SPAN Session and Packet Capture APIs

Table 1: SPAN Session APIs

Action	Method	Payload Required	API
To create a SPAN session	POST	Yes	/api/config/monitor
To get the SPAN monitor session status	GET	No	/api/operational/monitor\?deep
To get the SPAN session configuration details	GET	No	/api/config/monitor\?deep
to get the SPAN session operational status	GET	No	• /api/operational/system/monitor/session • /api/operational/system/monitor/session\?deep

Example for a SPAN Session Payload

```
<session>
    <number>20</number>
    <destination>
        <vm-vnic>
            <vm-name>Linux2</vm-name>
            <vnic-id>0</vnic-id>
        </vm-vnic>
    </destination>
    <source>
    <interfaces>
        <vm-vnic>
            <vm-name>Linux1</vm-name>
            <vnic-id>0</vnic-id>
            <direction>both</direction>
       </vm-vnic>
        <interface>
            <name>GE0-0</name>
            <direction>both</direction>
        </interface>
```

```
    </interfaces>
    </source>
</session>
```

Table 2: SPAN Session Payload Description

Property	Type	Description	Mandatory/Default Value
number	Integer	SPAN session number	Yes
destination	String	Destination for the mirrored traffic	Yes
vm-name	String	Name of the VM	Yes
vnic-id	String	Virtual network interface controller ID	Yes In the case of virtio net or SRIOV VF, you have to specify the NIC ID of the VM interface.
source	String	Source the mirrored traffic	Yes
direction	String	Direction of the traffic	Yes
interface	String	Source or destination interface.	Yes

- Example: POST SPAN Session API, on page 2
- Example: GET SPAN Session APIs, on page 3
- Packet Capture APIs, on page 6

Example: POST SPAN Session API

```
curl -v -u admin:XXXX -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -k -X
POST https://209.165.201.1/api/config/monitor -d '<session><number>20</number>
<destination><vm-vnic><vm-name>Linux2</vm-name><vnic-id>0</vnic-id></vm-vnic></destination>
<source><interfaces><vm-vnic><vm-name>Linux1</vm-name><vnic-id>0</vnic-id><direction>both</direction></vm-vnic>
<interface><name>GE0-0</name><direction>both</direction></interface></interfaces></source></session>'

Note: Unnecessary use of -X or --request, POST is already inferred.
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
```

```
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 13 23:55:53 2017 GMT
*  expire date: Mar 11 23:55:53 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /api/config/monitor HTTP/1.1
> Authorization: Basic YWRtaW46TXlUZXN0MTIzIw==
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
> Content-Length: 330
>
* upload completely sent off: 330 out of 330 bytes
< HTTP/1.1 201 Created
< Server: nginx/1.10.1
< Date: Wed, 15 Mar 2017 02:42:25 GMT
< Content-Type: text/html
< Content-Length: 0
< Connection: keep-alive
< Last-Modified: Wed, 15 Mar 2017 02:42:25 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1489-545745-460682
< Pragma: no-cache
<
sj22lab-as2:145>
```

Example: GET SPAN Session APIs

Use this operational API to get the SPAN monitor session status.

```
curl -v -u admin:XXXXX -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -k -X
GET https://209.165.201.1/api/operational/monitor\?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 13 23:55:53 2017 GMT
*  expire date: Mar 11 23:55:53 2027 GMT
```

```
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/operational/monitor?deep HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46TXlUZXN0MTIzIw==
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 15 Mar 2017 04:43:15 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache

<

<monitor xmlns="http://www.cisco.com/nfv/span_session" xmlns:y="http://tail-f.com/ns/rest"
  xmlns:span-session="http://www.cisco.com/nfv/span_session">
  <session>
    <number>20</number>
    <source>
      <interfaces>
        <vm-vnic>
          <vm-name>Linux1</vm-name>
          <vnic-id>0</vnic-id>
          <direction>both</direction>
        </vm-vnic>
        <interface>
          <name>GE0-0</name>
          <direction>both</direction>
        </interface>
      </interfaces>
    </source>
    <destination>
      <vm-vnic>
        <vm-name>Linux2</vm-name>
        <vnic-id>0</vnic-id>
      </vm-vnic>
    </destination>
    <status>CREATE_SUCCESS</status>
  </session>
</monitor>
```

Use this GET API to get the SPAN session configuration details.

```
curl -v -u admin:XXXXX -H Accept:application/vnd.yang.data+xml -H
Content-Type:application/vnd.yang.data+xml -k -X
GET https://209.165.201.1/api/config/monitor\?deep
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 209.165.201.1...
* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* NPN, negotiated HTTP1.1
```

```
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Unknown (67):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* Server certificate:
*  subject: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  start date: Mar 13 23:55:53 2017 GMT
*  expire date: Mar 11 23:55:53 2027 GMT
*  issuer: CN=Cisco-Enterprise-NFVIS-Self-Signed-Certificate
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> GET /api/config/monitor?deep HTTP/1.1
> Host: 209.165.201.1
> Authorization: Basic YWRtaW46TXlUZXN0MTIzIw==
> User-Agent: curl/7.50.1
> Accept:application/vnd.yang.data+xml
> Content-Type:application/vnd.yang.data+xml
>
< HTTP/1.1 200 OK
< Server: nginx/1.10.1
< Date: Wed, 15 Mar 2017 04:39:29 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
< Connection: keep-alive
< Last-Modified: Wed, 15 Mar 2017 02:42:25 GMT
< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
< Etag: 1489-545745-460682
< Pragma: no-cache
<

<monitor xmlns="http://www.cisco.com/nfv/span_session" xmlns:y="http://tail-f.com/ns/rest"
  xmlns:span-session="http://www.cisco.com/nfv/span_session">
  <session>
    <number>20</number>
    <source>
      <interfaces>
        <vm-vnic>
          <vm-name>Linux1</vm-name>
          <vnic-id>0</vnic-id>
          <direction>both</direction>
        </vm-vnic>
        <interface>
          <name>GE0-0</name>
          <direction>both</direction>
        </interface>
      </interfaces>
    </source>
    <destination>
      <vm-vnic>
        <vm-name>Linux2</vm-name>
        <vnic-id>0</vnic-id>
      </vm-vnic>
    </destination>
  </session>
</monitor>
```

