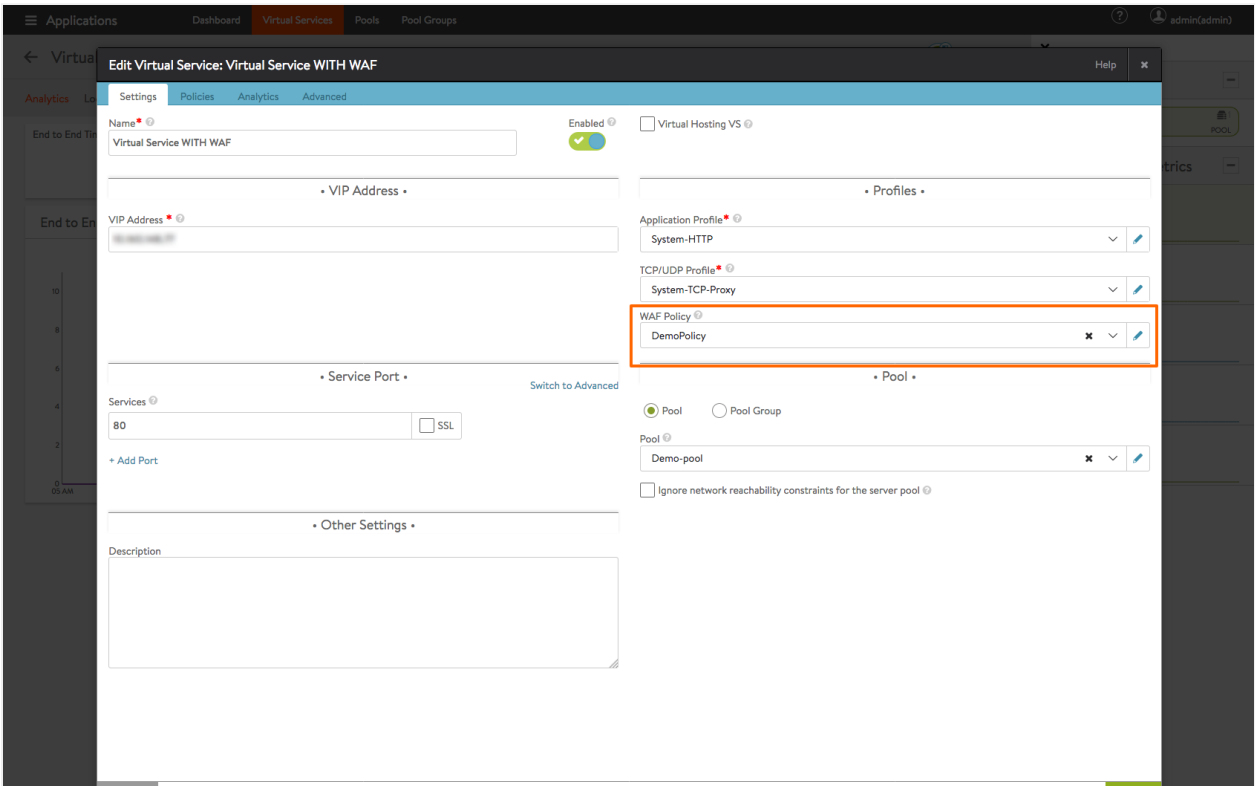
Navigate to Applications > Virtual Services. Select the virtual service and click on the edit icon.



In the Profiles section, click on the drop-down menu under WAF Policy, to either select an existing policy or to create a new one.

> *Caution: Attaching a WAF policy to a virtual service, will immediately put that policy into effect. If the policy is in enforcement mode, then it will start blocking requests. So, for new applications and untuned WAF policies, we recommend running the policy only in detection mode initially.*

Save the configuration.



Navigate to Applications > Dashboard to verify if WAF is enabled on the virtual service. If enabled, you will notice a halo and a shield on the attached virtual service object.

WAF Mode

Detection only and enforcement are the two modes supported for a WAF policy in Avi Vantage. Every policy runs in one of these modes to evaluate the requests and responses. The following section discusses the differences between these two modes.

```
<th width="20%"> <center></center> </th>
<th width="40%"> <center>Detection Only</center> </th>
<th width="40%"> <center>Enforcement</center> </th>
```

```
<td>Policy</td>
<td>Logs alerts during an attack, but no deny action is taken.</td>
<td>Rejects requests when a policy is matched and deny action is taken.</td>
```

```
<td>Operation</td>
<td>Evaluates the whole policy without stopping at the first rule hit.</td>
<td>Matches the first rule that rejects the request and implements the default action or returns a rule specific error
```

```
<td>Log files </td>
 <td>Contains the WAF log section where the policy violation was found and entries for every rule that is matched. </td
 <td>Contains specific WAF log section which has the first rule that rejected the request.
 <i>Note:</i> This is to improve performance. If a request is already detected as an attack, further checks are not req
```

```
<td>Response Code</td>
 <td>200 OK</td>
 <td>Default is 403 Forbidden. This response code can be modified.</td>
```

Usage Recommendations

Follow the steps provided to enable a suitable mode for different usage scenarios.

For New Applications

- Create a virtual service for the application.
- Add WAF policy in detection only mode.
- Iterate through false positive mitigation.
- Eliminate WAF findings which are not attacks.
- Once no legitimate traffic is flagged by WAF, change to enforcement mode.

For Existing Applications

- Add WAF policy in detection only mode.
- Iterate through false positive mitigation.
- Once no legitimate traffic is flagged by WAF, change to enforcement mode.

> *Note: The time taken for evaluating detection only mode depends on several factors such as total number of requests seen, paranoia mode, and application coverage of request.*

Paranoia Mode

Paranoia mode can be set for each WAF policy which defines its rigidity. Specific rules are enabled or disabled based on the set paranoia mode. The available modes are: * 1- Low * 2- Medium * 3- High * 4- Extreme

Two aspects that should be considered while setting the paranoia mode are: * Risk level of an application. * Resources available for policy tuning.

The following table maps paranoia modes to different risks levels and resource availability.

```
<td> High application risk level</td>
<td> High paranoia mode</td>
```

```
<td> Low application risk level</td>
<td> Low paranoia mode</td>
```

```
<td> Resources available for tuning</td>
 <td> Higher paranoia mode</td>
```

```
<td> Limited resources avilable for tuning</td>
 <td> Lower paranoia mode</td>
```

For more information on paranoia mode, refer to [OWASP ModSec CRS Paranoia Mode.]

WAF Administrator Role

WAF administrator role assigns users specific access to several components in Avi Vantage. This role differentiates access rights between the security team and other administrators. With this, the team can independently check the security status and implement policy changes.

The WAF administrator role provides read access to essential components such as virtual service, pool, and pool groups. Components such as WAF profile and WAF policy are provided write access.

To locate the WAF administrator role, navigate to Administration > Accounts > Roles. *WAF-admin* defines role access for all components as shown in the screenshot below.