VRF-Aware Management on ASR Configuration Examples

Document ID: 116093

Contributed by Atri Basu, Rudresh Veerappaji, and Wen Zhang, Cisco
TAC Engineers.
Jan 15, 2016

Contents

Introduction

This document describes the use of Virtual Routing and Forwarding-Aware (VRF-Aware) management on the Cisco Aggregation Services Router 1000 Series (ASR1K) with the management interface (**GigabitEthernet0**). The information is also applicable to any other interface in a VRF, unless explicitly specified otherwise. Various access protocols for both **to-the-box** and **from-the-box** connection scenarios are described.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Management protocols, such as SSH, Telnet, and HTTP
- File transfer protocols, such as Secure Copy Protocol (SCP), TFTP, and FTP
- VRFs

Components Used

The information in this document is based on these software and hardware versions:

- Cisco IOS® XE Version 3.5S (15.2(1)S) or later Cisco IOS-XE Versions

  Note: VRF-Aware SCP requires at least this version, whereas other protocols described in this document work with previous versions as well.
- ASR1K

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure you understand the potential impact of any command used.

Background Information

Management Interface: The purpose of a management interface is to allow users to perform management tasks on the router. It is basically an interface that should not, and often cannot, forward dataplane traffic. Otherwise, it can be used for remote access to the router, often via Telnet and Secure Shell (SSH), and to perform most management tasks on the router. The interface is most useful before a router begins routing, or in troubleshooting scenarios when the Shared Port Adapter (SPA) interfaces are inactive. On ASR1K, the management interface is in a default VRF named **Mgmt-intf**.

The **ip *<protocol>* source-interface** command is used in this document extensively (where the <*protocol*> keyword can be SSH, FTP, TFTP). This command is used in order to specify the IP address of an interface to be used as the source address when ASR is the client device in a connection (for example, the connection is initiated from the ASR or from-the-box traffic). This also means that if ASR is not the initiator of the connection, the **ip *<protocol>* source-interface** command is not applicable, and ASR does not use this IP address for the reply traffic; instead, it uses the IP address of the closest interface to the destination. This command allows you to source traffic (for the protocols supported) from a VRF-Aware interface.

Management Protocols

Note: Use the Command Lookup Tool (registered customers only) in order to obtain more information on the commands used in this article.

SCP

In order to use the SCP client service on an ASR from a VRF-enabled interface, use this configuration.

Configure

The **ip ssh source-interface** command is used in order to point the Management interface to the **Mgmt-intf** VRF for both SSH and SCP client services, since SCP uses SSH. There is no other option in the **copy scp** command to specify the VRF. Therefore, you must use this **ip ssh source-interface** command. The same logic applies for any other VRF-enabled interface.

```
ASR(config)#ip ssh source-interface GigabitEthernet0
```

Note: On the ASR1k platform, VRF-Aware SCP does not work until Version XE3.5S (15.2(1)S).

Verify

Use these commands in order to verify the configuration.

```
ASR#show vrf
  Name                              Default RD          Protocols   Interfaces
  Mgmt-intf                         <not set>           ipv4,ipv6   Gi0
ASR#
```

In order to copy a file from ASR to a remote device with SCP, enter this command:

```
ASR#copy running-config scp://guest@10.76.76.160/router.cfg
Address or name of remote host [10.76.76.160]?
Destination username [guest]?
Destination filename [router.cfg]?
Writing router.cfg Password:
!
Sink: C0644 2574 router.cfg
2574 bytes copied in 20.852 secs (123 bytes/sec)
ASR#
```

In order to copy a file from a remote device to ASR with SCP, enter this command:

```
ASR#copy scp://guest@10.76.76.160/router.cfg bootflash:
Destination filename [router.cfg]?
Password:
Sending file modes: C0644 2574 router.cfg
!
2574 bytes copied in 17.975 secs (143 bytes/sec)
```

TFTP

In order to use the TFTP client service on an ASR1k from a VRF-enabled interface, use this configuration.

