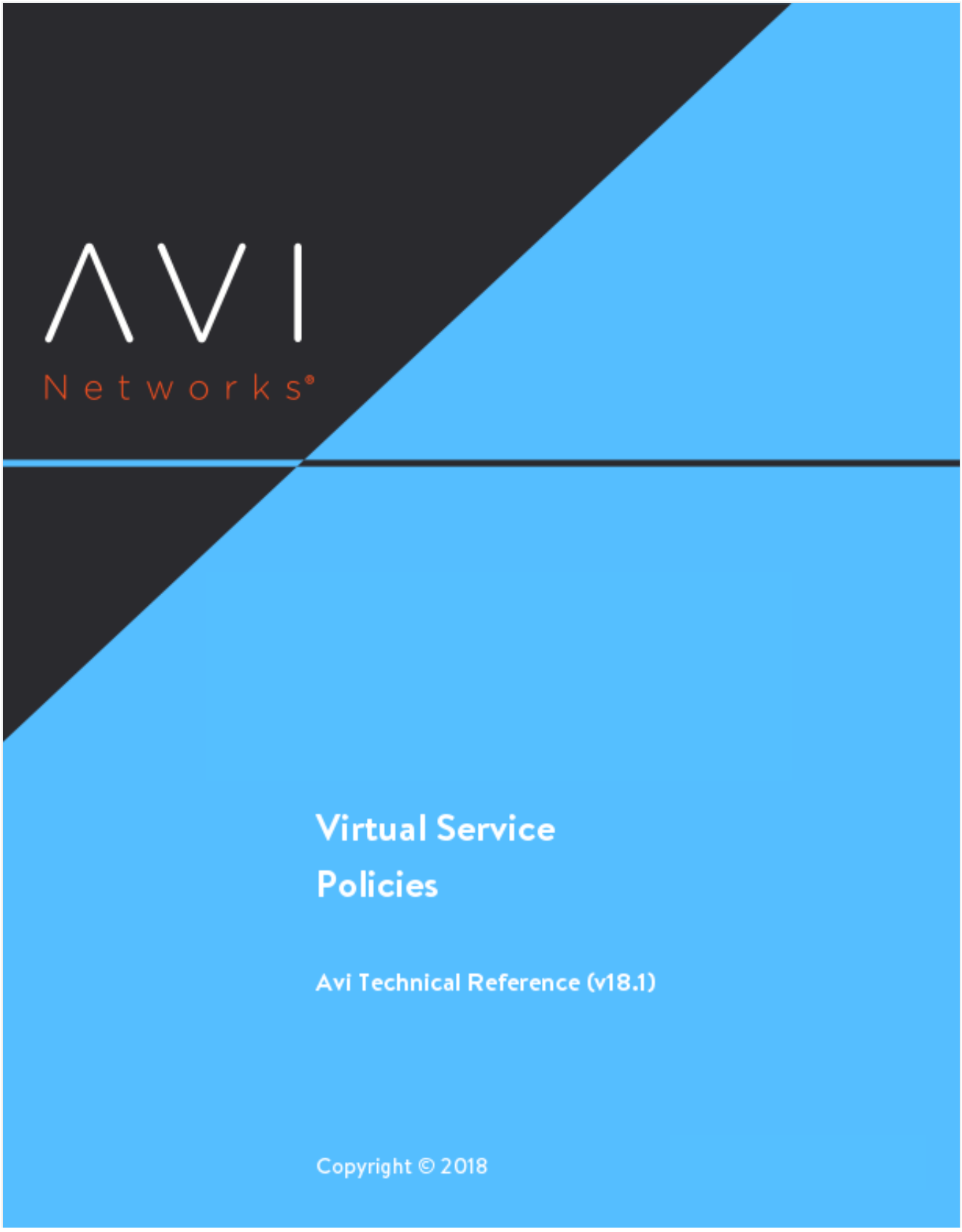


Virtual Service Policies

Avi Technical Reference (v18.1)

Copyright © 2018

Virtual Service Policies

view online

Policies allow advanced customization of network layer security, HTTP security, HTTP requests, and HTTP responses. A policy may be used to control security, client request attributes, or server response attributes. Policies are comprised of *matches* and *actions*, similar to an if/then logic. If something is true, then it matches the policy, and therefore Avi Vantage performs the following action.

