针对初学者的Nexus备忘单故障排除

目录

简介

本文档介绍可用于对Nexus产品进行故障排除的各种工具，您可以使用这些工具来诊断和修复问题。

概述

了解哪些工具可用，以及在什么情况下可以使用它们获得最大收益非常重要。事实上，有时某个工具并非完全可行，因为它设计用于其他方面。

此表汇编了在Nexus平台上排除故障的各种工具及其功能。有关详细信息和CLI示例，请参阅Nexus工具部分。

工具	功能	使用案例示例	优点	缺点	持久性	受影响平面	使用的CLI命令
Ethanalyzer	捕获发往或发自CPU的流量	流量缓慢问题、延迟和拥塞	非常适合处理缓慢、拥塞和延迟问题	通常只看到控制平面流量，速率受限	不适用	控制层面.可用于某些场景中的数据平面（SPAN到CPU）	#ethanalyzer本地口带内 #ethanalyzer local interface [interface ID] display filter [WORD] 示例： #ethanalyzer local interface Ethern display filter ICM
SPAN	捕获并镜像大量数	失败 ping s、无序数据包等	非常适合间歇性流量丢	需要运行嗅探器软件的外部	需要配置并启用/禁	控制+数据	#monitor sessio #description [NA

	据包		失	设备 需要TCAM资源	用SPAN会话		#source interfac [port ID] #destin interface [port ID shut
DMirror	仅捕获发往或发自CPU的Broadcom Nexus设备的流量	流量缓慢问题、延迟和拥塞	非常适合处理缓慢、拥塞和延迟问题	仅适用于Broadcom Nexus设备。速率有限（CloudScale Nexus 9k确实具有SPAN到CPU）	不适用	控制层面.在某些情况下可用于数据平面	因平台而异，请 ELAM概述 — 思
ELAM	捕获进入[或离开(如果Nexus 7K]Nexus交换机的单个数据包	检验数据包是否到达Nexus、检查转发决策、检查数据包是否更改、检验数据包的接口/VLAN等	非常适合处理数据包流和转发问题。非侵入式	需要深入了解硬件。利用特定于架构的独特触发机制。仅在您知道要检查的流量时有用	不适用	控制 +数据	# attach module [MODULE NUM # debug platform internal <>
Nexus 9k Packet Tracer	检测数据包的路径	连接问题和丢包	为流统计信息提供计数器，可用于间歇性/完全丢失。非常适合无TCAM雕刻的线卡	无法捕获ARP流量。仅适用于Nexus 9k	不适用	数据 +控制	# test packet-tra src_IP [SOURC dst_IP [DESTINATION test packet-trace start # test pack tracer stop # tes packet-tracer sh
Traceroute	检测数据包相对于L3跳的路径	ping失败，无法到达主机/目标/互联网等	检测路径中的各种跳以隔离L3故障。	仅标识L3边界破裂的位置（不标识问题本身）	不适用	数据 +控制	# traceroute [目 参数包括： port、port numb source、interfac vrf、source-inte
ping	测试网络中两点之间的连通性	测试设备之间的连通性	用于测试连通性的快速简单工具	仅确定主机是否可达	不适用	数据 +控制	# ping [目标IP] 参数包括： count、packet-s source interface interval、multica loopback、time
PACL/RACL/VACL	捕获特定端口或VLAN的进入/出口流量	主机之间的间歇性数据包丢失，确认数据包是否到达/离开Nexus等	非常适合间歇性流量丢失	需要TCAM资源。对于某些模块，需要手动TCAM雕刻	持久性(应用于running-配置)	数据 +控制	# ip access-list NAME] # ip port access-group [A NAME] # ip acc group [ACL NA 参数包括： deny、fragment no、permit、 remark、show、

							statistics、end、exit、pop、push、where
LogFlash	无论设备重新加载，都会全局存储交换机的历史数据，例如日志帐户、故障文件和事件	设备突然重新加载/关闭，每当重新加载设备时，日志闪存数据都会提供一些有助于分析的信息	信息在设备重新加载时保留（永久存储）	Nexus 7K上的外部=必须在管理引擎平台上安装/集成，才能收集这些日志（con不适用于3K/9K，因为logflash是内部存储设备的分区）	Reload-Persistent	数据+控制	# dir logflash:
OBFL	存储特定模块的历史数据，例如故障和环境信息	设备突然重新加载/关闭，每当重新加载设备时，日志闪存数据都会提供一些有用的信息	信息在设备重新加载时保留（永久存储）	支持有限数量的读取和写入	Reload-Persistent	数据+控制	# show logging onboard module 参数包括：boot-uptime，c boot-history，c first-power-on，counter-stats，device-version，endtime，environmental-history，error-stats，exceptio log，internal，interrupt-stats，history，stat-tra starttime，statu
事件历史记录	需要当前运行的特定进程的信息时	nexus中的每个进程都有自己的事件历史记录，例如CDP、STP、OSPF、EIGRP、BGP、vPC、LACP等	对Nexus上运行的特定进程进行故障排除	设备重新加载后，信息将丢失（非持久性）	非持久性	数据+控制	# show [PROCE internal event-hi [ARGUMENT] 参数包括：邻接关系、cli、、泛洪、ha、he ldp、lsa、msgs objstore、redistribution、r segrt、spf、spf- trigger、statistic
调试	当您需要更精细的实时/实	可以在nexus中的每个进程上进行调试，例如	对在Nexus上运行的特定进	可能影响网络性能	非持久性	数据+控制	# debug process [PROCESS] 示例：

	时信息用于特定流程时	CDP、STP、OSPF、IGRP、BGP、vPC、LACP等	程进行实时故障排除，以实现更精细化				# debug ip ospf
金牌	提供硬件组件（如I/O和管理引擎模块）的启动、运行时间和按需诊断	测试硬件，例如USB、Bootflash、OBFL、ASIC内存、PCIE、端口环回、NVRAM等	只能在6(2)8版及更高版本上检测硬件故障并采取必要的纠正措施	仅检测硬件问题	非持久性	不适用	# show diagnos content module show diagnostic description mod [#] test all
EEM	监视设备上的事件并采取必要的措施	需要某些操作/解决方法/通知的任何设备活动，例如接口关闭、风扇故障、CPU利用率等	支持Python脚本	必须具有网络管理员权限才能配置EEM	EEM脚本和触发器驻留在配置中	不适用	视情况而定，请配置嵌入式事件器

Nexus工具

如果您需要对各种命令及其语法或选项进行详细说明，请参阅 Cisco Nexus 9000系列交换机 — 命令参考 — 思科。

- **Ethanalyzer**

Ethanalyzer是一种NX-OS工具，旨在捕获数据包CPU流量。此工具可以捕获任何命中CPU的入口或出口。它基于广泛使用的开源网络协议分析器Wireshark。有关此工具的更多详细信息，请参阅 Nexus 7000上的Ethanalyzer故障排除指南 — 思科