Configure

The **ip tftp source-interface** option is used in order to point the Management interface to the **Mgmt-intf** VRF. There is no other option in the **copy tftp** command to specify the VRF. Therefore, you must use this **ip tftp source-interface** command. The same logic applies for any other VRF-enabled interface.

```
ASR(config)#ip tftp source-interface GigabitEthernet0
```

Verify

Use these commands in order to verify the configuration.

```
ASR#show vrf
```

```
  Name                              Default RD            Protocols   Interfaces
  Mgmt-intf                         <not set>             ipv4,ipv6   Gi0
ASR#
```

In order to copy a file from ASR to the TFTP server, enter this command:

```
ASR#copy running-config tftp
Address or name of remote host [10.76.76.160]?
Destination filename [ASRconfig.cfg]?
!!
2658 bytes copied in 0.335 secs (7934 bytes/sec)
ASR#
```

In order to copy a file from the TFTP server to ASR bootflash, enter this command:

```
ASR#copy tftp://10.76.76.160/ASRconfig.cfg bootflash:
Destination filename [ASRconfig.cfg]?
Accessing tftp://10.76.76.160/ASRconfig.cfg...
Loading ASRconfig.cfg from 10.76.76.160 (via GigabitEthernet0): !
[OK - 2658 bytes]

2658 bytes copied in 0.064 secs (41531 bytes/sec)
ASR#
```

FTP

In order to use the FTP client service on an ASR from a VRF-enabled interface, use this configuration.

Configure

The **ip ftp source-interface** option is used in order to point the Management interface to the **Mgmt-intf** VRF. There is no other option in the **copy ftp** command to specify the VRF. Therefore, you must use the **ip ftp source-interface** command. The same logic applies for any other VRF-enabled interface.

```
ASR(config)#ip ftp source-interface GigabitEthernet0
```

Verify

Use these commands in order to verify the configuration.

```
ASR#show vrf
  Name                              Default RD            Protocols   Interfaces
  Mgmt-intf                         <not set>             ipv4,ipv6   Gi0
```

In order to copy a file from ASR to an FTP server, enter this command:

```
ASR#copy running-config ftp://username:password@10.76.76.160/ASRconfig.cfg
Address or name of remote host [10.76.76.160]?
Destination filename [ASRconfig.cfg]?
Writing ASRconfig.cfg !
2616 bytes copied in 0.576 secs (4542 bytes/sec)
ASR#
```

In order to copy a file from the FTP server to ASR bootflash, enter this command:

```
ASR#copy ftp://username:password@10.76.76.160/ASRconfig.cfg bootflash:
Destination filename [ASRconfig.cfg]?
Accessing ftp://*****:*****@10.76.76.160/ASRconfig.cfg...
Loading ASRconfig.cfg !
[OK - 2616/4096 bytes]
```

```
2616 bytes copied in 0.069 secs (37913 bytes/sec)
ASR#
```

Management Access Protocols

Regular Access

SSH

Caution: One common problem seen with ASR1ks is that the SSH fails due to low memory. For more information in regards to this problem, reference the SSH Authentication Failure Due to Low Memory Conditions Cisco article.

There are two options used In order to run the SSH client service on the ASR (SSH from-the-box). One option is to specify the VRF name in the **ssh** command itself, so you can source SSH traffic from a particular VRF.

```
ASR#ssh -vrf Mgmt-intf -l cisco 10.76.76.161
Password:
Router>en
Password:
Router#
```

The other option is to use the **ip ssh source-interface** option in order to source SSH traffic from a particular VRF-enabled interface.

```
ASR(config)#ip ssh source-interface GigabitEthernet0
ASR#
ASR#ssh -l cisco 10.76.76.161
Password:
Router>en
Password:
Router#
```

In order to use the SSH server service (SSH to-the-box), follow the procedure to enable SSH on any other Cisco IOS router. Refer to the Telnet and SSH Overview for the Cisco ASR 1000 Series Routers section of the **Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide** for more information.