Policies are comprised of one or more rules, which are match/action pairs. A rule may contain many matches, or have many actions. Multiple policies may be configured for a virtual service. Policies may alter the default behavior of the virtual service, or if matching criteria are not met, they may be benign for a particular connection, request, or response.

Policies are not shared; they are defined on a per-virtual-service basis. While powerful, policies are intended to be simple point-and-click functionality.

For more advanced capabilities, see DataScripts.

Policies are configured within the Policies tab of the virtual service editor.

Prioritizing Policies

Policies may be used to recreate similar functionality found elsewhere within Avi Vantage. For instance, a policy may be configured to generate an HTTP redirect from HTTP to HTTPS. This same functionality may be configured within the Secure-HTTP application profile. Because a policy is more specific than a general purpose profile, the policy will take precedence.

If the profile is set to redirect HTTP to HTTPS via port 443, and the policy is set to redirect HTTP to HTTPS on port 8443, the client will be sent to port 8443. (See Execution Priority for more on this topic.)

A virtual service may have multiple policies defined, one for each of the four types. Once defined, policies for the four types are implemented in the following order of priority.

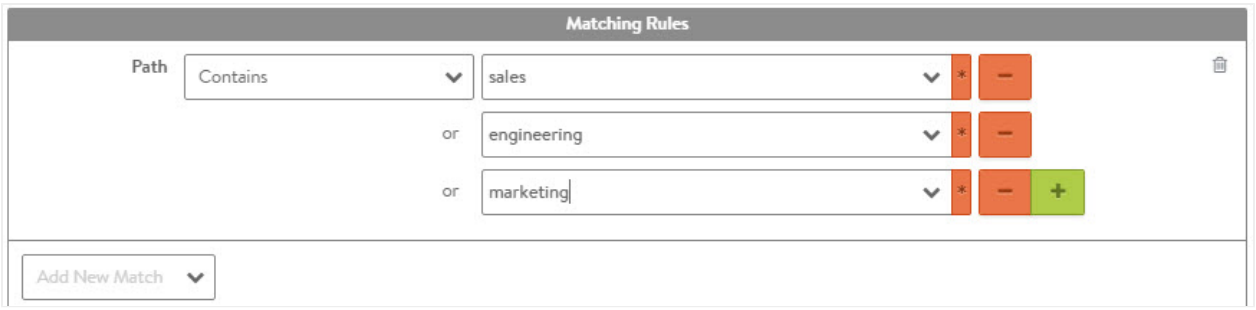
1. Network security policy
2. HTTP security policy
3. HTTP request policy
4. HTTP response policy

For example, a network policy that is set to discard traffic takes precedence over an HTTP request policy to modify a header. Since the connection is discarded, the HTTP request policy will not execute. Each policy type may contain multiple rules, which in turn can be prioritized to process in a specified order. This is done by moving the policies up or down in the ordered list within the Avi Vantage UI.

Match / Action

All policies are made up of match and action rules, which are similar in concept to "if / then" logic. Administrators set match criteria for all connections, requests, or response to the virtual service. Avi Vantage then executes the configured actions for all traffic that meets the match criteria.

Copyright © 2018 Avi Networks, Inc.



A single match with multiple entries is treated as "or" operation. For example, if a single match type has the criteria "marketing", "sales", and "engineering" set, then the match is true if the path contains "marketing", or "sales", or "engineering".

