



Troubleshooting vPath and vServices

This chapter includes the following sections:

- [VNS Agent, page 3-1](#)
- [Using vPath Ping to Determine Connectivity, page 3-2](#)

See the *Cisco Virtual Security Gateway for Microsoft Hyper-V Troubleshooting Guide, Release 5.2(1)VSG1(4.1)* for more information on troubleshooting VSG.

VNS Agent

Virtual Network Service (VNS) agent-related event logs are maintained on the Virtual Supervisor Module (VSM), not on the Cisco VSG.

This section includes the following topics:

- [Core Module, page 3-1](#)
- [VPath Module, page 3-1](#)
- [License Module, page 3-2](#)

Core Module

Core events are those events that are related to port attach, port detach, Internet Protocol Database (IPDB), and to port-profile CLI.

This example shows how to enable/disable error messages for the vns_agent core module:

```
vsm# event-log vns-agent core-error [terminal] ----->enable messages to the terminal
vsm# no event-log vns-agent core-error [terminal] ----->disable messages to the terminal
```

This example shows how to enable/disable informational messages for the vns_agent core module:

```
vsm# event-log vns-agent core-info [terminal] ----->enable messages to the terminal
vsm# no event-log vns-agent core-info [terminal] ----->disable messages to the terminal
```

VPath Module

Because the vPath module works based on core-module events, you should always enable core module event logs before you enable the vPath module events.

This example shows how to enable/disable error messages for the vns_agent vPath module:

```
vsm# event-log vns-agent vpath-error [terminal] ----->enable messages to the terminal
vsm# no event-log vns-agent vpath-error [terminal] ----->disable messages to the terminal
```

This example shows how to enable/disable informational messages for the vns_agent vPath module:

```
vsm# event-log vns-agent vpath-info [terminal] ----->enable messages to the terminal
vsm# no event-log vns-agent vpath-info [terminal] ----->disable messages to the terminal
```

License Module

Because the license module works based on core-module events, you should always enable the core module event logs before enabling the license module.

This example shows how to enable/disable error messages for the vns_agent license module:

```
vsm# event-log vns-agent license-error [terminal] ----->enable messages to the terminal
vsm# no event-log vns-agent license-error [terminal] ----->disable messages to the terminal
```

This example shows how to enable/disable informational messages for the vns_agent license module:

```
vsm# event-log vns-agent license-info [terminal] ----->enable messages to the terminal
vsm# no event-log vns-agent license-info [terminal] ----->disable messages to the terminal
```

Using vPath Ping to Determine Connectivity

You can use the vpath **ping** command to determine the connectivity between the Cisco VSG and the VEM.

This example shows how to ping the Cisco VSG connections and if they are reachable:

```
VSM-1# ping vsn all src-module all
ping vsn 106.1.1.1 vlan 0 from module 3 5, seq=0 timeout=1-sec
  module(usec) : 3(156) 5(160)
ping vsn 110.1.1.1 vlan 0 from module 3 5, seq=0 timeout=1-sec
  module(failed) : 3(VSN ARP not resolved) 5(VSN ARP not resolved)

ping vsn 106.1.1.1 vlan 0 from module 3 5, seq=1 timeout=1-sec
  module(usec) : 3(230) 5(151)
ping vsn 110.1.1.1 vlan 0 from module 3 5, seq=1 timeout=1-sec
  module(failed) : 3(VSN ARP not resolved) 5(VSN ARP not resolved)

ping vsn 106.1.1.1 vlan 0 from module 3 5, seq=2 timeout=1-sec
  module(usec) : 3(239) 5(131)
ping vsn 110.1.1.1 vlan 0 from module 3 5, seq=2 timeout=1-sec
  module(failed) : 3(VSN ARP not resolved) 5(VSN ARP not resolved)

ping vsn 106.1.1.1 vlan 0 from module 3 5, seq=3 timeout=1-sec
  module(usec) : 3(248) 5(153)
ping vsn 110.1.1.1 vlan 0 from module 3 5, seq=3 timeout=1-sec
  module(failed) : 3(VSN ARP not resolved) 5(VSN ARP not resolved)

ping vsn 106.1.1.1 vlan 0 from module 3 5, seq=4 timeout=1-sec
  module(usec) : 3(259) 5(126)
ping vsn 110.1.1.1 vlan 0 from module 3 5, seq=4 timeout=1-sec
  module(failed) : 3(VSN ARP not resolved) 5(VSN ARP not resolved)
```

This example shows how to display VSN ping options:

```
VSM-1# ping vsn ?
  all  All VSNs associated to VMs
  ip   IP Address
```

This example shows how to display VSN ping options for all source modules:

```
VSM-1# ping vsn all src-module ?
<3-66>      Module number
all         All modules in VSM
vpath-all  All modules having VMs associated to VSNs
```

This example shows how to set up a ping for all source modules from a specified IP address:

```
VSM-1# ping vsn ip 10.1.1.60 src-module all
ping vsn 10.1.1.60 vlan 0 from module 4 5 7, seq=0 timeout=1-sec
  module(usec)      : 4(301) 5(236)
  module(failed)    : 7(VSN ARP not resolved)

ping vsn 10.1.1.60 vlan 0 from module 4 5 7, seq=1 timeout=1-sec
  module(usec)      : 4(241) 5(138) 7(270)

ping vsn 10.1.1.60 vlan 0 from module 4 5 7, seq=2 timeout=1-sec
  module(usec)      : 4(230) 5(155) 7(256)

ping vsn 10.1.1.60 vlan 0 from module 4 5 7, seq=3 timeout=1-sec
  module(usec)      : 4(250) 5(154) 7(284)

ping vsn 10.1.1.60 vlan 0 from module 4 5 7, seq=4 timeout=1-sec
  module(usec)      : 4(231) 5(170) 7(193)
```

This example shows to set up a ping for all vPath source modules for a specified IP address:

```
VSM-1# ping vsn ip 10.1.1.60 src-module vpath-all
ping vsn 10.1.1.60 vlan 0 from module 4 5, seq=0 timeout=1-sec
  module(usec)      : 4(223) 5(247)

ping vsn 10.1.1.60 vlan 0 from module 4 5, seq=1 timeout=1-sec
  module(usec)      : 4(206) 5(167)

ping vsn 10.1.1.60 vlan 0 from module 4 5, seq=2 timeout=1-sec
  module(usec)      : 4(241) 5(169)
```

This example shows how to set up a ping for all source modules of a specified IP address with a time-out and a count:

```
VSM-1# ping vsn ip 10.1.1.60 src-module all timeout 2 count 3
ping vsn 10.1.1.60 vlan 0 from module 4 5 7, seq=0 timeout=2-sec
  module(usec)      : 4(444) 5(238) 7(394)

ping vsn 10.1.1.60 vlan 0 from module 4 5 7, seq=1 timeout=2-sec
  module(usec)      : 4(259) 5(154) 7(225)

ping vsn 10.1.1.60 vlan 0 from module 4 5 7, seq=2 timeout=2-sec
  module(usec)      : 4(227) 5(184) 7(216)
```