需要注意的是，一般情况下，Ethanalyzer会捕获进出管理引擎的所有流量，也就是说，它不支持特定于接口的捕获。特定接口增强功能适用于较新代码点的选定平台。此外，Ethanalyzer仅捕获由CPU交换而不是硬件交换的流量。例如，您可以在带内接口、管理接口或前面板端口（如果支持）上捕获流量：

```
Nexus9000_A(config-if-range)# ethanalyzer local interface inband
Capturing on inband
2020-02-18 01:40:55.183177 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/cc:98:91:fc:55:80  Cost = 0  Port = 0x800b
2020-02-18 01:40:55.184031 f8:b7:e2:49:2d:f2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:55.184096 f8:b7:e2:49:2d:f5 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:55.184147 f8:b7:e2:49:2d:f4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
```

```
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:55.184190 f8:b7:e2:49:2d:f3 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:55.493543 dc:f7:19:1b:f9:85 -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/dc:f7:19:1b:f9:80  Cost = 0  Port = 0x8005
2020-02-18 01:40:56.365722      0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID
0xc82a6d3
2020-02-18 01:40:56.469094 f8:b7:e2:49:2d:b4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:57.202658 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/cc:98:91:fc:55:80  Cost = 0  Port = 0x800b
2020-02-18 01:40:57.367890      0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID
0xc82a6d3
10 packets captured

Nexus9000_A(config-if-range)# ethanalyzer local interface mgmt
Capturing on mgmt0
2020-02-18 01:53:07.055100 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:09.061398 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:11.081596 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:13.080874 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:15.087361 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:17.090164 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:19.096518 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:20.391215 00:be:75:5b:d9:00 -> 01:00:0c:cc:cc:cc CDP Device ID:
Nexus9000_A(FDO21512ZES)  Port ID: mgmt0
2020-02-18 01:53:21.119464 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:23.126011 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
10 packets captured

Nexus9000-A# ethanalyzer local interface front-panel eth1/1
Capturing on 'Eth1-1'
1 2022-07-15 19:46:04.698201919 28:ac:9e:ad:5c:b8  01:80:c2:00:00:00 STP 53 RST. Root =
32768/1/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
2 2022-07-15 19:46:04.698242879 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/1/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
3 2022-07-15 19:46:04.698314467 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/10/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
4 2022-07-15 19:46:04.698386112 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/20/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
5 2022-07-15 19:46:04.698481274 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/30/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
6 2022-07-15 19:46:04.698555784 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/40/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
7 2022-07-15 19:46:04.698627624 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/50/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
```

此输出显示很少能通过Ethanalyzer捕获的消息。请注意，默认情况下，Ethanalyzer最多只能捕获10个数据包。但是，您可以使用此命令提示CLI无限期地捕获数据包。使用CTRL+C退出捕获模式。

```
Nexus9000_A(config-if-range)# ethanalyzer local interface inband limit-captured-frames 0
Capturing on inband
2020-02-18 01:43:30.542588 f8:b7:e2:49:2d:f2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
```

```
2020-02-18 01:43:30.542626 f8:b7:e2:49:2d:f5 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:30.542873 f8:b7:e2:49:2d:f4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:30.542892 f8:b7:e2:49:2d:f3 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:31.596841 dc:f7:19:1b:f9:85 -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/dc:f7:19:1b:f9:80  Cost = 0  Port = 0x8005
2020-02-18 01:43:31.661089 f8:b7:e2:49:2d:b2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:31.661114 f8:b7:e2:49:2d:b3 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:31.661324 f8:b7:e2:49:2d:b5 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:31.776638 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/cc:98:91:fc:55:80  Cost = 0  Port = 0x800b
2020-02-18 01:43:33.143814 f8:b7:e2:49:2d:b4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:33.596810 dc:f7:19:1b:f9:85 -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/dc:f7:19:1b:f9:80  Cost = 0  Port = 0x8005
2020-02-18 01:43:33.784099 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/cc:98:91:fc:55:80  Cost = 0  Port = 0x800b
2020-02-18 01:43:33.872280 f8:b7:e2:49:2d:f2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:33.872504 f8:b7:e2:49:2d:f5 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:33.872521 f8:b7:e2:49:2d:f4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
15 packets captured
```

您还可以将过滤器与Ethanalyzer配合使用，以专注于特定流量。有两种类型的过滤器可用于乙醛树脂，它们称为捕获过滤器和显示过滤器。捕获过滤器仅捕获与捕获过滤器中定义的条件匹配的流量。显示过滤器仍会捕获所有流量，但仅显示与显示过滤器中定义的条件匹配的流量。

```
Nexus9000_B# ping 10.82.140.106 source 10.82.140.107 vrf management count 2
PING 10.82.140.106 (10.82.140.106) from 10.82.140.107: 56 data bytes
64 bytes from 10.82.140.106: icmp_seq=0 ttl=254 time=0.924 ms
64 bytes from 10.82.140.106: icmp_seq=1 ttl=254 time=0.558 ms

Nexus9000_A(config-if-range)# ethanalyzer local interface mgmt display-filter icmp
Capturing on mgmt0
2020-02-18 01:58:04.403295 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 01:58:04.403688 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 01:58:04.404122 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 01:58:04.404328 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply

4 packets captured
```

您还可以使用detail选项捕获数据包，并在终端中查看它们，类似于在Wireshark中查看的方式。这样您就可以根据数据包的丢弃结果查看完整的报头信息。例如，如果帧已加密，您将无法看到加密负载。请参阅以下示例：

```
Nexus9000_A(config-if-range)# ethanalyzer local interface mgmt display-filter icmp detail
Capturing on mgmt0
Frame 2 (98 bytes on wire, 98 bytes captured)
    Arrival Time: Feb 18, 2020 02:02:17.569801000
    [Time delta from previous captured frame: 0.075295000 seconds]
    [Time delta from previous displayed frame: 0.075295000 seconds]
    [Time since reference or first frame: 0.075295000 seconds]
    Frame Number: 2
    Frame Length: 98 bytes
```

```
    Capture Length: 98 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:icmp:data]
Ethernet II, Src: 00:be:75:5b:de:00 (00:be:75:5b:de:00), Dst: 00:be:75:5b:d9:00
(00:be:75:5b:d9:00)
    Destination: 00:be:75:5b:d9:00 (00:be:75:5b:d9:00)
        Address: 00:be:75:5b:d9:00 (00:be:75:5b:d9:00)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: IP (0x0800)
>>>>>>>Output Clipped
```

