Implementing Host Services and Applications

Cisco IOS XR software Host Services and Applications features on the router are used primarily for checking network connectivity and the route a packet follows to reach a destination, mapping a hostname to an IP address or an IP address to a hostname, and transferring files between routers and UNIX workstations.

Prerequisites for implementing Host Services and Applications

Ensure to install the relevant optional RPM package before using the host services or applications.

HTTP Client application

HTTP Client allows files to be transferred from an HTTP server to another device over a network using the HTTP protocol.

You can configure the HTTP Client and its associated parameters by using the **http client** command.

This table lists the commands used to configure HTTP Client settings:

Table 1: Commands used to configure HTTP Client settings

Features	Description
connection	Configure HTTP Client connection by using either retry or timeout options.
response	How long HTTP Client waits for a response from the server for a request message before giving up.
secure-verify-host	Verify host in peer's certificate. To disable verifying this, you can use the command **http client secure-verify-host disable**
secure-verify-peer	Verify authenticity of the peer's certificate.
source-interface	Specifies the source interface for all outgoing HTTP connections. You can enter an IPv4 address, an IPv6 address, or both addresses.

Features	Description
ssl version	SSL version (configuration) to be used for HTTPS requests.
tcp-window-scale scale	Sets the TCP window-scale factor for high-latency links.
version version	HTTP version to be used in HTTP requests. • 1.0—HTTP1.0 will be used for all HTTP requests. • 1.1—HTTP1.1 will be used for all HTTP requests. • default libcurl—will use HTTP version automatically.
vrf name	Name of the VRF.

Configure HTTP Client application

The HTTP Client application is available by default. You can configure HTTP client settings, or you can view and modify existing settings.

To configure the settings, use the **http client** command in XR config mode.

Use this task to configure the settings.

Procedure

Step 1 Enter the XR config mode.

Step 2 Run the **http client** command in XR config mode.

Example:

```
Router #configure
Router(config)#http client ?
connection          Configure HTTP Client connection
response            How long HTTP Client waits for a response from the server
                    for a request message before giving up
secure-verify-host  Verify that if server certificate is for the server it is known as
secure-verify-peer  Verify authenticity of the peer's certificate
source-interface    Specify interface for source address
ssl                 SSL configuration to be used for HTTPS requests
tcp-window-scale    Set tcp window-scale factor for High Latency links
version             HTTP Version to be used in HTTP requests
vrf                 Name of vrf
```

Example:

This example shows how to set the TCP window-scale to 8.

```
Router(config)#http client tcp-window-scale 8
```

This example shows how to set the HTTP version to 1.0.

```
Router(config)#http client version 1.0
```

Note
The HTTP Client uses libcurl version 7.30.

Transmission Control Protocol

Transmission Control Protocol (TCP) is a connection-oriented protocol that defines the format of data and acknowledgments exchanged between two computer systems to facilitate data transfer.

- TCP outlines procedures to ensure that the data being transferred arrives correctly at the intended destination.
- TCP enables multiple applications on a system to communicate simultaneously. It manages the demultiplexing of incoming traffic among application programs.

TCP dump file converter

The TCP dump file converter is a tool that converts IOS-XR dump files from binary format into user-friendly formats such as PCAP or text.

Key features

The key features of the file converter include:

- The converter is especially useful when Non-Stop Routing (NSR) is disabled or a session flap occurs on your system. In these cases, the TCP process on the NCS system automatically stores the latest 200 packet traces in binary format in a temporary folder.
- TCP dump packet traces also include data about the configured routing protocols and overall network traffic on your system. This data provides insights to help you identify and resolve network infrastructure issues, enabling proactive troubleshooting.

Binary files

You can view packet traces binary files in the user-readable format using these methods:

- Use the **show tcp dump-file** *binary-filename* command to view each binary file in text format manually. For more information, refer to View binary files in text format manually, on page 4

Note This process consumes much time, as you have to view each file manually one after another.

- Convert all stored packet traces in binary files into PCAP, text, or both using the **tcp dump-file convert** command.

For more information, refer to Convert binary files to readable format using TCP dump file converter , on page 5. This active approach greatly improves the efficiency and ease of packet analysis during network troubleshooting.

Limitations and restrictions for TCP dump file converter

The TCP dump file converter has these limitations and restrictions:

- The system only stores the most recent 200 message exchanges that occurred right before the session termination, when NSR is disabled, or during a session flap.
- You can view only one binary file in text format using the **show tcp dump-file** *binary- filename* command.
- TCP dump files are generated by default for BGP, MSDP, MPLS LDP, and SSH.

View binary files in text format manually

Use this task to view each packet trace binary file in text format without using the TCP dump file converter.

Procedure

Step 1 Run the **show tcp dump-file list all** command to view the list of packet traces in binary files stored in the tcpdump folder.

Example:

```
RP/0/RP0/CPU0:ios# show tcp dump-file list all
                        total 1176
                        -rw-r--r-- 1 root root 5927 Nov 22 12:42 31_0_0_126.179.20966.cl.1700656933
                        -rw-r--r-- 1 root root 5892 Nov 22 12:42 31_0_0_127.179.35234.cl.1700656933
                        -rw-r--r-- 1 root root 6148 Nov 22 12:42 31_0_0_149.179.54939.cl.1700656933
                        -rw-r--r-- 1 root root 5894 Nov 22 12:42 31_0_0_155.179.18134.cl.1700656933
                        -rw-r--r-- 1 root root 6063 Nov 22 12:42 31_0_0_156.179.25445.cl.1700656933
                        -rw-r--r-- 1 root root 5860 Nov 22 12:42 31_0_0_161.179.30859.cl.1700656933
                        -rw-r--r-- 1 root root 5832 Nov 22 12:42 31_0_0_173.179.36935.cl.1700656933
                        -rw-r--r-- 1 root root 5906 Nov 22 12:42 31_0_0_190.179.25642.cl.1700656933
```

Step 2 Run the the **show tcp dump-file** *binary- filename* command to view each packet traces binary file in text format.