Telnet

There are two options used in order to run the Telnet client service on the ASR (Telnet from-the-box). One option is to specify the source interace or the VRF in the **telnet** command itself as shown here:

```
ASR#telnet 10.76.76.160 /source-interface GigabitEthernet 0 /vrf Mgmt-intf
Trying 10.76.76.160 ... Open

User Access Verification

Username: cisco
Password:

Router>en
Password:
Router#
```

The other option is to use the **ip telnet source-interface** command. You still must specify the VRF name in the next step with the **telnet** command, as shown here:

```
ASR(config)#ip telnet source-interface GigabitEthernet0
ASR#
ASR#telnet 10.76.76.160 /vrf Mgmt-intf
Trying 50.50.50.3 ... Open

User Access Verification

Username: cisco
Password:

Router>en
password:
Router#
```

In order to use the Telnet server service (Telnet to-the-box), follow the procedure to enable Telnet on any other router. Refer to the Telnet and SSH Overview for the Cisco ASR 1000 Series Routers section of the **Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide** for more information.

HTTP

The legacy web user interface that is available for all routers is also available for the ASR1K. Enable HTTP server or client service on the ASR as shown in this section.

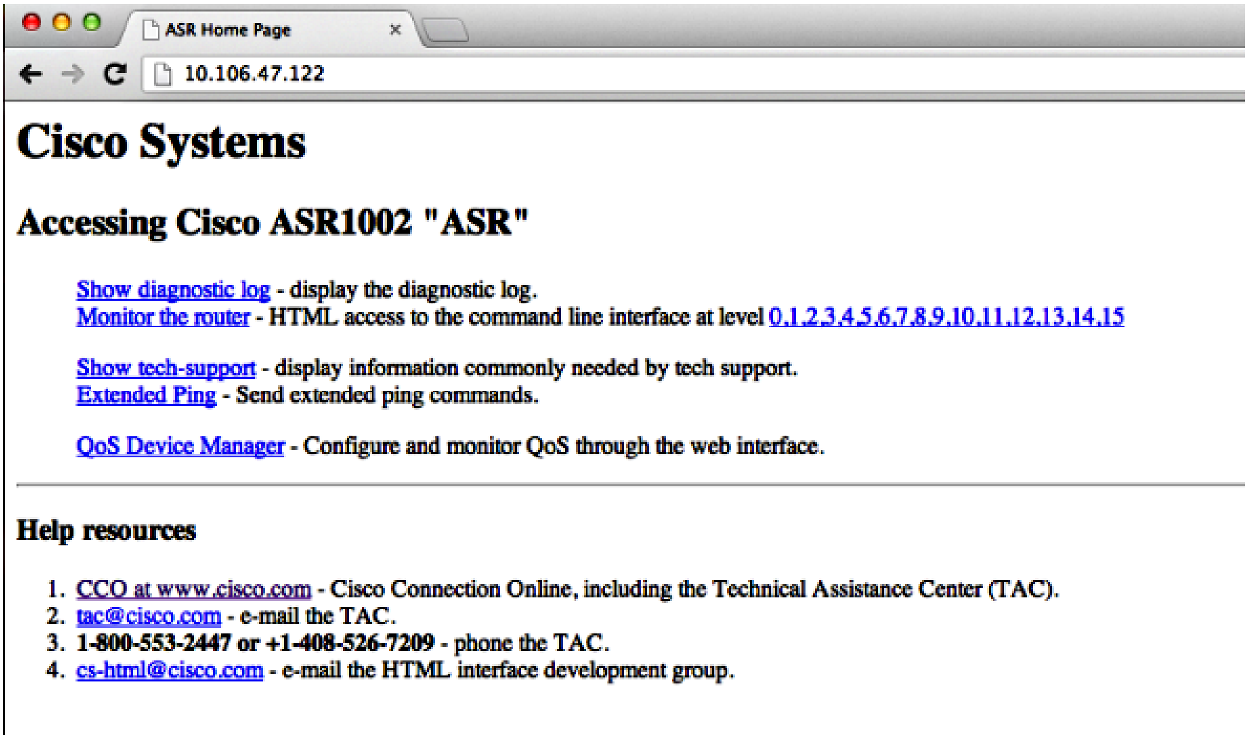
In order to enable legacy HTTP access to-the-box service (server) and use the web-based GUI access, use this configuration that uses local authentication (you could also use an external Authentication, Authorization, and Accounting (AAA) server).

```
ASR(config)#ip http
ASR(config)#ip http authentication local
ASR(config)#username <> password <>
```

Here is the configuration to enable the HTTP secure server (HTTPS):

```
ASR(config)#ip http secure-server
ASR(config)#ip http authentication local
ASR(config)#username <> password <>
```

Browse to the IP address of an interface on the ASR, and log in with the user account that you created. Here is a screenshot:



Cisco Systems

Accessing Cisco ASR1002 "ASR"

Show diagnostic log - display the diagnostic log.
Monitor the router - HTML access to the command line interface at level 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

Show tech-support - display information commonly needed by tech support.
Extended Ping - Send extended ping commands.