Packet Capture APIs

Table 3: Packet Capture APIs

Action	Method	Payload Required	API
To configure packet capture on a physical or virtual network interface controller	POST	Yes	api/operations/packet-capture/tcpdump

Example for the Packet Capture Payload for a Physical Port

```
<input>
    <port>eth0</port>
    <time>10</time><
/input>
```

Example for the Packet Capture Payload for a vNIC

```
<input>
    <vnic>
        <tenant-name>admin</tenant-name>
        <deployment-name>1489084431</deployment-name>
        <vm-name>ROUTER</vm-name>
        <vnic-id>0</vnic-id>
    </vnic>
    <time>10</time>
</input>
```

Table 4: Packet Capture Payload Description

Property	Type	Description	Mandatory/Default Value
port	String	Physical or virtual network interface controller	Yes
time	String	Time period over which packets are captured. The default value is 60 seconds.	Yes
tenant-name	String	Name of the tenant	Yes
deployment-name	String	Name of the VM deployment	Yes
vm-name	String	Name of the VM	Yes
vnic-id	Integer	Virtual network interface controller ID	Yes

Example: POST Packet Capture APIs

Use this POST API to configure packet capture on a physical port.

```
curl -v -k -u admin:Cisco123# -H "Content-Type: application/vnd.yang.data+xml"  -H "Accept:
 application/vnd.yang.data+xml" -X
POST https://209.165.201.1/api/operations/packet-capture/tcpdump  -d
'<input><port>eth0</port><time>10</time></input>'
*   Trying 209.165.201.1...

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)

* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate

* Server auth using Basic with user 'admin'

> POST /api/operations/packet-capture/tcpdump HTTP/1.1

> Host: 209.165.201.1

> Authorization: Basic YWRtaW46Q2lzY28xMjMj

> User-Agent: curl/7.43.0

> Content-Type: application/vnd.yang.data+xml

> Accept: application/vnd.yang.data+xml

> Content-Length: 47

>

* upload completely sent off: 47 out of 47 bytes

< HTTP/1.1 200 OK

< Server: nginx/1.10.1

< Date: Wed, 29 Mar 2017 20:35:50 GMT

< Content-Type: application/vnd.yang.operation+xml

< Content-Length: 151

< Connection: keep-alive

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Vary: Accept-Encoding

< Pragma: no-cache

<

<output xmlns='http://www.cisco.com/nfvos/packet_capture'>

  <pcap-location>/data/intdatastore/pktcaptures/tcpdump_eth0.pcap</pcap-location>

</output>

* Connection #0 to host 209.165.201.1 left intact
```

Use this POST API to configure packet capture on a vNIC.

```
curl -v -k -u admin:Cisco123# -H "Content-Type: application/vnd.yang.data+xml"  -H "Accept:
 application/vnd.yang.data+xml" -X
POST https://209.165.201.1/api/operations/packet-capture/tcpdump  -d
'<input><vnic><tenant-name>admin</tenant-name>
<deployment-name>1489084431</deployment-name><vm-name>ROUTER</vm-name><vnic-id>0</vnic-id></vnic><time>10</time></input>'

*   Trying 209.165.201.1...

* Connected to 209.165.201.1 (209.165.201.1) port 443 (#0)

* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

* Server certificate: Cisco-Enterprise-NFVIS-Self-Signed-Certificate

* Server auth using Basic with user 'admin'

> POST /api/operations/packet-capture/tcpdump HTTP/1.1

> Host: 209.165.201.1

> Authorization: Basic YWRtaW46Q2lzY28xMjMj

> User-Agent: curl/7.43.0

> Content-Type: application/vnd.yang.data+xml

> Accept: application/vnd.yang.data+xml

> Content-Length: 47

>

* upload completely sent off: 47 out of 47 bytes

< HTTP/1.1 200 OK

< Server: nginx/1.10.1

< Date: Wed, 29 Mar 2017 20:35:50 GMT

< Content-Type: application/vnd.yang.operation+xml

< Content-Length: 151

< Connection: keep-alive

< Cache-Control: private, no-cache, must-revalidate, proxy-revalidate

< Vary: Accept-Encoding

< Pragma: no-cache

<

<output
xmlns='/data/intdatastore/pktcaptures/1489084431_ROUTER_vnic0.pcaphttp://www.cisco.com/nfvos/packet_capture'>
<pcap-location>/data/intdatastore/pktcaptures/1489084431_ROUTER_vnic0.pcap</pcap-location>
</output>

* Connection #0 to host 209.165.201.1 left intact
```