使用Ethanalyzer，您可以：

- 将输出（PCAP文件）写入到各个目标文件系统中的指定文件名：bootflash、logflash、USB等 …… 然后，您可以将保存的文件传输到设备外部，并根据需要在Wireshark中查看该文件。
- 从bootflash中读取文件并在终端上显示。就像直接从CPU接口读取数据一样，如果使用detail关键字，也可以显示完整的数据包信息。

有关各种接口源和输出选项，请参阅以下示例：

```
Nexus9000_A# ethanalyzer local interface mgmt capture-filter "host 10.82.140.107" write
bootflash:TEST.PCAP
Capturing on mgmt0
10
Nexus9000_A# dir bootflash:
       4096    Feb 11 02:59:04 2020  .rpmstore/
       4096    Feb 12 02:57:36 2020  .swtam/
       2783    Feb 17 21:59:49 2020  09b0b204-a292-4f77-b479-1ca1c4359d6f.config
       1738    Feb 17 21:53:50 2020  20200217_215345_poap_4168_init.log
       7169    Mar 01 04:41:55 2019  686114680.bin
       4411    Nov 15 15:07:17 2018  EBC-SC02-M2_303_running_config.txt
   13562165    Oct 26 06:15:35 2019  GBGBLD4SL01DRE0001-CZ07-
        590    Jan 10 14:21:08 2019  MDS20190110082155835.lic
       1164    Feb 18 02:18:15 2020  TEST.PCAP
>>>>>>>Output Clipped

Nexus9000_A# copy bootflash: ftp:
Enter source filename: TEST.PCAP
Enter vrf (If no input, current vrf 'default' is considered): management
Enter hostname for the ftp server: 10.122.153.158
Enter username: calo
Password:
***** Transfer of file Completed Successfully *****
Copy complete, now saving to disk (please wait)...
Copy complete.

Nexus9000_A# ethanalyzer local read bootflash:TEST.PCAP
2020-02-18 02:18:03.140167 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:03.140563 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 02:18:15.663901 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.664303 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 02:18:15.664763 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.664975 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 02:18:15.665338 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.665536 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 02:18:15.665864 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.666066 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply

RTP-SUG-BGW-1# ethanalyzer local interface front-panel eth1-1 write bootflash:e1-1.pcap
Capturing on 'Eth1-1'
10
```

```
RTP-SUG-BGW-1# ethanalyzer local read bootflash:e1-1.pcap detail
Frame 1: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface Eth1-1, id 0
    Interface id: 0 (Eth1-1)
        Interface name: Eth1-1
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul 15, 2022 19:59:50.696219656 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1657915190.696219656 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 53 bytes (424 bits)
    Capture Length: 53 bytes (424 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:llc:stp]
```

- SPAN

SPAN代表SwitchPort Analyzer，用于捕获接口的所有流量并将该流量镜像到目标端口。目标端口通常连接到网络分析器工具（例如运行Wireshark的PC），以便分析通过这些端口的流量。您可以对来自单个或多个端口和VLAN的流量执行SPAN。

SPAN会话包括源端口和目的端口。源端口可以是以太网端口（无子接口）、端口通道、Supervisor带内接口，并且不能同时作为目标端口。此外，对于9300和9500平台等设备，也支持FEX（交换矩阵扩展器）端口。目标端口可以是以太网端口（接入或中继）、端口通道（接入或中继），对于某些设备（如9300上行链路端口）也受支持，而FEX端口不受支持。

可以将多个SPAN会话配置为入口/出口/两者。单个设备可以支持的SPAN会话总数有限制。例如，Nexus 9000最多可支持32个会话，而Nexus 7000只能支持16个会话。您可以在CLI上检查此项或参阅所用产品的SPAN配置指南。

请注意，对于每个NX-OS版本和产品类型，支持的接口类型和功能不同。请参阅您使用的产品和版本的最新配置指南和限制。以下分别是Nexus 9000和Nexus 7000的链接：

Cisco Nexus 9000系列NX-OS系统管理配置指南，版本9.3(x) — 配置SPAN [Cisco Nexus 9000系列交换机] — 思科

Cisco Nexus 7000系列NX-OS系统管理配置指南 — 配置SPAN [Cisco Nexus 7000系列交换机] — 思科

SPAN会话有多种类型。下面列出了一些较常见的类型：

- 本地SPAN：一种源主机和目的主机都位于交换机本地的SPAN会话。换句话说，设置SPAN会话所需的所有配置都应用于单个交换机，即源主机端口和目的主机端口所在的交换机。
- 远程SPAN(RSPAN)：一种源主机和目的主机不在交换机本地的SPAN会话。换句话说，您可以在一台交换机上配置源RSPAN会话，在目标交换机上配置目标RSPAN，并扩展与RSPAN VLAN的连接。

注意：Nexus不支持RSPAN

- 扩展远程SPAN(ERSPAN):交换机使用GRE（通用路由封装）隧道报头封装复制的帧，并将数据包路由到已配置的目的地。在封装和解封交换机（两台不同的设备）上配置源会话和目的会话。 这使我们能够跨第3层网络传输SPAN流量。

- SPAN到CPU：指定给特殊类型的SPAN会话的名称，其中目标端口是管理引擎或CPU。它是本地SPAN会话的一种形式，可用于您无法使用标准SPAN会话的情况。一些常见原因是：没有可用或合适的SPAN目标端口、站点不可访问或不受管理的站点、没有可连接到SPAN目标端口的设备，等等。有关详细信息，请参阅此链接Nexus 9000 Cloud Scale ASIC NX-OS SPAN-to-CPU Procedure - Cisco。必须记住，SPAN到CPU的速率受CoPP（控制平面策略）的限制，因此 sniffing 一个或多个超出监察器的源接口可能导致SPAN到CPU会话发生丢弃。如果发生这种情况，数据并不完全反映线路上的情况，因此SPAN到CPU并不总是适用于高数据速率和/或间歇性丢失的故障排除场景。配置SPAN到CPU会话并管理启用后，您需要运行Ethanalyzer以查看发送到CPU的流量以便执行相应的分析。

以下示例说明如何在Nexus 9000交换机上配置简单的本地SPAN会话：

```
Nexus9000_A(config-monitor)# monitor session ?

*** No matching command found in current mode, matching in (config) mode ***
  <1-32>
  all     All sessions

Nexus9000_A(config)# monitor session 10
Nexus9000_A(config-monitor)#?
  description  Session description (max 32 characters)
  destination  Destination configuration
  filter       Filter configuration
  mtu          Set the MTU size for SPAN packets
  no           Negate a command or set its defaults
  show         Show running system information
  shut         Shut a monitor session
  source       Source configuration
  end          Go to exec mode
  exit         Exit from command interpreter
  pop          Pop mode from stack or restore from name
  push         Push current mode to stack or save it under name
  where        Shows the cli context you are in

Nexus9000_A(config-monitor)# description Monitor_Port_e1/1
Nexus9000_A(config-monitor)# source interface ethernet 1/1
Nexus9000_A(config-monitor)# destination interface ethernet 1/10
Nexus9000_A(config-monitor)# no shut
```

