Rule Matching

https://campus.barracuda.com/doc/16418/

A rule consists of three patterns: host, URL, and extended match rules. The policies of the "best matching" rule are applied to a request. Host and URL rules are used to match Host and URL fields, respectively. Extended match rules can be compared to any combination of HTTP headers and/or query string parameters in a request. A "*" rule (to be read as a rule consisting of URL "*") matches any value for that parameter.

The best matching rule is the one with the longest matching host and URL and is the first matching rule in hierarchical order. If more than one extended match rule is configured with the same host and URL keys, the extended match rules are searched based on the extended match sequence.

Rule Match Algorithm

Host and URL rules (used for both URL Policies and Rule Groups) are treated as <prefix, suffix> pairs by the rule-match engine:

- A Prefix rule key is the part of the rule preceding the asterisk (*). The asterisk is treated as a wildcard, meaning any value.
- A Suffix rule key is the part of the rule following the asterisk.

If a rule does not have an asterisk, its suffix rule key is NULL.

The following algorithm is used by the Rule Match engine:

1. Find best matching Host Prefix Rule Key. The best match is defined as the longest rule matching the HTTP request Host header, left to right. The number of characters matched is the length of the Prefix Rule Key. If no Prefix Rule matches, the Rule Match engine terminates with failure and the request is dropped.
2. Find the best matching Host Suffix Rule Key. Best match is defined as the longest Suffix Rule Key matching the HTTP request Host header right to left. The number of characters matched is the length of Suffix Rule Key. If a matching rule is found, the current < Prefix, Suffix > pair matches the Host Rule Key, so go to Step 3. If no Host suffix Rule matches, discard this Prefix Rule Key and go to Step 1 to find the next matching Host Prefix Rule.
3. Find the best matching URL Prefix Rule Key. If none found, discard this Host Rule Key and go to Step 1 to find the next matching Host Rule Key. Best matching URL Prefix Key is defined as the longest URL Prefix Rule matching the HTTP Request-URI header, left to right. The number of characters matched is the length of URL Prefix Rule Key.
4. Find the best matching URL Suffix Rule Key. Best match is defined as the longest Suffix Rule Key matching the HTTP Request-URI header, right to left. If found, the current < Prefix, Suffix > pair is the matching URL Rule Key. We found the best matching Host Rule Key and URL Rule Key and continue at Step 5. If no match is found, discard this Prefix URL Rule and return to Step 3 to find the next matching URL Prefix Rule.

5. Find the first matching Extended Match Rule. The Extended Match rules are sorted based on the extended match sequence. If a matching rule is found, the Rule match engine terminates with success and the HTTP Request follows the policies defined on this rule. If no match is found, discard this Extended Match Rule and go back to Step 4 to find the next matching URL Suffix Rule Key.

A successful search terminates at Step 5. An unsuccessful search terminates at Step 1.

Hierarchical Match

The hierarchical match first compares the host header in the request to all host matches configured, and takes the best match. The ACL list is reduced to this set. Next, the URL path in the request is compared to all URLs in the reduced set of ACLs. This is again a best match. If multiple ACLs match, then each extended match rule is evaluated in ascending order of extended match sequence. The first extended match rule that matches will be the ACL match.

For example, consider the requests given below and see how they are matched with the ACLs in the following table:

ACLs	Host Match	URL Match	Extended Match	Extended Match Sequence
1	www.abc.com	/sales1/*	Header User-Agent co IE5.0 1	1
2	www.abc.com	/sales1/*	Header User-Agent co Mozilla 2	2
3	www.abc.com	/sales1/*	*	3
4	www.abc.com	/sales2/*	Header User-Agent co wget 0	0
5	*.abc.com	/sales2/*	*	0
6	*.abc.com	/sales3/*	*	0
7	*	/sales1/*	*	0
8	*	*	*	0

