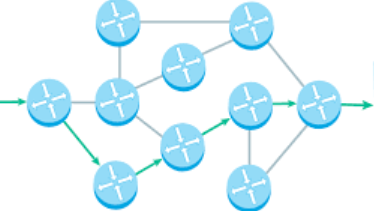
**Configuring BGP filters using Extended Routing Policy Language (XPL) on Huawei**

Created: 2023-09-25 13:40:17Latest reply: 2023-09-26 06:40:25

262 1 0 0

View the author0#

The theme of this short article is to explain a little about how Huawei's Extended Routing Policy Language (XPL) can help build a "robust" BGP routing policy that complies with best practices.



**eXtended routing-Policy Language - XPL**

According to the manufacturer's own documentation, the extended routing policy language (XPL) is used precisely to filter or modify attributes of routes/prefixes learned via BGP.

XPL has the same functionalities as "route-policies", but uses completely different editing and filtering methods. This is the key difference that brings the flexibility needed to build more robust routing policies.

In addition to the already well-known route-policy operators, XPL also brings the following fundamental blocks:

Start/end of block: demarcates the beginning/end of a configuration block.

Examples:

|  |  |
| --- | --- |
| **Start** | **end** |
| xpl global-value: start of the global variables section | end-global-value: end of the global variables section. |
| xpl ip-prefix-list ip-prefix-list-name: start of an IPv4 prefix list | end-list: end of an IPv4 prefix list |
| xpl community-list community-list-name: start of a community list | end-list: end of a community list |
| xpl route-filter route-filter-name: start of a route-filter | end-filter: end of a route-filter |

Logical operators: indicates logical relationships. **Examples:**

* eq/ge/le: equal to/greater than or equal to/less than or equal to
* in: included/gift set
* if: "if" / starts a conditional comparison block
* elseif: "else-if"/ comparison conditional
* else: "else"/ comparison conditional
* then: "then"/ used in the format if+<condition/criteria>+then or elseif+<condition/criteria>+then.
* apply: "action" that will be implemented if the evaluated <condition/criterion> matches.
* endif: "end-if"/ends the conditional comparison block.
* <condition/criteria> elements: these are the conditions to be "matched" in the route-filter.

The list above is just to give the reader a "guideline" regarding the blocks

**Example 1:**

Create a prefix list with routes 1.1.1.0/24, 2.2.2.0/24, and 3.3.3.3/32. Create a filter that applies a local-preference value of 900 for these prefixes, and 500 otherwise:

xpl ip-prefix-list CLIENT-DEMO

1.1.1.0 24,

2.2.2.0 24,

3.3.3.3 32

end-list

xpl route-filter CLIENT-DEMO-IMPORT

if ip route-destination in CLIENT-DEMO then

apply local-preference 900

else

apply local-preference 500

endif

end-filter

**Example 2:**

Create an as-path access-list with the routes received from AS 65501, the routes that passed through AS 65502, and the routes that were originated by AS 65503. Create a filter that denies the routes received from AS65501, the routes that passed through AS 65502, and the routes that were originated by AS 65503:

xpl as-path-list AS\_PATH-DENY-X

regular ^65501\_,

regular \_65502\_,

regular \_65503$

end-list

xpl route-filter DENY-X

if as-path in AS\_PATH-DENY-X then

refuse

else

approve

endif

end-filter

**Example 3**

Create a prefix-list and an as-path access list for the AS65555-192.0.2.0/24 transit client. Create a filter that validates and accepts the received routes and the client's as-path, and tags them with the transit community for the upstreams.

xpl as-path-list AS65555-TRAN-ASPATH

regular \_65555$

end-list

xpl ip-prefix-list AS65555-IPV4-NETS

192.0.2.0 24

end-list

xpl route-filter CUST-AS65555-IPV4-IMPORT

if ( ip route-destination in AS65555-IPV4-NETS ) and (as-path in AS65555-TRAN-ASPATH) then

apply community COMM-ALL-TRAN-CUSTS additive

approve

endif

end-filter

This article described the basic elements of XPL and how to use it to build more robust filters for your edge/transit router. The language learning curve is very low, and the benefits of using it are immediate. Anyone who has had experience with JunOS or IOS-XR must have noticed the numerous similarities between XPL and the languages ​​available on these platforms.