QoS Device Manager - Configure and monitor QoS through the web interface.

Help resources

1. CCO at www.cisco.com - Cisco Connection Online, including the Technical Assistance Center (TAC).
2. tac@cisco.com - e-mail the TAC.
3. **1-800-553-2447 or +1-408-526-7209** - phone the TAC.
4. cs-html@cisco.com - e-mail the HTML interface development group.

In order to use the HTTP client service, enter the **ip http client source-interface <interface name>** command source for the HTTP client traffic from a VRF-enabled interface, as shown:

```
ASR(config)#ip http client source-interface GigabitEthernet0
```

Here is an example that illustrates the use of HTTP client service in order to copy an image from a remote HTTP server to the flash:

```
ASR#
ASR#copy http://username:password@10.76.76.160/image.bin flash:
Destination filename [image.bin]?
Accessing http://10.106.72.62/image.bin...
Loading http://10.106.72.62/image.bin
1778218 bytes copied in 20.038 secs (465819 bytes/sec)
ASR#
```

Persistent Access

This section is applicable only for to-the-box Telnet/SSH/HTTP connections.

With persistent SSH and persistent Telnet, you can configure a transport map that defines the treatment of incoming SSH or Telnet traffic on the Management Ethernet interface. So this creates the ability to access the router via diagnostic mode even when the Cisco IOS process is not active. For more information on diagnostic mode, refer to the Understanding the Diagnostic Mode section of the Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide.

Note: Persistent SSH or persistent Telnet can only be configured on the Management interface, **GigabitEthernet0.**

Note: In versions that do not have the fix for Cisco bug ID CSCuj37515, the authentication method for peristent access is dependant upon the method that is used under line **VTY**. Persistent access requires that the authentication is local, so that diagnostic mode access still works when external authentication fails. This means that any normal SSH and Telnet access also requires the use of local authentication.

Caution: In versions that do not have the fix for Cisco bug ID CSCug77654, the use of the default AAA method restricts the user ability to enter the SSH prompt when persistent SSH is used. The user is always forced to enter the diagnostic prompt. For these versions, Cisco recommends that you use a name authentication method, or ensure that normal SSH and Telnet are enabled.

Persistent SSH

Create a transport map in order to allow persistent SSH as shown in the next section:

Configure

```
ASR(config)#crypto key generate rsa label ssh-keys modulus 1024
The name for the keys will be: ssh-keys

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

ASR#
ASR(config)#transport-map type persistent ssh
persistent-ssh-map
ASR(config-tmap)#rsa keypair-name ssh-keys
ASR(config-tmap)#transport interface GigabitEthernet0
ASR(config-tmap)#banner wait X
Enter TEXT message.  End with the character 'X'.
--Waiting for vty line--
X
ASR(config-tmap)#
ASR(config-tmap)# banner diagnostic X
Enter TEXT message.  End with the character 'X'.
--Welcome to Diagnostic Mode--
c
ASR(config-tmap)#connection wait allow interruptible
ASR(config-tmap)#exit
ASR(config)#transport type persistent ssh input persistent-ssh
*Jul 10 15:31:57.102: %UICFGEXP-6-SERVER_NOTIFIED_START: R0/0: psd:
Server persistent ssh has been notified to start
```

Now you must enable local authentication for persistent SSH. This can be done either with the **aaa new-model** command or without it. Both of the scenarios are described here. (In either case, ensure that you have a local username/password account on the router).

You can choose which configuration based on whether you have AAA enabled on the ASR.