If a rule has multiple matches configured, then all match types must be true for the action to be executed. In the figure above, both the path and HTTP method matches must be true. Within each of these two match types, only one of the entries to be true for that match type to be true. For HTTP method, a client request must be of type GET or HEAD. Multiple rules may be configured for each policy, and they may be configured to occur in a specified order. If no match is applied, the condition is automatically met and the actions will execute for each connection as appropriate for the policy type.

Matches against HTTP content are case insensitive. This is true for header names and values, cookies, host names, paths, and queries. For HTTP policies, Avi Vantage compares Uniform Resource Identifier (URI) matches against the decoded HTTP URI. Many browsers and web servers encode human-readable format content differently. For example, a browser?s URI encoding might translate the dollar character "$" to "%24". The Service Engine (SE) translates the "%24" back to "$" before evaluating it against the match criteria.

Create a Policy

The virtual service editor defines policies that consist of one or more rules that control the flow of requests through the virtual service. To create a policy:

1. Policy Type: First select the policy type to add by selecting one of the following categories:
    - HTTP Security: HTTP security policies perform defined actions such as allow/deny, redirect to HTTPS, or respond with a static page.
    - Network Security: Is configured to explicitly allow or block traffic of user-defined types onto the network.
    - HTTP Request: HTTP request policies allow manipulation of HTTP Requests and content switching; they allow customized actions based on client HTTP requests.
    - HTTP Response: HTTP response policies evaluate responses from the server, and can be used to modify the server?s response headers. HTTP response policies are most often used in conjunction with HTTP request policies to provide an Apache Mod_ProxyPass capability for rewriting a website?s name between the client and server.

Copyright © 2018 Avi Networks, Inc.

2. Create Rule: Create a new rule by clicking the green add rule button and then entering the following information for the new rule:
    - Enable/Disable: By default, the new rule will be enabled. The green icon can be clicked to change to gray, which means this rule will be disabled, and will have no effect on traffic while in this state.
    - Rule Name: Enter a unique name for the rule in the rule name field, or leave the default system generated name in place.
    - Logging: Select the checkbox if you want logging enabled for this rule. When enabled, a log will be generated for any connection or request that matches the rule?s match criteria. If a virtual service is already set to log all connections or requests, this logging checkbox will not create a duplicate log. Client logs will be flagged with an entry for the policy type and rule name that matched. When viewing the policy?s logs within the logs tab of the virtual service, the logs will be part of the significant logs option unless the specific connection or request is an error, in which case it may be visible via the default non-significant logs filter.
    - Match: Add one or more matches using the pull-down menu. The match options will vary depending on the context defined by the policy type you are creating. If a rule is not given a match, then all connections or requests will be considered true or matched.
    - Action: Add one or more actions to be taken when the matches are true. The available options will vary depending on the type of rule you are creating.
    - Save: Click the save rule button to save the new rule.
3. Ordering: Rules are enforced in the order in which they appear in the list. For example, if you add a rule to close a connection based on a client IP address followed by a rule that redirects an HTTP request from that IP address to a secure HTTP (HTTPS) connection, then Avi Vantage will close the connection without forwarding the request. Alter the order in which rules are applied by clicking the up and down arrow icons until the rules are in the desired order.

Network Security

The following table lists both the available network security match criteria and the configurable actions that can occur when a match is made.

Note: Starting with release 18.1.2, this feature is supported for IPv6 in Avi Vantage.

