

## ular Expressions

In the previous section we looked at identifying routes based on IP address. In this section we will use regular expressions to identify routes based on AS path information. A regular expression is a pattern to match against an input string. When a regular expression is created, it specifies the pattern that a string must match. The following is a list of special characters that have special meaning when used in regular expressions:

Character	Symbol	Meaning
Any character	.	Match any character including white space.
Zero or more	*	Match zero or more sequences of the pattern.
One or more	+	Match one or more sequences of the pattern.
Optional	?	Matches zero or one occurrences of the pattern.
Start of string	^	Begins with.
End of string	\$	Ends with.
Backslash	\	Match the following.
Range	[ ]	Match a single value in range.
Group	( )	Separates the endpoints of a range.

## Filtering Based on AS Path

For this exercise, let's configure a regular expression in conjunction with a filter list on RouterC that will prevent any network that passes through AS 300 from being sent via BGP to RouterD. Filtering routes based on AS path information can be very useful when all routes from a particular AS need to be filtered. If filtering based on AS path was not used, the administrator would have to list each route one by one or potentially filter on a prefix. AS path filtering provides an efficient alternative to this.

In order to filter routes based on AS path information, we need to identify the AS path based on the defined regular expression and apply this to a BGP neighbor through a filter list:

1. Define the regular expression to deny any route that passed through AS 300.

```
RouterC#configure terminal
RouterC(config)#ip as-path access-list 1 deny _300_ ← Deny any route that
                                                    passes through AS 300
RouterC(config)#ip as-path access-list 1 permit .*
```

Use the `show ip bgp regexp` command to see what routes the regular expression matches. The following is the output from the command. Note that network 2.0.0.0 is the only route that matches the regular expression (`_300_`). This command is very useful in verifying that the regular expression covers the routes that you intend it to.

```
RouterC#show ip bgp regexp _300_
BGP table version is 19, local router ID is 195.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2.0.0.0	193.1.1.1	0		0	100 300 400 i
* I	192.1.1.1	0	100	0	100 300 400 i

2. Apply the filter list to BGP neighbor 195.1.1.1.

```
RouterC(config)#router bgp 200
RouterC(config-router)#neighbor 195.1.1.1 filter-list 1 out
```

In order for the changes to take effect, the BGP neighbor must be reset. To do this use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, restarting the neighbor negotiations from scratch and invalidating the cache.

```
RouterC#clear ip bgp *
```

Display the AS path access list on RouterC with the command `show ip as-path access-list`. The following is the output from the command. This command is very useful in quickly determining what strings will be permitted or denied.

```
ip as-path-access-list
s list 1
```

Display the BGP filter list configured on RouterC with the command **show ip bgp filter-list 1**. The following is the output from the command. This command shows which routes conform to a specified filter list and therefore will be passed.

```
ip bgp filter-list 1
sion is 5, local router ID is 195.1.1.2
s suppressed, d damped, h history, * valid, > best, i - internal
i - IGP, e - EGP, ? - incomplete
```

Next Hop	Metric	LocPrf	Weight	Path
193.1.1.1	0		0	100 i
192.1.1.1	0	100	0	100 i

Display the BGP table on RouterD with the command **show ip bgp**. The following is the output from the command. Notice that the route to network 1.0.0.0 via RouterC is no longer present in the routing table.

```
ip bgp
sion is 5, local router ID is 4.4.4.4
s suppressed, d damped, h history, * valid, > best, i - internal
i - IGP, e - EGP, ? - incomplete
```

Next Hop	Metric	LocPrf	Weight	Path
<b>192.1.1.1</b>	<b>0</b>	<b>100</b>	<b>0</b>	<b>100 300 400 i</b>
193.1.1.1	0	100	0	100 i
192.1.1.1	0	100	0	100 I

The following is a list of the regular expressions and their significance:

Expression	Significance
_300_	Match any routes that pass via AS 300.
_300\$	Match any routes that originated in AS 300.
^300_	Only match routes received from AS 300.
^300\$	Only match routes that originated from AS 300 and did not pass through any other AS.
*	All routes.