1. http://www.abc.com/sales1/index.html, from IE5.0
   1. Host header is www.abc.com, and therefore ACLs 1 to 4 match this request. Also, ACLs 4 to 8 match, but 1 to 4 are better matches since they are more specific.
   2. URL Path is /sales1/index.html, and therefore ACLs 1 to 3 match.
   3. Evaluate extended match rule for ACL 1 first, since extended match sequence = 1.
   4. As a result, ACL 1 is matched.
2. http://www.abc.com/sales2/index.html, from IE5.0
   1. Host header is www.abc.com, and therefore ACLs 1 to 4 match. Also, ACLs 4 to 8 match, but this is the best match, since it is most specific.
   2. URL Path is /sales2/index.html, and therefore only ACL 4 matches.
   3. Evaluate extended match rule for ACL 4. ACL 4 does not match, since User-Agent is expected to be wget, which is not true.

4. Thus, no matches in ACLs 1 to 4.
5. The next-best matching ACLs on the host header are ACLs 5 to 6.
6. URL Path matches only ACL 5.
7. Evaluate extended match rule for ACL 5. * matches anything.
8. As a result, ACL 5 is matched.

3. http://www.abc.com/sales3/index.html
   1. Host header is www.abc.com, and therefore ACLs 1 to 4 match. Also, ACLs 4 to 8 match, but this is the best match since it is most specific.
   2. URL Path is /sales3/index.html, which does not match any of the URLs in ACLs 1 to 4.
   3. The next-best matching ACLs in the host header are ACLs 5 to 6.
   4. URL path matches only ACL 6.
   5. Header rule matches ACL 6.
   6. As a result, ACL 6 is matched.

4. http://mirror.abc.com/sales4/index.html
   1. Host header is mirror.abc.com, and therefore ACLs 5 and 6 match.
   2. URL Path does not match any of the URLs in ACLs 5 and 6.
   3. The next-best matching ACLs in the host header are ACLs 7 and 8.
   4. URL path matches only ACL 8.
   5. Header rule matches ACL 8.
   6. As a result, ACL 8 is matched.

Sequential Match

Sequential match completely ignores the host header and URL path. Each extended match rule is evaluated in sequential order based on the extended match sequence. The first rule that matches is the ACL match.

To explain how the rule-match engine selects the best match, consider the following Rule Match table:

ACLs	Host Match	URL Match	Extended Match Rule	Extended Match Sequence
1	*	*	Header Host eq www.abc.com && Header User-Agent co IE5.0 && URI req /sales1/*	1
2	*	*	Header Host eq www.abc.com && Header User-Agent co Mozilla && URI req /sales1/*	2
3	*	*	* * Header Host eq www.abc.com && URI req /sales1/*	3
4	*	*	* * Header Host eq www.abc.com && Header User-Agent co wget && URI req /sales2/*	4
5	*	*	* * Header Host req .abc.com && URI req /sales2/*	5
6	*	*	* * Header Host req *.abc.com && URI req /sales3/*	6

7	*	*	* URI req /sales1/*	7
8	*	*	*	8

1. sales2http://www.abc.com/sales1/index.html, from IE5.0
    1. ACLs 1 to 8 match host and URL keys.
    2. Evaluate extended match rule for ACL 1 first, since extended match sequence = 1.
    3. As a result, ACL 1 is matched.
2. http://www.abc.com/sales2/index.html, from IE5.0
    1. ACLs 1 to 8 match host and URL keys.
    2. Evaluate extended match rule for ACL 1 to 8 in order.
    3. Extended match rule for ACL 5 matches.
    4. As a result, ACL 5 is matched.
3. http://www.abc.com/sales3/index.html
    1. ACLs 1 to 8 match host and URL keys.
    2. Evaluate header rules for ACL 1 to 8 in order.
    3. Header rule for ACL 6 matches.
    4. As a result, ACL 6 is matched.
4. http://mirror.abc.com/sales4/index.html
    1. ACLs 1 to 8 match host and URL keys.
    2. Evaluate header rules for ACL 1 to 8 in order.
    3. Header rule matches ACL 8.
    4. As a result, ACL 8 is matched.