Match	Client IP: Client IP address or a group of client addresses. • Use a "-" to specify a range: `10.0.0.0-10.1.255.255` • Use a "/" to specify a netmask: `10.0.0.0/24` Service Port: The ports on which the virtual service is listening.
Actions	Logging: Selecting the logging checkbox causes Avi Vantage to log when an action has been invoked. Allow / Deny: Explicitly allow or deny any matched traffic. Denied traffic will be issued a reset (RST), unless the system is under a volumetric or denial of service attack, in which case the connection may be silently discarded. Rate Limit: Restrict clients from opening greater than the specified number of connections per second in the Maximum Rate field. Clients that exceed this number will have their excessive connection attempts silently discarded. If burst size is enabled, clients may be able to burst above the maximum rate, provided they have not recently been opening connections. This feature may be applied to TCP or UDP. All clients that match the match criteria will be treated as one bucket. For instance, if no match is defined, any and all IP addresses will increment the max rate counter. Throttling will occur for all new connecting clients. To enable per client throttling, see the Advanced tab for the virtual service. The manual for this page also contains a more robust description of connecting throttling.

HTTP Security

The following table lists both the available HTTP security match criteria and the configurable actions that can occur when a match is made.

Copyright © 2018 Avi Networks, Inc.

Match	Client IP: Client IP address or a Group of client addresses. • Use a "-" to specify a range: `10.0.0.0-10.1.255.255` • Use a "/" to specify a netmask: <code Service Port: The ports on which the virtual service is listening. Protocol Type: HTTP or HTTPS. Example: *https://www.avinetworks.com/marketing/index.html?a=1&b=2* HTTP Method: The method used by the client request. The match is true if any one of the methods that an administrator specifies is true. HTTP Version: True if the client version is .9, 1.0, or 1.1 Path: The path or a group of paths. Paths do not need to begin with a forward slash (/). For comparison purposes, Avi Vantage automatically omits any initial slash specified in the match field. Example: https://www.avinetworks.com/marketing/index.html?a=1&b=2 Query: A query or a group of queries. Do not add the leading ??? or ?&? characters to a match. Example: *https://www.avinetworks.com/marketing/index.html?a=1&b=2* Headers: True if a header exists, or if it exists and contains a specified value Cookie: True if a cookie exists, or if it exists and contains a specified value Host Header: The request?s host header. Example: *https://www.avinetworks.com/marketing/index.html?a=1&b=2* Location Header: The location header may not exist for every website. HTTP Status: The status of the response, such as 200 (success), 404 (file not found), or similar. The statuses can be separated by commas, or be a range. For example: 301, 302, 307, 308, 300-599 Response Header: Match based on a specific header sent by the server.
Actions	Logging: Selecting the logging checkbox causes Avi Vantage to log when an action has been invoked. Action Allow: Allows matched requests to continue on to further policies or to the destination pool servers. Action Close Conn: Matched requests will cause Avi Vantage to close the TCP connection that received the request via a FIN. Many browsers open multiple connections, which are not closed unless requests sent over those connections also trigger a close connection action. Redirect To HTTPS: Respond to the request with a temporary redirect to the desired port for SSL. Action Send Response: Avi Vantage may serve an HTTP response using HTTP status code 200 (success), 403 (unauthorized), or 404 (file not found). A default page is rendered by the browser for each of these status codes, or you may upload a custom HTML file. This file may have links to images or other files, but only the initial HTML file will be stored and served via the Send Response.

Policy Tokens

In more complex scenarios, an administrator may wish to capture data from one location and apply it to another location. Avi Vantage supports the use of variables and tokens, which can be used for this purpose.

Variables may be used to insert dynamic data into the modify header actions of HTTP request and HTTP response policies. Two variables are supported, $client_ip and $vs_port. For example, a new header may be added to an HTTP request called origin_ip, with a value set to $client_ip, which will insert the client's source address as the value of the header.