The example configuration demonstrated was tested and validated in the laboratory using a Huawei NE40 router. However, Huawei's documentation also mentions support for the NE20 model.

Still on the configuration, it should be noted that when we build a filter using the XPL language, its application on the BGP peer uses the "route-filter" keyword, while a "conventional/non-XPL" filter uses the keyword "route-policy" key.

Finally, as can be seen in the examples using XPL, it is possible to express the routing policy adopted locally in a clear and objective way. The use of XPL opens up another interesting possibility, which is the automation of the construction/maintenance of prefix lists, as-path acl's and route-filters using external data sources.

Thanks.

**Problem configuring BGP filters (XPL)**

Created: 2024-05-26 10:11:10Latest reply: 2024-06-09 07:44:12

5624 18 0 0

View the author0#

Hi,

Very commonly, those who use the BGP routing protocol are faced with the need to install filters. We often want to apply some rules to best manage traffic.

In this case we will see a situation where we need to filter the outgoing networks of our BGP.

**Issue description**

At the moment I am advertising my entire IP network outwards, I intend to restrict it to the following IPs:

10.10.10.0/24  
10.10.20.0/24  
10.10.30.0/24

**Handling Process**

In the image we can see the networks that are being announced:

|  |
| --- |
| [HUAWEI]display bgp routing-table peer XXX.XXX.XXX.XXX advertised-routes   BGP Local router ID is 10.1.1.1  Status codes: \* - valid, > - best, d - damped, x - best external, a - add path,                h - history,  i - internal, s - suppressed, S - Stale                Origin : i - IGP, e - EGP, ? - incomplete  RPKI validation codes: V - valid, I - invalid, N - not-found   Total Number of Routes: 32         Network            NextHop                       MED        LocPrf    PrefVal Path/Ogn   \*>i    10.10.10.0/24       XX.XX.XX.XX                                        0      100?  \*>i    10.10.20.0/24      XX.XX.XX.XX                                          0     100?  \*>i    10.10.30.0/24      XX.XX.XX.XX                                          0     100?  \*>i    10.10.40.0/24      XX.XX.XX.XX                                          0     100?  \*>i    10.10.50.0/24      XX.XX.XX.XX                                          0      100?  \*>i    10.10.60.0/24      XX.XX.XX.XX                                          0      100?  \*>i    10.10.70.0/24     XX.XX.XX.XX                                          0      100?  \*>i    10.10.80.0/24      XX.XX.XX.XX                                          0      100? |

The aim is to announce only 3 of these networks, already mentioned above.

**Root cause**

There is no rule that allows you to filter what you want to announce to the BGP peer. One of the ways to resolve this situation is by using Extended Routing Policy Language (XPL) on our Huawei router.



**Solution**

For those who don't understand BGP filters using Extended Routing Policy Language (XPL), this type of filter is very useful, it allows us to manage the incoming and outgoing traffic of our BGP peers. The application of this type of filters is mandatory, whether for traffic management or even for security reasons, sometimes we are not interested in receiving everything that the peer we are connected to sends us.

To solve the problem we have on our router with the BGP routing protocol, we will first create a PREFIX LIST, where we will place all the networks we want to advertise:

|  |
| --- |
| **xpl ip-prefix-list PFX\_NETWORK\_OUT** **10.10.10.0 24,** **10.10.20.0 24,** **10.10.30.0 24** **end-list** |

As you can see, we only include the networks we intend to advertise in the Prefix list.

Now we will have to create a Route Filter to apply our Prefix List.

|  |
| --- |
| **xpl route-filter RF\_NETWORK\_OUT** **if ip route-destination in PFX\_NETWORK\_OUT  then** **approve** **else** **refuse** **endif** **end-filter** |

This Route Filter must be applied in the BGP peer configuration.

|  |
| --- |
| **bgp XXX** **peer xxx.xxx.xxx.xxx as-number xxx** **peer xxx.xxx.xxx.xxx description "IP TRANSIT"** **#** **ipv4-family unicast** **peer xxx.xxx.xxx.xxx enable** **peer xxx.xxx.xxx.xxx route-filter RF\_NETWORK\_OUT export**  # |

Thank you！

Extended Routing Policy

xpl community-list