1. With AAA enabled:

   ```
   ASR(config)#aaa new-model
   ASR(config)#aaa authentication login default local
   ASR(config)#line vty 0 4
   ASR(config-line)#login authentication default
   ```
2. Without AAA enabled:

   ```
   ASR(config)#line vty 0 4
   ASR(config-line)#login local
   ```

Verify

SSH to the ASR with the IP address of the VRF-enabled **Gigabitethernet0** interface. Once the password is entered, you must enter the break sequence (**Ctrl-C** or **Ctrl-Shift-6**).

```
management-station$ ssh -l cisco 10.106.47.139
cisco@10.106.47.139's password:

--Waiting for vty line--

--Welcome to Diagnostic Mode--
ASR(diag)#
```

Note: Enter the break sequence (**Ctrl-C** or **Ctrl-Shift-6**) when **--Waiting for vty line--** displays on the terminal in order to enter diagnostic mode.

Persistent Telnet

Configure

With similar logic as described in the previous section for SSH, create a transport map for persistent Telnet as shown here:

```
ASR(config)#transport-map type persistent telnet persistent-telnet
ASR(config-tmap)#banner diagnostic X
Enter TEXT message.  End with the character 'X'.
--Welcome to Diagnostic Mode--
X
ASR(config-tmap)#banner wait X
Enter TEXT message.  End with the character 'X'.
--Waiting for IOS Process--
X
ASR(config-tmap)#connection wait allow interruptible
ASR(config-tmap)#transport interface gigabitEthernet 0
ASR(config-tmap)#exit
ASR(config)#transport type persistent telnet input persistent-telnet
*Jul 10 15:26:56.441: %UICFGEXP-6-SERVER_NOTIFIED_START: R0/0: psd:
Server persistent telnet has been notified to start
```

As discussed in the last section for SSH, there are two ways to configure local authentication as shown here:

1. With AAA enabled:

   ```
   ASR(config)#aaa new-model
   ASR(config)#aaa authentication login default local
   ASR(config)#line vty 0 4
   ASR(config-line)#login authentication default
   ```
2. Without AAA:

   ```
   ASR(config)#line vty 0 4
   ASR(config-line)#login local
   ```

Verify

Telnet to the IP address of **GigabitEthernet0** interface. After you enter the credentials, enter the break sequence, and wait for a few seconds (sometimes it might take a while) before you log into diagnostic mode.

```
Management-station$ telnet 10.106.47.139
Trying 10.106.47.139...
Connected to 10.106.47.139.
Escape character is '^]'.
Username: cisco
Password:

--Waiting for IOS Process--
```

```
--Welcome to Diagnostic Mode--
ASR(diag)#
```

Note: Enter the break sequence **Ctrl+C** or **Ctrl+Shift+6**, and wait for a few seconds. When **--Waiting for IOS Process--** displays on the terminal, you are able to enter diagnostic mode.

Persistent HTTP

In order to enable persistent HTTP access to-the-box (HTTP from-the-box or HTTP client service is not available) and use the new web-based GUI access, use this configuration that utilizes local authentication (you can also use an external AAA server).

Configure

In these configurations, **http-webui** and **https-webui** are the names of the transport-maps.