此示例显示已启动的SPAN到CPU会话的配置，然后使用Ethanalyzer捕获流量：

```
N9000-A#show run monitor

monitor session 1
source interface Ethernet1/7 rx
destination interface sup-eth0 << this is what sends the traffic to CPU
no shut

RTP-SUG-BGW-1# ethanalyzer local interface inband mirror limit-c 0
Capturing on 'ps-inb'
2020-02-18 02:18:03.140167 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.663901 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
```

- Dmirror

Dmirror是基于Broadcom的Nexus平台的SPAN到CPU会话的一种类型。其概念与SPAN到CPU相同，速率限制为50 pps（每秒数据包数）。该功能通过bcm-shell CLI实施，用于调试内部数据路径。由于相关的限制，没有NX-OS CLI允许用户配置到管理引擎的SPAN会话，因为它可能会影响控制流量并使用CoPP类。

- **ELAM**

ELAM代表嵌入式逻辑分析器模块。它能够查看ASIC并确定针对单个数据包做出的**转发决**策。因此，使用ELAM，您可以确定数据包是否到达转发引擎，以及哪些端口/VLAN信息。您还可以检查L2 - L4数据包结构，以及是否对数据包进行了任何更改。

了解ELAM依赖于架构，捕获数据包的过程因平台而异（取决于内部架构）非常重要。您必须知道硬件的ASIC映射才能正确应用该工具。对于Nexus 7000，对单个数据包进行两次捕获，一次是在决策之前进行**Data BUS(DBUS)**，另一次是在决策之后进行**Result BUS(RBUS)**。 当您查看DBUS信息时，可以看到数据包接收的内容/位置，以及第2层至第4层信息。在RBUS中的结果可以显示数据包转发到的位置，以及帧是否已更改。您需要为DBUS和RBUS设置触发器，确保它们准备就绪，然后尝试实时捕获数据包。各种线卡的步骤如下：

有关各种ELAM过程的详细信息，请参阅下表中的链接：

ELAM概述	ELAM概述 — 思科
Nexus 7K F1模块	Nexus 7000 F1模块ELAM流程 — 思科
Nexus 7K F2模块	Nexus 7000 F2模块ELAM流程 — 思科
Nexus 7K F3模块	F3 - ELAM示例
Nexus 7K M模块	Nexus 7000 M系列模块ELAM流程 — 思科
Nexus 7K M1/M2和F2模块	适用于M1/M2、F2和Ethanalyzer的Nexus 7K ELAM
Nexus 7K M3模块	Nexus 7000 M3模块ELAM流程 — 思科

适用于Nexus 7000 - M1/M2的ELAM（Eureka平台）

- 使用命令**show module**检查模块编号。
- 通过附加模块x**附加到模块**，其中x是模块编号。
- 使用**show hardware internal dev-port-map**命令检查内部ASIC映射，并检查L2LKP和L3LKP。

```
Nexus7000(config)#show module
Mod  Ports  Module-Type                        Model              Status
---  -----  ---------------------------------- ------------------ ----------
1    0      Supervisor Module-2                N7K-SUP2E          active *
2    0      Supervisor Module-2                N7K-SUP2E          ha-standby
3    48     1/10 Gbps Ethernet Module          N7K-F248XP-25E     ok
4    24     10 Gbps Ethernet Module            N7K-M224XP-23L     ok


Nexus7000(config)# attach module 4
Attaching to module 4 ...
To exit type 'exit', to abort type '$.'
Last login: Fri Feb 14 18:10:21 UTC 2020 from 127.1.1.1 on pts/0

module-4# show hardware internal dev-port-map
--------------------------------------------------------------
CARD_TYPE:       24 port 10G
>Front Panel ports:24
--------------------------------------------------------------
 Device name             Dev role              Abbr num_inst:
--------------------------------------------------------------
> Skytrain               DEV_QUEUEING           QUEUE  4
> Valkyrie               DEV_REWRITE            RWR_0  4
> Eureka                 DEV_LAYER_2_LOOKUP     L2LKP  2
> Lamira                 DEV_LAYER_3_LOOKUP     L3LKP  2
> Garuda                 DEV_ETHERNET_MAC       MAC_0  2
> EDC                    DEV_PHY                PHYS   6
```

```
> Sacramento Xbar ASIC   DEV_SWITCH_FABRIC      SWICHF 1
+-------------------------------------------------------------------------+
+----------------+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++------------+
+-------------------------------------------------------------------------+
FP port |  PHYS | SECUR | MAC_0 | RWR_0 | L2LKP | L3LKP | QUEUE |SWICHF
   1       0       0       0       0,1     0       0       0,1     0
   2       0       0       0       0,1     0       0       0,1     0
   3       0       0       0       0,1     0       0       0,1     0
   4       0       0       0       0,1     0       0       0,1     0
   5       1       0       0       0,1     0       0       0,1     0
   6       1       0       0       0,1     0       0       0,1     0
   7       1       0       0       0,1     0       0       0,1     0
   8       1       0       0       0,1     0       0       0,1     0
   9       2       0       0       0,1     0       0       0,1     0
   10      2       0       0       0,1     0       0       0,1     0
   11      2       0       0       0,1     0       0       0,1     0
   12      2       0       0       0,1     0       0       0,1     0
   13      3       1       1       2,3     1       1       2,3     0
   14      3       1       1       2,3     1       1       2,3     0
   15      3       1       1       2,3     1       1       2,3     0
   16      3       1       1       2,3     1       1       2,3     0
   17      4       1       1       2,3     1       1       2,3     0
   18      4       1       1       2,3     1       1       2,3     0
   19      4       1       1       2,3     1       1       2,3     0
   20      4       1       1       2,3     1       1       2,3     0
   21      5       1       1       2,3     1       1       2,3     0
   22      5       1       1       2,3     1       1       2,3     0
   23      5       1       1       2,3     1       1       2,3     0
   24      5       1       1       2,3     1       1       2,3     0
+-------------------------------------------------------------------------+
+-------------------------------------------------------------------------+
```

