iControl 101 - #17 - PortMirror

Joe Pruitt, 2008-12-06

Port mirroring, also known as Interface mirroring, is a feature that allows you to copy traffic from any port or set of ports to a single, separate port where a sniffing device is attached. This is commonly used for network appliances that require monitoring of network traffic, such as an intrusion-detection system (IDS). The BIG-IP allows for creating mirroring on network interfaces and this article will discuss how to create and manage those mirrors with the iControl API.

Initialization

This article uses PowerShell and the iControl Cmdlets for PowerShell as the client environment for accessing the BIG-IP. The following setup will be required for the examples contained in this article. Since Port Mirroring is a feature of the interfaces, we will be using the Networking Interfaces interface as well to query the available interfaces on the device.

```
PS C:\> Add-PSSnapin iControlSnapin
PS C:\> Initialize-F5.iControl -hostname theboss -Username admin -Password admin
True
PS C:\> $PortMirror = (Get-F5.iControl).NetworkingPortMirror
PS C:\> $Interfaces = (Get-F5.iControl).NetworkingInterfaces
```

Querying the currently configured port mirrors

By default Port Mirroring is turned off, so you will likely get an empty list returned from the get_list() method. If you are going to want to create a Port Mirror, you'll need to know the available interfaces on the device. This can be accomplished with the get_list() method in the Networking.Interfaces interface.

```
PS C:\> $PortMirror.get_list()

PS C:\> $Interfaces.get_list()
mgmt
1.10
1.11
1.12
1.13
1.14
1.15
1.16
1.9
2.1
2.2
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
2.3
2.4
```

Creating a Port Mirror

To create a Port Mirror of an interface, you will need to look at the create() method in the Networking.PortMirror interface. This method takes as parameters an array of interfaces and a 2-d array of the interfaces to mirror to. Since there is not a default copy operator for a 2-d object array into a 2-d string array, you must explicitly declare the interfaces parameter as a 2-d array of strings. This can be done with the fun code below where a mirror is created on interface 1.5 to interfaces 2.1 and 2.2. I follow up the method call with a call to get_list to verify that the mirror has been created.

```
PS C:\> [string[][]]$ifacesAofA = @(, [string[]]@("2.1", "2.2"))
PS C:\> $PortMirror.create( (,"1.5"), $ifacesAofA)
PS C:\> $PortMirror.get_list()
1.5
```

Querying the Interfaces Within a Port Mirror

The get_interfaces() method in the Networking.PortMirror interface will return the interfaces being mirrored for the given network interface. The following code shows that for the network interface "1.5", it is being mirrored to interfaces "2.1" and "2.2" as configured in the port mirror creation above.

```
PS C:\> $PortMirror.get_interface(,"1.5")
2.1
2.2
```

Modifying a Port Mirror

The add_interface() method in the Networking.PortMirror interface has the same parameters as the create() method but instead of creating a new port mirror, it adds the supplied interfaces to an existing port mirror. In the following example, I'm adding the "2.3" and "2.4" interfaces to the existing port mirror on interface "1.5" created above. A call to get_interface() is then made to show that the newly added interfaces have been added.

```
PS C:\> [string[][]]$newifacesAofA = @(, [string[]]@("2.3", "2.4"))
PS C:\> $PortMirror.add_interface( (,"1.5"), $newifacesAofA)
PS C:\> $PortMirror.get_interface(,"1.5")
2.1
2.2
2.3
2.4
```

Similar to the add_interface() method, we have the remove_interface() method that will remove specified interfaces from an existing port mirror. The following code will remove the newly added "2.3" and "2.4" interfaces from the "1.5" interface port mirror and then call get_interface() to validate that the removal worked.

```
PS C:\> $PortMirror.remove_interface( (,"1.5"), $newifacesAofA)
PS C:\> $PortMirror.get_interface(,"1.5")
2.1
2.2
```

Deleting Port Mirrors

When your sniffer is tired of sniffing your content, you can easily delete a port mirror with the delete_port_mirror() method. This method takes an array of interfaces that are being mirrored and removes all port mirroring from those interfaces. The following code will remove the mirroring on the "1.5" interface created at the start of this article. You'll notice that it no longer exists by the results of the get_list() method.

```
PS C:\> $PortMirror.delete_port_mirror(,"1.5")
PS C:\> $PortMirror.get_list()
PS C:\> $PortMirror.get_list()
```

For those of you that like to live on the edge, there is also the remove_all_interfaces() and delete_all_port_mirrors() to remove all interfaces from an existing port mirror and deleting all existing port mirrors respectively. These can be handy but make sure you use them wisely.

Conclusion

So there you have it. If you ever find yourself in the need to automate the management of network interface port mirroring, this article should have all the information you need to get the job done.

Get the Flash Player to see this player.
20080612-iControl101_17_PortMirror.mp3