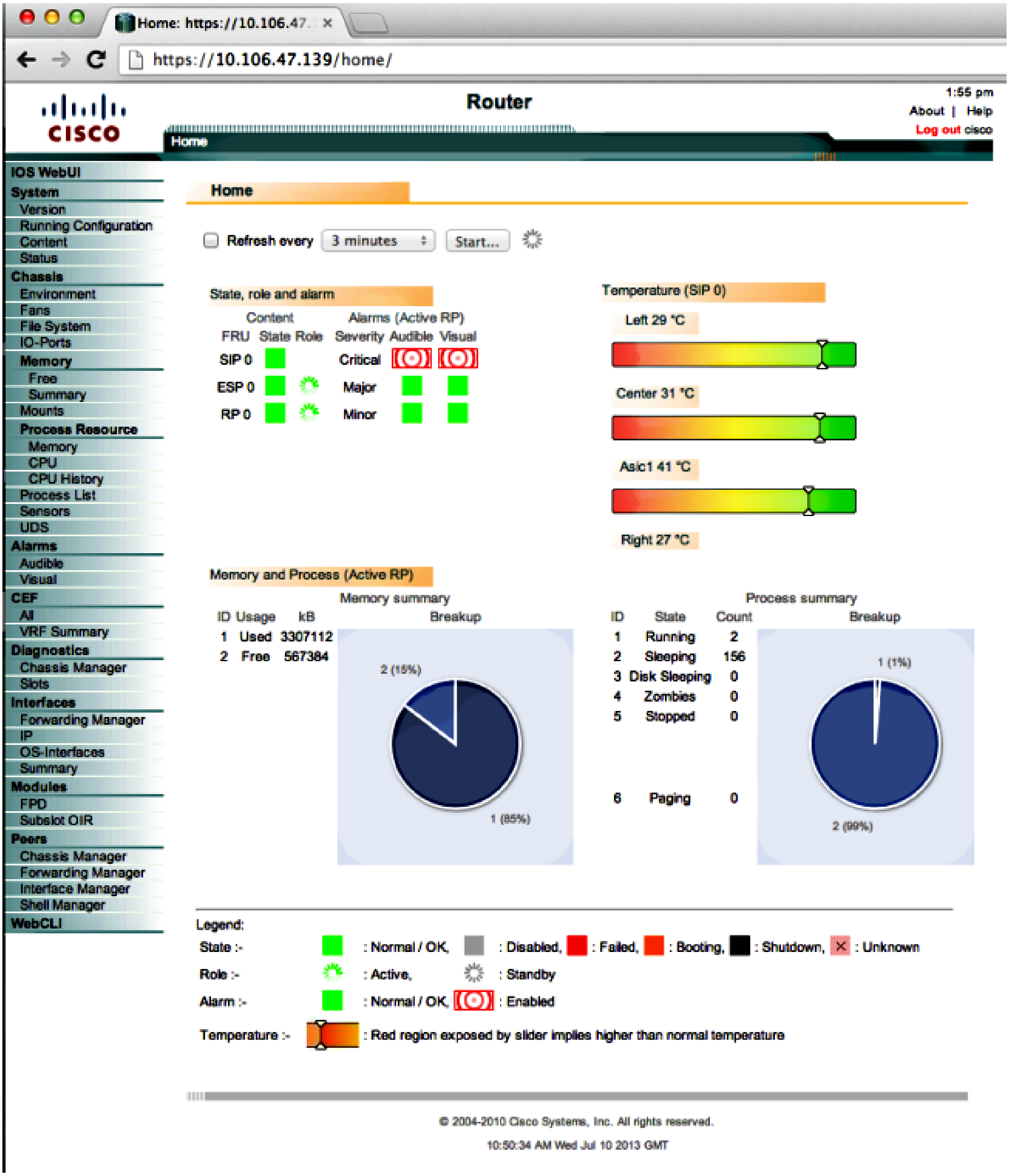
```
ASR(config)#ip http serverASR(config)#ip http authentication local
ASR(config)#username <> password <>
ASR(config)#transport-map type persistent webui http-webui
ASR(config-tmap)#server
ASR(config-tmap)#exit
ASR(config)#transport type persistent webui input http-webui
```

Here is the configuration used in order to enable HTTP secure server (HTTPS).

```
ASR(config)#ip http secure-serverASR(config)#ip http authentication local
ASR(config)#username <> password <>
ASR(config)#transport-map type persistent webui https-webui
ASR(config-tmap)#secure-server
ASR(config-tmap)#exit
ASR(config)#transport type persistent webui input https-webui
```

Verify

Browse to the IP address of an interface on the ASR. Log in with the username/password you created in order to launch the home page. Health and monitoring related information displays, along with a **IOS WebUI** where you can apply commands. Here is a screenshot of the homepage:



☐ Refresh every 3 minutes

State, role and alarm

	Content		Alarms (Active RP)		
FRU	State	Role	Severity	Audible	Visual
SIP 0	🟩		Critical	🟥	🟥
ESP 0	🟩	✳	Major	🟩	🟩
RP 0	🟩	✳	Minor	🟩	🟩

Temperature (SIP 0)

Left 29 °C

Center 31 °C

Asic1 41 °C

Right 27 °C

Memory and Process (Active RP)