- 首先，您捕获L2中的数据包并查看转发决策是否正确。为此，请查看L2LKP映射列并确定与端口对应的ASIC实例编号。
- 接下来，使用命令**elam asic eureka instance x**在此实例上运行ELAM其中，x是ASIC实例编号，并为DBUS和RBUS配置触发器。使用命令**status**检查触发器的状态，并确认已配置触发器。

```
module-4(eureka-elam)# trigger dbus dbi ingress ipv4 if source-ipv4-address 192.0.2.2
destination-ipv4-address 192.0.2.4 rbi-corelate
module-4(eureka-elam)# trigger rbus rbi pb1 ip if cap2 1

module-4(eureka-elam)# status

Slot: 4, Instance: 1
EU-DBUS: Configured
trigger dbus dbi ingress ipv4 if source-ipv4-address 192.168.10.1
EU-RBUS: Configured
trigger rbus rbi pb1 ip if cap2 1
```

- 使用命令**start**激活触发器，并使用命令**status**验证触发器的状态，以确认触发器已准备好。

```
module-4(eureka-elam)# start
module-4(eureka-elam)# status

Slot: 4, Instance: 1 EU-DBUS: Armed <<<<<<<<<<
trigger dbus dbi ingress ipv4 if source-ipv4-address 192.168.10.1
EU-RBUS: Armed <<<<<<<<<<
trigger rbus rbi pb1 ip if cap2 1
```

- 一旦状态显示触发器已准备就绪，它们就可以捕获了。此时，您必须发送流量通过，并再次检查状态，以查看您的触发器是否实际触发。

```
module-4(eureka-elam)# status

Slot: 4, Instance: 1
EU-DBUS: Triggered    <<<<<<<<<<trigger dbus dbi ingress ipv4 if source-ipv4-address
192.168.10.1 EU-RBUS: Triggered <<<<<<<<<<
trigger rbus rbi pb1 ip if cap2 1
```

- 触发后，请检查rbus和dbus的数据包序列号，以确认它们都捕获了相同的数据包。这可以通过命令**show dbus**完成 **| i序列；show rbus | i seq**。如果序列号匹配，您可以查看dbus和rbus的内容。否则，请重新运行捕获，直到您可以捕获同一数据包为止。

  注意：为了提高准确性，请始终多次运行ELAM以确认转发问题。

- 您可以使用命令**show dbus**和**show rbus**查看rbus和dbus的内容。捕获中的重要内容是序列号和源/目标索引。Dbus显示源索引，它告诉您接收数据包的端口。Rbus显示数据包转发到的端口的目标索引。此外，您还可以查看源和目的IP/MAC地址以及VLAN信息。
- 通过源和目标索引（也称为LTL索引），您可以使用命令**show system internal pixm info ltl #**检查关联的前面板端口。

适用于Nexus 7000 - M1/M2的ELAM（Lamira平台）

Lamira平台的程序也一样，但也有一些差异：

- 您使用关键字Lamira elam asic **lamira instance x运行ELAM**。
- 触发ELAM的命令包括：

```
module-4(lamira-elam)#trigger dbus ipv4 if source-ipv4-address 192.0.2.2 destination-ipv4-
address 192.0.2.4
module-4(lamira-elam)# trigger rbus
```

- 您使用**status**命令检查状态，并确保它们在发送流量之前为Armed并在捕获流量之后触发。
- 然后，您可以按照类似于Eureka的方式解释dbus和show bus的输出。

适用于Nexus 7000 - F2/F2E的ELAM（Clipper平台）

同样，操作步骤相似，只是触发器不同。几个差异如下：

- 使用关键字Clipper elam asic clipper实**例x运行ELAM**，并指定第2层或第3层模式。

```
module-4# elam asic  clipper instance 1
module-4(clipper-elam)#
```

- 触发ELAM的命令如下：

```
module-4(clipper-l2-elam)# trigger dbus ipv4 ingress if source-ipv4-address 192.0.2.3
destination-ipv4-address 192.0.2.2
module-4(clipper-l2-elam)# trigger rbus ingress if trig
```

- 您使用**status**命令检查状态，并确保它们在发送流量之前为Armed并在捕获流量之后触发。
- 然后，您可以按照类似于Eureka的方式解释dbus和show bus的输出。

适用于Nexus 7000 - F3的ELAM（侧翼平台）

同样，操作步骤相似，只是触发器不同。几个差异如下：

- 使用关键字Flanker **elam asic flanker instance x运行ELAM**并指定第2层或第3层模式。

```
module-4# elam asic flanker instance 1
module-4(flanker-elam)#
```

- 触发ELAM的命令如下：

```
module-9(fln-l2-elam)# trigger dbus ipv4 if destination-ipv4-address 10.1.1.2
module-9(fln-l2-elam)# trigger rbus ingress if trig
```

- 您使用**status**命令检查状态，并确保它们在发送流量之前为Armed并在捕获流量之后触发。
- 然后，您可以按照类似于Eureka的方式解释dbus和rbus的输出。

适用于Nexus 9000的ELAM（Tahoe平台）

在Nexus 9000中，该过程与Nexus 7000略有不同。对于Nexus 9000，请参阅[Nexus 9000云扩展ASIC(Tahoe)NX-OS ELAM — 思科链接]

- 首先，使用命令**show hardware internal tah interface #**检查接口映射。此输出中最重要的信息是**ASIC #、Slice #**和**源ID(srcid)**。
- 此外，您还可以使用**show system internal ethpm info interface #命令**仔细检查此信息 **| i src**。除前面列出的值外，此处的重要事项是dpid和dmod值。
- 使用命令**show module**检查模块编号。
- 通过附加模块x**附加到模块**，其中x是模块编号。
- 使用命令**module-1# debug platform internal tah elam asic #在模块上运行**ELAM
- 根据要捕获的流量类型(L2、L3、封装流量（如GRE或VXLAN等）配置内部或外部触发器：

```
Nexus9000(config)# attach module 1
module-1# debug platform internal tah elam asic 0
module-1(TAH-elam)# trigger init asic # slice # lu-a2d 1 in-select 6 out-select 0 use-src-id #
module-1(TAH-elam-insel6)# reset
module-1(TAH-elam-insel6)# set outer ipv4 dst_ip 192.0.2.1 src_ip 192.0.2.2
```

- 设置触发器后，使用命令**start**启动ELAM，发送流量并使用命令**report**查看输出。报告的输出显示传出和传入接口以及vlan ID、源和目标IP/MAC地址。

```
SUGARBOWL ELAM REPORT SUMMARY
slot - 1, asic - 1, slice - 1
==============================

Incoming Interface: Eth1/49
Src Idx : 0xd, Src BD : 10
Outgoing Interface Info: dmod 1, dpid 14
```

```
Dst Idx : 0x602, Dst BD : 10

Packet Type: IPv4
Dst MAC address: CC:46:D6:6E:28:DB
Src MAC address: 00:FE:C8:0E:27:15
.1q Tag0 VLAN: 10,  cos = 0x0
Dst IPv4 address: 192.0.2.1
Src IPv4 address: 192.0.2.2

Ver     =  4, DSCP    =    0, Don't Fragment = 0 Proto   =  1, TTL     =   64, More Fragments =
0 Hdr len = 20, Pkt len =   84, Checksum       = 0x667f
```

