Role-Based CLI for IOS
February 14, 2011
By Tim Rowley, CCIE #25960 (Security), CCSI# 33858

Starting with 12.3(11)T Cisco introduced a feature to IOS called Role-Based CLI Access. This introduces the idea of "views" which basically translates to local command authorization. Typically, in order to provide command authorization you would either configure ACS via Shell Command Authorization Sets, or if ACS was not an option, you could tweak privilege levels of commands and users to give varying levels of access based on a user's privilege level locally. However, with some environments, this may be difficult to support because there could be 5 or 6 different teams who administer the router, yet, they should all have differing levels of access. The configuration can get gnarly pretty quick! Role-Based CLI allows you to create views, each of which can have different levels of command authorization. This is done locally, but, as we'll talk about in just a bit, you can optionally configure ACS to perform login authentication and send a special attribute back to the router which places the particular user into their view. View's can be broken into 3 general types: Root, CLI, and super. The root view is the master view more or less. It is the only view which allows you to edit other views, and has access to all commands. Even if you configure a CLI view and allow all commands (priv 15), this view still cannot edit other views, only the root view can do this. CLI views are the views you create to assign to users or groups, and Superviews are the combination of multiple CLI views. The superview is similar to a class-map nested in a class-map, or an object-group nested in an object-group. It gets the same level of access as the nested CLI views. Let's have a look at the config:

In order to configure the router to provide Role-Based CLI Access, you still need to enable aaa and give an enable secret:

Router(config)#aaa new-model

Next, you enable the root view and enter the enable pw. This is also the pw used to switch to the root view.

Router#enable view

Password: <enable secret>

*Feb 14 18:20:19.895: %PARSER-6-VIEW_SWITCH: successfully set to view 'root'.

Now you create the CLI views, give the view a pw (which is used to access the view) and assign commands. The syntax is similar to local command authorization.

Router(config)# parser view <name>
Router(config-view)# secret <password>
Router(config-view)# commands <mode> <attributes> <commands>

Let's look at an example of creating 3 views. One for Support, one for NOC, and a Superview which can perform both Support and NOC commands. Support should have the ability to configure ip addresses on interface f0/0 while NOC should only be able to issue the show version and show log command.

```
parser view Support
 secret cisco
 commands interface include ip address
 commands interface include ip
 commands configure include interface
 commands exec include configure terminal
 commands exec include configure
 commands configure include interface FastEthernet0/0
```

```
parser view NOC
 secret cisco
 commands exec include show version
 commands exec include show logging
 commands exec include show

parser view Super superview
 secret cisco
 view Support
 view NOC
```

Before we verify, there are a couple additional keywords worth discussing. Notice the available options while configuring commands:

```
Router(config-view)#commands exec ?
  exclude           Exclude the command from the view
  include           Add command to the view
  include-exclusive  Include in this view but exclude from others
```

You have the ability to include, exclude or include-exclusive, which means that other views will have an auto-exclude for the particular command.

There is also wildcard support via the keyword “all”. As an example, let’s give the Support view the ability to issue any show command:

```
Router(config)#parser view Support
Router(config-view)#command exec include all show
```

Now let’s verify. One of the neat things about views is that when you issue the ? you will only see options for which you have the ability to perform. Let’s look at Support first:

```
Router#enable view Support
Password:
*Feb 14 19:02:44.263: %PARSER-6-VIEW_SWITCH: successfully set to view 'Support'.
Router#show parser view
Current view is 'Support'
```

Using the ? we see that our options are limited. Configure and show, that’s a good start. Exit, enable are required of course to exit and switch views, so they are permitted without specifically doing so within the parser view.

```
Router#?
Exec commands:
  configure   Enter configuration mode
  credential  load the credential info from file system
  enable      Turn on privileged commands
  exit        Exit from the EXEC
  show        Show running system information
```

Now let’s add an IP to f0/0 and then attempt to do so on f0/1. F0/1 should fail:

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router(config)#int f0/0
Router(config-if)#ip address 50.50.50.50 255.255.255.0

Router(config-if)#int f0/1
                  ^
% Invalid input detected at '^' marker.
```

Excellent. Finally, let's do some show commands:

```
Router#show ip int brie
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        50.50.50.50     YES manual administratively down down
FastEthernet0/1        unassigned      YES NVRAM  administratively down down
Serial0/0/0            unassigned      YES NVRAM  administratively down down
Serial0/0/1            unassigned      YES NVRAM  administratively down down
Serial0/1/0            unassigned      YES NVRAM  administratively down down
Serial0/1/1            unassigned      YES NVRAM  administratively down down

Router#sh int f0/0
FastEthernet0/0 is administratively down, line protocol is down
  Hardware is MV96340 Ethernet, address is 0025.84bf.1f58 (bia 0025.84bf.1f58)
  Description: cisco
  Internet address is 50.50.50.50/24
  MTU 1500 bytes, BW 1000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  …
```

The show run command follows the traditional idea that you only have the ability to see what you can actually configure. Our show run reflects just that.

```
Router#sh run
Building configuration...

Current configuration : 85 bytes
!
interface FastEthernet0/0
 ip address 50.50.50.50 255.255.255.0
!
end
```

Now, let's switch to the NOC view:

```
Router#enable view NOC
Password:
*Feb 14 19:08:50.371: %PARSER-6-VIEW_SWITCH: successfully set to view 'NOC'.
Router#show parser view
Current view is 'NOC'
```

Verify only show version and show log (plus some defaults) are available

```
Router#show ?
  flash:   display information about flash: file system
```

```
  logging  Show the contents of logging buffers
  parser   Show parser commands
  version  System hardware and software status
```

Looks good, and quickly, let's verify the superview:

```
Router#enable view Super
Password:
*Feb 14 19:10:22.827: %PARSER-6-VIEW_SWITCH: successfully set to view 'Super'
Router#sh parser view
Current view is 'Super'

Router#show ?
  aaa                 Show AAA values
  aal2                Show commands for AAL2
  access-expression       List access expression
  access-lists            List access lists
  accounting              Accounting data for active sessions
  adjacency               Adjacent nodes
  alarm-interface         Display information about a specific Alarm
                      Interface Card
  aliases                 Display alias commands
  alignment               Show alignment information
  alps                Alps information
  appfw                 Application Firewall information
  appletalk               AppleTalk information
  arap                Show Appletalk Remote Access statistics
  archive               Archive of the running configuration information
  arp                 ARP table
  ase                 Display ASE specific information
  async                 Information on terminal lines used as router
                      interfaces
  auto                Show Automation Template
  autoupgrade             Show autoupgrade related information
  backhaul-session-manager  Backhaul Session Manager information
.....

Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int f0/1
                ^
% Invalid input detected at '^' marker.

Router(config)#int f0/0
Router(config-if)#ip address 60.60.60.60 255.255.255.0
```

Looks good, we could do all show commands, could not configure int f0/1, but could configure an ip address on f0/0.

So we're all set! Role-Based CLI is an excellent alternative to using ACS for command authorization and provides a much more granular level of control over traditional local command authorization. I mentioned previously that you have the ability to perform login authentication via TACACS and push an attribute down to place the particular user into a view. This is done at the Group or User level in ACS via the "cli-view-name" attribute. First, you must navigate to Interface Configuration, TACACS+, and under Advanced Options select the checkbox next to "Display a window for each service selected in which you can enter customized TACACS+ attributes". Now, navigate to the user or group, scroll down to the TACACS+ settings section and notice the Custom Attributes section. Check the box and enter cli-view-name=<viewname>.