Memory summary

ID	Usage	kB
1	Used	3307112
2	Free	567384

Process summary

ID	State	Count
1	Running	2
2	Sleeping	156
3	Disk Sleeping	0
4	Zombies	0
5	Stopped	0
6	Paging	0

Legend:

State :- 🟩 : Normal / OK, ⬜ : Disabled, 🟥 : Failed, 🟧 : Booting, ⬛ : Shutdown, ☒ : Unknown

Role :- ✳ : Active, ✳ : Standby

Alarm :- 🟩 : Normal / OK, 🟥 : Enabled

Temperature :- 🟧 : Red region exposed by slider implies higher than normal temperature

© 2004-2010 Cisco Systems, Inc. All rights reserved.

10:50:34 AM Wed Jul 10 2013 GMT

Troubleshoot

If the WebUI is not available via HTTPS, then verify that the certificate and Rivest-Shamir-Adleman (RSA) key are present and operational. You can use this **debug** command in order to determine the reason that the WebUI does not start properly:

```
ASR#debug platform software configuration notify webui
ASR#config t
ASR(config)#no transport type persistent webui input https-webui
%UICFGEXP-6-SERVER_NOTIFIED_STOP: SIP0: psd: Server wui has been notified to stop
ASR(config)#transport type persistent webui input https-webui

 CNOTIFY-UI: Setting transport map
 CNOTIFY-UI: Transport map https-webui input being processed
 CNOTIFY-UI: Processing map association
 CNOTIFY-UI: Attempting to send config
 CNOTIFY-UI: Preparing to send config
 CNOTIFY-UI: server cache: false, tm: false
 CNOTIFY-UI: secure-server cache: true, tm: true
 CNOTIFY-UI: Validating server config
 CNOTIFY-UI: Validating secure server config
 CNOTIFY-UI: Checking if secure server config is ok
 CNOTIFY-UI: Secure server is enabled in map
 CNOTIFY-UI: Getting trust point
 CNOTIFY-UI: Getting self-signed trust point
 CNOTIFY-UI: Could not get self-signed trustpoint
 CNOTIFY-UI: A certificate for does not exist
 CNOTIFY-UI: Getting rsa key-pair name
 CNOTIFY-UI: Failed to get rsa key pair name
 CNOTIFY-UI: Key needed to generate the pem file
 CNOTIFY-UI: Secure-server config invalid
 CNOTIFY-UI: Config analysis indicates no change
 CNOTIFY-UI: Failed to prepare config
```

RSA Key

In order to verify the presence of the RSA key, enter this command:

```
ASR#show crypto key mypubkey rsa
% Key pair was generated at: XX:XX:XX XXX XXX XX XXXX
Key name: ASR.ASR
Key type: RSA KEYS
 Storage Device: not specified
 Usage: General Purpose Key
 Key is not exportable. Redundancy enabled.
 Key Data&colon;
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXX
% Key pair was generated at: XX:XX:XX XXX XXX XX XXXX
Key name: ASR.ASR.server
Key type: RSA KEYS
Temporary key
 Usage: Encryption Key
 Key is not exportable. Redundancy enabled.
 Key Data&colon;
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
  XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXX
ASR#
```

Take note of the key name, as it is required in order to create the certificate. If a key is not present, you can create one with these commands:

```
ASR(config)#ip domain-name Router
ASR(config)#crypto key generate rsa
The name for the keys will be: Router.Router
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

ASR(config)#
*Dec 22 10:57:11.453: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

Certificate

Once the key is present, you can enter this command in order to verify the certificate:

```
ASR#show crypto pki certificates
 ASR Self-Signed Certificate
  Status: Available
  Certificate Serial Number (hex): 01
  Certificate Usage: General Purpose
  Issuer:
   serialNumber=XXXXXXXXXXX+ipaddress=XXX.XXX.XXX.XXX+hostname=ASR
   cn=XXX.XXX.XXX.XXX
   c=US
   st=NC
   l=Raleigh
  Subject:
   Name: Router
   IP Address: XXX.XXX.XXX.XXX
   Serial Number: XXXXXXXXXXX
   serialNumber=XXXXXXXXXXX+ipaddress=XXX.XXX.XXX.XXX+hostname=aSR
   cn=XXX.XXX.XXX.XXX
   c=US
   st=NC
   l=Raleigh
  Validity Date:
   start date: XX:XX:XX XXX XXX XX XXXX
   end   date: XX:XX:XX XXX XXX XX XXXX
  Associated Trustpoints: local
```

If the certificate is invalid or is not present, then you can create the certificate with these commands:

```
ASR(config)#crypto pki trustpoint local
ASR(ca-trustpoint)#enrollment selfsigned
ASR(ca-trustpoint)#subject-name CN=XXX.XXX.XXX.XXX; C=US; ST=NC; L=Raleigh
ASR(ca-trustpoint)#rsakeypair ASR.ASR 2048
ASR(ca-trustpoint)#crypto pki enroll local
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: XXX.XXX.XXX.XXX
Generate Self Signed Router Certificate? [yes/no]: yes

Router Self Signed Certificate successfully created
```

Once the RSA key and certificate are updated and are valid, the certificate can be associated with the HTTPS configuration:

```
ASR(config)#ip http secure-trustpoint local
```

You can then disable and re-enable the WebUI in order to ensure that it is functional:

```
ASR#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ASR(config)#no transport type persistent webui input https-webui
ASR(config)#
```

```
 CNOTIFY-UI: Setting transport map
 CNOTIFY-UI: Transport map usage being disabled
 CNOTIFY-UI: Processing map association
 CNOTIFY-UI: Attempting to send config
 CNOTIFY-UI: Preparing to send config
 CNOTIFY-UI: Persistent webui will be shutdown if running
 CNOTIFY-UI: Creating config message
 CNOTIFY-UI: Secure-server state actually being set to: disabled
 CNOTIFY-UI: Webui server information: changed: true, status: disabled, port: 80
 CNOTIFY-UI: Webui secure server information: changed: true, status: disabled, port: 443
 CNOTIFY-UI: Webui service (re)start: false. Sending all config
ASR(config)#
ASR(config)#transport type persistent webui input https-webui
ASR(config)#
 CNOTIFY-UI: Setting transport map
 CNOTIFY-UI: Transport map https-webui input being processed
 CNOTIFY-UI: Processing map association
 CNOTIFY-UI: Attempting to send config
 CNOTIFY-UI: Preparing to send config
 CNOTIFY-UI: server cache: false, tm: false
 CNOTIFY-UI: secure-server cache: true, tm: true
 CNOTIFY-UI: Validating server config
 CNOTIFY-UI: Validating secure server config
 CNOTIFY-UI: Checking if secure server config is ok
 CNOTIFY-UI: Secure server is enabled in map
 CNOTIFY-UI: Getting trust point
 CNOTIFY-UI: Using issued certificate for identification
 CNOTIFY-UI: Getting rsa key-pair name
 CNOTIFY-UI: Getting private key
 CNOTIFY-UI: Getting certificate
 CNOTIFY-UI: Secure server config is ok
 CNOTIFY-UI: Secure-server config is valid
 CNOTIFY-UI: Creating config message
 CNOTIFY-UI: Secure-server state actually being set to: enabled
 CNOTIFY-UI: Adding rsa key pair
 CNOTIFY-UI: Getting base64 encoded rsa key
 CNOTIFY-UI: Getting rsa key-pair name
 CNOTIFY-UI: Getting private key
 CNOTIFY-UI: Added rsa key
 CNOTIFY-UI: Adding certificate
 CNOTIFY-UI: Getting base64 encoded certificate
 CNOTIFY-UI: Getting certificate
 CNOTIFY-UI: Getting certificate for local
 CNOTIFY-UI: Certificate added
 CNOTIFY-UI: Webui server information: changed: false, status: disabled, port: 80
 CNOTIFY-UI: Webui secure server information: changed: true, status: enabled, port: 443
 CNOTIFY-UI: Webui service (re)start: true. Sending all config

%UICFGEXP-6-SERVER_NOTIFIED_START: SIP0: psd:  Server wui has been notified to start
```

Related Information

- **Console Port, Telnet, and SSH Handling**
- **Understanding the Diagnostic Mode**
- **Technical Support & Documentation - Cisco Systems**

Updated: Jan 15, 2016 Document ID: 116093