适用于Nexus 9000的ELAM（NorthStar平台）

NorthStar平台的过程与Tahoe平台相同，唯一的区别在于当进入ELAM模式时，使用关键字**ns**而不是**tah**:

```
module-1#debug platform internal ns elam asic 0
```

- **N9K Packet Tracer**

Nexus 9000 Packet Tracer工具可用于跟踪数据包的路径，并且其内置的流量统计计数器使其成为可用于间歇性/完全流量丢失场景的重要工具。当TCAM资源有限或无法运行其他工具时，它非常有用。此外，此工具无法捕获ARP流量，并且不显示数据包内容的详细信息（如Wireshark）。

要配置Packet Tracer，请使用以下命令：

```
N9K-9508#test packet-tracer src_ip



              <==== provide your src and dst ip
N9K-9508# test packet-tracer start                                  <==== Start packet tracer
N9K-9508# test packet-tracer stop                                   <==== Stop packet tracer
N9K-9508# test packet-tracer show                                   <==== Check for packet
matches
```

有关详细信息，请参阅链接Nexus 9000:Packet Tracer工具说明 — Cisco

- **Traceroute和Ping**

这些命令是两个最有用的命令，可用于快速确定连接问题。

Ping使用Internet控制消息协议(ICMP)将ICMP回应消息发送到特定目标，并等待该目标的ICMP回应应答。如果主机之间的路径工作正常且没有问题，您可以看到应答返回，ping操作成功。默认情况下，ping命令发送5x ICMP回应消息（两个方向大小相等），如果一切正常，您可以看到5x ICMP回应应答。有时，当交换机在地址解析协议(ARP)请求期间获取MAC地址时，最初的回应请求会失败。如果随后立即再次运行ping，则没有初始ping丢失。此外，还可以使用以下关键字设置ping次数、数据包大小、源、源接口和超时间隔：

```
F241.04.25-N9K-C93180-1# ping 10.82.139.39 vrf management
PING 10.82.139.39 (10.82.139.39): 56 data bytes
36 bytes from 10.82.139.38: Destination Host Unreachable
Request 0 timed out
64 bytes from 10.82.139.39: icmp_seq=1 ttl=254 time=23.714 ms
64 bytes from 10.82.139.39: icmp_seq=2 ttl=254 time=0.622 ms
64 bytes from 10.82.139.39: icmp_seq=3 ttl=254 time=0.55 ms
64 bytes from 10.82.139.39: icmp_seq=4 ttl=254 time=0.598 ms
```

```
F241.04.25-N9K-C93180-1# ping 10.82.139.39 ?
  <CR>
  count             Number of pings to send
  df-bit            Enable do not fragment bit in IP header
  interval          Wait interval seconds between sending each packet
  packet-size       Packet size to send
  source            Source IP address to use
  source-interface  Select source interface
  timeout           Specify timeout interval
  vrf               Display per-VRF information
```

Traceroute用于标识数据包在到达其目的地之前经过的各种跳。这是一个非常重要的工具，因为它有助于确定发生故障的L3边界。还可以使用以下关键字使用端口、源和源接口：

```
F241.04.25-N9K-C93180-1# traceroute 10.82.139.39 ?
  <CR>
  port              Set destination port
  source            Set source address in IP header
  source-interface  Select source interface
  vrf               Display per-VRF information
```

```
Nexus_1(config)# traceroute 192.0.2.1
traceroute to 192.0.2.1 (192.0.2.1), 30 hops max, 40 byte packets
 1 198.51.100.3 (198.51.100.3)  1.017 ms  0.655 ms  0.648 ms
 2 203.0.113.2 (203.0.113.2)  0.826 ms  0.898 ms  0.82 ms
 3 192.0.2.1 (192.0.2.1)  0.962 ms  0.765 ms  0.776 ms
```

- PACL/RACL/VACL

ACL代表访问控制列表。它是一种重要工具，允许您根据相关定义的标准过滤流量。在ACL中填入匹配条件的条目后，即可应用它来捕获入站或出站流量。ACL的一个重要方面就是它能够提供流量统计数据的计数器。术语PACL/RACL/VACL是指这些ACL的各种实施，它们允许您使用ACL作为强大的故障排除工具，特别是对于间歇性流量丢失而言。下面简要介绍这些术语：

- PACL代表端口访问控制列表：将访问列表应用于L2交换机端口/接口时，该访问列表称为PACL。
- RACL代表路由器访问控制列表：将访问列表应用于L3路由端口/接口时，该访问列表称为RACL。
- VACL代表VLAN Access Control List:您可以将VACL配置为应用于路由到VLAN或路由出VLAN或在VLAN内桥接的所有数据包。VACL严格用于安全数据包过滤并将流量重定向到特定物理接口。VACL不按方向（入口或出口）定义。

下表提供了不同ACL版本的比较。

ACL类型	PACL	RACL	VACL
功能	过滤L2接口上接收的流量。	过滤L3接口上接收的流量	过滤vLAN流量
APPLIED ON	- L2接口/端口。 - L2端口通道接口。 — 如果应用于TRUNK端口，ACL将过滤该TRUNK端口上允许的所有VLAN上的流量。	- VLAN接口。 — 物理L3接口。 - L3子接口。 - L3端口通道接口。 -管理接口.	启用后，ACL将应用到该VLAN中的所有端口（包括端口）。
应用方向	仅限入站。	入站或出站	-

下面是如何配置访问列表的示例。有关详细信息，请参阅链接Cisco Nexus 9000系列NX-OS安全配置指南，版本9.3(x) — 配置IP ACL [Cisco Nexus 9000系列交换机] — 思科

```
Nexus93180(config)# ip access-list



Nexus93180(config-acl)# ?
  <1-4294967295>  Sequence number
  deny            Specify packets to reject
  fragments       Optimize fragments rule installation
  no              Negate a command or set its defaults
  permit          Specify packets to forward
  remark          Access list entry comment
  show            Show running system information
  statistics      Enable per-entry statistics for the ACL
  end             Go to exec mode
  exit            Exit from command interpreter
  pop             Pop mode from stack or restore from name
  push            Push current mode to stack or save it under name
  where           Shows the cli context you are in

Nexus93180(config)# int e1/1

Nexus93180(config-if)# ip port access-group



         >>>>>> When you configure ACL like this, it is PACL.
in  Inbound packets
Nexus93180(config-if)# ip access-group



              >>>>>> When you configure ACL like this, it is RACL.
  in   Inbound packets
  out  Outbound packets
```