Example:

```
RP/0/RP0/CPU0:ios# show tcp dump-file 10_106_0_73.179.34849.cl.1707424077 location 0/RP0/CPU0
                        Filename: 10_106_0_73.179.34849.cl.1707424077

                        ==============================================================
                        Connection state is CLOSED, I/O status: 0, socket status: 103
                        PCB 0x00007f86bc05e3b8, SO 0x7f86bc05e648, TCPCB 0x7f86bc0c3718, vrfid
0x60000000,
                        Pak Prio: Medium, TOS: 192, TTL: 1, Hash index: 1593
                        Local host: 10.106.0.72, Local port: 179 (Local App PID: 11354)
                        Foreign host: 10.106.0.73, Foreign port: 34849
                        (Local App PID/instance/SPL_APP_ID: 11354/1/0)

                        Current send queue size in bytes: 0 (max 0)
                        Current receive queue size in bytes: 0 (max 0)  mis-ordered: 0 bytes
                        Current receive queue size in packets: 0 (max 0)
                        Timer          Starts    Wakeups         Next(msec)
                        Retrans        103448          8                0
                        SendWnd             0          0                0
                        TimeWait            1          0                0
                        AckHold        106815     106545                0
                        KeepAlive           1          0                0
                        PmtuAger            0          0                0
                        GiveUp              0          0                0
                        Throttle            0          0                0
                        FirstSyn            0          0                0

                        iss: 161240548   snduna: 163206936   sndnxt: 163206936
                        sndmax: 163206936   sndwnd: 63104       sndcwnd: 18120
                        irs: 3691232436  rcvnxt: 3693473072  rcvwnd: 26099   rcvadv: 3693499171
```

This sample shows only a portion of the full output; the complete output provides additional details.

Convert binary files to readable format using TCP dump file converter

Use this task to convert the dump packet traces in binary files into PCAP and text formats.

Procedure

Step 1 Run the **tcp dump-file convert all-formats all** command to convert the dump packet traces in binary files into PCAP and text formats.

Example:

```
RP/0/RP0/CPU0:ios# tcp dump-file convert all-formats all
                        ascii file is saved at :
/harddisk:/decoded_dumpfiles/text_tcpdump_peer_all_node0_RP0_CPU0_2024_3_19_10_8_53.462070.txt
                        pcap file is saved at :
/harddisk:/decoded_dumpfiles/pcap_tcpdump_peer_all_node0_RP0_CPU0_2024_3_19_10_8_40.154838.pcap
                        [OK]
```

By default, the system stores the converted files in the "decoded_dumpfiles" folder on the "hard disk".

Use the **location node-id** and **file** *file path* keywords to save the converted TCP dump file to your desired location.

For example, **tcp dump-file convert all-formats all location** *0/RP0/CPU0* **file** */harddisk:/demo2* .

```
RP/0/RP0/CPU0:ios# tcp dump-file convert all-formats all location 0/RP0/CPU0 file /harddisk:/demo2
                        ascii file is saved at : /harddisk:/demo2.txt
                        pcap file is saved at : /harddisk:/demo2.pcap
                        [OK]
```

Step 2 Run the **run cat text** *file path* command to view the converted text file in the CLI.

Example:

```
RP/0/RP0/CPU0:ios# run cat
/harddisk:/decoded_dumpfiles/text_tcpdump_peer_all_node0_RP0_CPU0_2024_3_19_10_8_53.462070.txt
                        Filename: 2024_3_19_10_8_53.462070

                        ==============================================================
                        Connection state is CLOSED, I/O status: 0, socket status: 103
                        PCB 0x0000000000f47a80, SO 0xf476d0, TCPCB 0xf6a370, vrfid 0x60000000,
                        Pak Prio: Medium, TOS: 192, TTL: 255, Hash index: 563
                        Local host: 14:11:11::1, Local port: 47743 (Local App PID: 19579)
```

```
Foreign host: 14:11:11::2, Foreign port: 179
(Local App PID/instance/SPL_APP_ID: 19579/1/0)

Current send queue size in bytes: 0 (max 0)
Current receive queue size in bytes: 0 (max 0)  mis-ordered: 0 bytes
Current receive queue size in packets: 0 (max 0)

Timer          Starts    Wakeups         Next(msec)
Retrans            70          2                0
SendWnd             0          0                0
TimeWait            2          0                0
AckHold            66         61                0
KeepAlive           1          0                0
PmtuAger            0          0                0
GiveUp              0          0                0
Throttle            0          0                0
FirstSyn            1          1                0

iss: 3113104891  snduna: 3113106213  sndnxt: 3113106213
sndmax: 3113106213  sndwnd: 31523       sndcwnd: 2832
irs: 4250126727  rcvnxt: 4250128049  rcvwnd: 31448   rcvadv: 4250159497
```

This sample shows only a portion of the full output; the complete output provides additional details.

Step 3 Run the **scp** command to copy the converted packet traces from the system to your local computer and view the converted PCAP file.