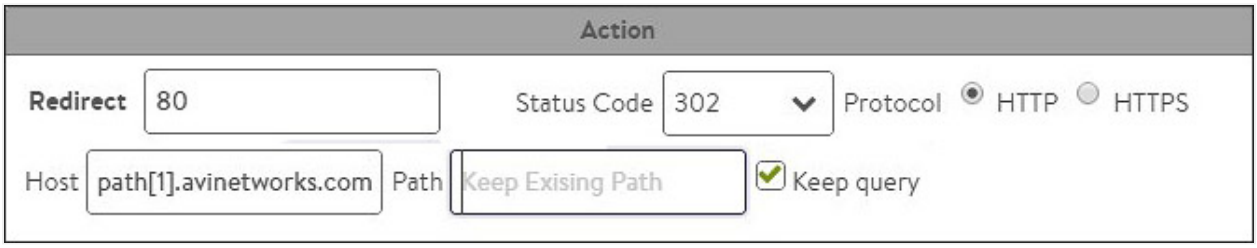
Copyright © 2018 Avi Networks, Inc.

Tokens may be used to find and reorder specific parts of the HTTP hostname or path. For example, it is possible to rewrite the original request http://support.avinetworks.com/docs/index.htm to http://www.avinetworks.com/support/docs/index.htm. Tokens can be used for HTTP host and HTTP path. The tokens are derived from the original URL. Token delimiter in host header is "." and in the URL path is "/".

Example 1

Original request URL:	support	avinetworks	com	docs	index.htm
Token:	host[0]	host[1]	host[2]	path[0]	path[1]

In the example above, the client request is broken down into HTTP host and HTTP path. Each section of the host and path are further broken down according to the "." and "/" delimiters for host and path. A host or path token may be used in an action to rewrite a header, a host, or a path. In the example below, a redirect of http://www.avinetworks.com/support/docs/index.htm would send requests to docs.avinetworks.com/support/docs/index.htm.



In addition to using the host[0], host[1], host[2] convention, a colon may be used to denote the system should continue till the end of the host or path. For instance, host[1:] implies to use avinetworks, followed by any further host fields. The result would be avinetworks.com. This is especially useful in a path, which may contain many levels. Tokens may also specify a range, such as path[2:5]. Host and path tokens may also be abbreviated as ?h? and ?p?, such as h[1:] and p[2].

In the rewrite URL, redirect, and rewrite location header actions, the host component of the URL can be specified in terms of tokens ? the tokens can be constants strings, tokens from existing host and path component of the URL.

Example 2

New URL: region.avinetworks.com/france/paris/index.htm

Request URL:	paris	france	avinetworks	com	region	index.htm
Token:	host[0]	host[1]	host[2]	host[3]	path[0]	path[1]
New Host:	path[0].host[2:]					
New Path:	/host[1]/host[0]/path[1]					

Example 3

Request URL:	www1	avinetworks	com	sales	foo	index.htm	auth=true
Token:	host[0]	host[1]	host[2]	path[0]	path[1]	path[2]	(query)

Copyright © 2018 Avi Networks, Inc.

New Host: www.host[1:]
New Path: /host[0]/path[0:]
Query: Keep Query enabled
New URL: www.avi.com/www1/sales/foo/index.htm?auth=true

- If the host header contains an IPv4 address and not a FQDN, and the rewrite URL or redirect action refers to a host token (e.g. host[0], host[1,2], etc.) the rule action is skipped and the next rule is evaluated.
- If the host header or path contains less tokens than that referenced in the action, then the rule action is skipped. For example, if the host name in host header has only 3 tokens (host name www.avinetworks.com - token host[0] = www, host1 = avinetworks, host2 = com). If the action refers to host[4] the rule action is skipped.
- If the location header in the HTTP response contains an IPv4 address and the response policy action is rewrite location header which refers to host tokens, the rule action is skipped.
- Rule engine does not recognize octal or hexadecimal IPv4 address in the host address. That is, the rule action will not be skipped if the host header has octal/hexadecimal IPv4 address and the action references a host token such as host1, etc.
- If an HTTP request arrives with multiple host headers, the first host header will be used.
- Per RFC, HTTP 1.1 requests must have a non-empty host header, else a 400 ?Bad Request? HTTP response will be returned by Avi Vantage.
- The HTTP processing is performed against decoded URIs.

Copyright © 2018 Avi Networks, Inc.