- LOGFLASH

LogFlash是Nexus平台上可用的一种持久存储类型，可作为外部紧凑型闪存、USB设备或管理引擎中的嵌入式磁盘。如果从交换机上删除，系统会定期通知用户缺少LogFlash。Logflash安装在Supervisor上，保存历史数据，例如记帐日志、系统日志消息、调试和嵌入式事件管理器(EEM)输出。本文稍后将讨论EEM。您可以使用以下命令检查LogFlash的内容：

```
Nexus93180(config)# dir logflash:
          0    Nov 14 04:13:21 2019  .gmr6_plus
      20480    Feb 18 13:35:07 2020  ISSU_debug_logs/
         24    Feb 20 20:43:24 2019  arp.pcap
         24    Feb 20 20:36:52 2019  capture_SYB010L2289.pcap
       4096    Feb 18 17:24:53 2020  command/
       4096    Sep 11 01:39:04 2018  controller/
       4096    Aug 15 03:28:05 2019  core/
       4096    Feb 02 05:21:47 2018  debug/
    1323008    Feb 18 19:20:46 2020  debug_logs/
       4096    Feb 17 06:35:36 2020  evt_log_snapshot/
       4096    Feb 02 05:21:47 2018  generic/
       1024    Oct 30 17:27:49 2019  icamsql_1_1.db
      32768    Jan 17 11:53:23 2020  icamsql_1_1.db-shm
     129984    Jan 17 11:53:23 2020  icamsql_1_1.db-wal
       4096    Feb 14 13:44:00 2020  log/
```

```
      16384    Feb 02 05:21:44 2018  lost+found/
       4096    Aug 09 20:38:22 2019  old_upgrade/
       4096    Feb 18 13:40:36 2020  vdc_1/

Usage for logflash://sup-local
 1103396864 bytes used
 7217504256 bytes free
 8320901120 bytes total
```

如果用户重新加载设备，或者设备由于事件而突然自行重新加载，所有日志信息都将丢失。在这种情况下，LogFlash可以提供历史数据，可以对这些数据进行审核，以确定问题的可能原因。当然，需要进一步开展尽职调查以找出根本原因，从而提供一些提示，告诉您万一再次发生此事件时需要寻找什么。

有关如何在设备上安装logflash的信息，请参阅Nexus 7000日志记录功能 — Cisco链接。

- **OBFL**

OBFL代表OnBoard Failure Logging。它是Nexus Top of Rack和模块化交换机都可用的持久性存储类型。与LogFlash一样，设备重新加载后信息会保留。OBFL存储故障和环境数据等信息。信息因平台和模块而异，但以下是Nexus 93108平台模块1的输出示例（即只有一个模块的固定机箱）：

```
Nexus93180(config)# show logging onboard module 1 ?
*** No matching command found in current mode, matching in (exec) mode ***
  <CR>
  >                     Redirect it to a file
  >>                    Redirect it to a file in append mode
  boot-uptime           Boot-uptime
  card-boot-history     Show card boot history
  card-first-power-on   Show card first power on information
  counter-stats         Show OBFL counter statistics
  device-version        Device-version
  endtime               Show OBFL logs till end time mm/dd/yy-HH:MM:SS
  environmental-history Environmental-history
  error-stats           Show OBFL error statistics
  exception-log         Exception-log
  internal              Show Logging Onboard Internal
  interrupt-stats       Interrupt-stats
  obfl-history          Obfl-history
  stack-trace           Stack-trace
  starttime             Show OBFL logs from start time mm/dd/yy-HH:MM:SS
  status                Status
  |                     Pipe command output to filter


Nexus93180(config)# show logging onboard module 1 status
---------------------------
OBFL Status
---------------------------
    Switch OBFL Log:                                    Enabled
    Module:  1 OBFL Log:                                Enabled
    card-boot-history                                   Enabled
    card-first-power-on                                 Enabled
    cpu-hog                                             Enabled
    environmental-history                               Enabled
    error-stats                                         Enabled
    exception-log                                       Enabled
    interrupt-stats                                     Enabled
    mem-leak                                            Enabled
    miscellaneous-error                                 Enabled
    obfl-log (boot-uptime/device-version/obfl-history)  Enabled
```

```
    register-log                                        Enabled
    system-health                                       Enabled
    temp Error                                          Enabled
    stack-trace                                         Enabled
```

同样，此信息在设备被用户有意重新加载或由于触发重新加载的事件重新加载时非常有用。在这种情况下，OBFL信息有助于从线路卡的角度确定问题所在。**show logging onboard**命令是一个很好的起点。请记住，您必须从模块情景内部捕获才能获得所需的一切。确保使用**show logging onboard module x**或**attach mod x;show logging onboard**。

- **事件历史记录**

事件历史记录是功能强大的工具之一，它可以为您提供在Nexus上运行的进程发生的各种事件的相关信息。换句话说，在Nexus平台上运行的每个进程都有在后台运行的事件历史记录，并存储有关该进程各种事件的信息（将其视为持续运行的调试）。 这些事件历史记录是非持久性的，设备重新加载时，存储的所有信息都将丢失。当您发现某个流程的问题并想要排除该流程故障时，这些选项非常有用。例如，如果OSPF路由协议不能正常工作，您可以使用与OSPF关联的事件历史记录来确定OSPF进程发生故障的位置。您可以找到与Nexus平台上几乎每个进程相关的事件历史记录，例如CDP/STP、UDLD、LACP/OSPF、EIGRP/BGP等。

通常使用此方法通过参考示例检查进程的事件历史记录。每个流程都有多个选项，因此要使用**吗？**检查进程下的各种可用选项。

```
Nexus93180(config)# show
```

```
Nexus93180# show ip ospf event-history ?
  adjacency       Adjacency formation logs
  cli             Cli logs
  event           Internal event logs
  flooding        LSA flooding logs
  ha              HA and GR logs
  hello           Hello related logs
  ldp             LDP related logs
  lsa             LSA generation and databse logs
  msgs            IPC logs
  objstore        DME OBJSTORE related logs
  redistribution  Redistribution logs
  rib             RIB related logs
  segrt           Segment Routing logs
  spf             SPF calculation logs
  spf-trigger     SPF TRIGGER related logs
  statistics      Show the state and size of the buffers
  te              MPLS TE related logs

Nexus93180# show spanning-tree internal event-history ?
  all      Show all event historys
  deleted  Show event history of deleted trees and ports
  errors   Show error logs of STP
  msgs     Show various message logs of STP
  tree     Show spanning tree instance info
  vpc      Show virtual Port-channel event logs
```

- **调试**

调试是NX-OS中功能强大的工具，允许您运行实时故障排除事件并将其记录到文件或在CLI中显示。强烈建议记录文件中的调试输出，因为它们确实影响CPU性能。在CLI上直接运行调试之前，请

务必小心。

通常仅在将问题确定为单个进程并要检查此进程如何实时处理网络中的实际流量时才运行调试。您需要根据定义的用户帐户权限启用调试功能。

就像事件历史记录一样，您可以在Nexus设备上为每个进程运行调试，例如CDP/STP、UDLD、LACP/OSPF、EIGRP/BGP等。

这是通常运行进程调试的方式。每个流程都有多个选项，因此要使用**吗？**检查进程下的各种可用选项。

```
Nexus93180# debug
```

```
Nexus93180# debug spanning-tree ?
  all      Configure all debug flags of stp
  bpdu_rx  Configure debugging of stp bpdu rx
  bpdu_tx  Configure debugging of stp bpdu tx
  error    Configure debugging of stp error
  event    Configure debugging of Events
  ha       Configure debugging of stp HA
  mcs      Configure debugging of stp MCS
  mstp     Configure debugging of MSTP
  pss      Configure debugging of PSS
  rstp     Configure debugging of RSTP
  sps      Configure debugging of Set Port state batching
  timer    Configure debugging of stp Timer events
  trace    Configure debugging of stp trace
  warning  Configure debugging of stp warning

Nexus93180# debug ip ospf ?
  adjacency               Adjacency events
  all                     All OSPF debugging
  database                OSPF LSDB changes
  database-timers         OSPF LSDB timers
  events                  OSPF related events
  flooding                LSA flooding
  graceful-restart        OSPF graceful restart related debugs
  ha                      OSPF HA related events
  hello                   Hello packets and DR elections
  lsa-generation          Local OSPF LSA generation
  lsa-throttling          Local OSPF LSA throttling
  mpls                    OSPF MPLS
  objectstore             Objectstore Events
  packets                 OSPF packets
  policy                  OSPF RPM policy debug information
  redist                  OSPF redistribution
  retransmission          OSPF retransmission events
  rib                     Sending routes to the URIB
  segrt                   Segment Routing Events
  snmp                    SNMP traps and request-response related events
  spf                     SPF calculations
  spf-trigger             Show SPF triggers
```

- **金牌**

GOLD代表通用在线诊断。顾名思义，这些测试通常用作系统运行状况检查，并用于检查或验证有问题的硬件。已执行各种在线测试，这些测试基于正在使用的平台，其中一些测试具有破坏性，而

另一些测试不具有破坏性。这些在线测试可分为以下几类：

- **启动诊断**:这些测试是在设备启动时运行的测试。他们还检查管理引擎与模块之间的连接，包括所有ASIC的数据和控制平面之间的连接。ManagementPortLoopback和EOBCLoopback等测试具有破坏性，而OBFL和USB测试不具有破坏性。
- **运行时或运行状况监控诊断**：这些测试提供有关设备运行状况的信息。这些测试是无中断的，在后台运行以确保硬件的稳定性。您可以根据需要或出于故障排除目的启用/禁用这些测试。
- **按需诊断**：可以按需重新运行上述所有测试，以定位问题。

您可以使用以下命令检查适用于交换机的各种类型的在线测试：

```
Nexus93180(config)# show diagnostic content module all
Diagnostics test suite attributes:
B/C/* - Bypass bootup level test / Complete bootup level test / NA
P/*   - Per port test / NA
M/S/* - Only applicable to active / standby unit / NA
D/N/* - Disruptive test / Non-disruptive test / NA
H/O/* - Always enabled monitoring test / Conditionally enabled test / NA
F/*   - Fixed monitoring interval test / NA
X/*   - Not a health monitoring test / NA
E/*   - Sup to line card test / NA
L/*   - Exclusively run this test / NA
T/*   - Not an ondemand test / NA
A/I/* - Monitoring is active / Monitoring is inactive / NA

Module 1: 48x10/25G + 6x40/100G Ethernet Module (Active)

                                                    Testing Interval
ID     Name                                Attributes     (hh:mm:ss)
____   _____________________________       ___________    _____________
 1)    USB--------------------------->     C**N**X**T*     -NA-
 2)    NVRAM------------------------->     ***N******A     00:05:00
 3)    RealTimeClock----------------->     ***N******A     00:05:00
 4)    PrimaryBootROM---------------->     ***N******A     00:30:00
 5)    SecondaryBootROM-------------->     ***N******A     00:30:00
 6)    BootFlash--------------------->     ***N******A     00:30:00
 7)    SystemMgmtBus----------------->     **MN******A     00:00:30
 8)    OBFL-------------------------->     C**N**X**T*     -NA-
 9)    ACT2-------------------------->     ***N******A     00:30:00
10)    Console----------------------->     ***N******A     00:00:30
11)    FpgaRegTest------------------->     ***N******A     00:00:30
12)    Mce--------------------------->     ***N******A     01:00:00
13)    AsicMemory-------------------->     C**D**X**T*     -NA-
14)    Pcie-------------------------->     C**N**X**T*     -NA-
15)    PortLoopback------------------>     *P*N**XE***     -NA-
16)    L2ACLRedirect----------------->     *P*N***E**A     00:01:00
17)    BootupPortLoopback------------>     CP*N**XE*T*     -NA-
```

要显示所提到的17项测试中的每项测试，您可以使用以下命令：

```
Nexus93180(config)#show diagnostic description module 1 test all
USB :
        A bootup test that checks the USB controller initialization
        on the module.

NVRAM :
        A health monitoring test, enabled by default that checks the
        sanity of the NVRAM device on the module.
```

```
RealTimeClock :
        A health monitoring test, enabled by default that verifies
        the real time clock on the module.

PrimaryBootROM :
        A health monitoring  test that verifies the primary BootROM
        on the module.

SecondaryBootROM :
        A health monitoring  test that verifies the secondary BootROM
        on the module.

BootFlash :
        A Health monitoring test, enabled by default, that verifies
        access to the internal compactflash devices.

SystemMgmtBus :
        A Health monitoring test, enabled by default, that verifies
        the standby System Bus.

OBFL :
        A bootup test that checks the onboard flash used for failure
        logging (OBFL) device initialization on the module.

ACT2 :
        A Health monitoring test, enabled by default, that verifies
        access to the ACT2 device.

Console :
        A health monitoring test,enabled by default that checks health
        of console device.

FpgaRegTest :
        A health monitoring test,enabled by default that checks read/write
        access to FPGA scratch registers on the module.

Mce :
        A Health monitoring test, enabled by default, that check for
        machine errors on sup.

AsicMemory :
        A bootup test that checks the asic memory.

Pcie :
        A bootup test that tests pcie bus of the module

PortLoopback :
        A health monitoring test that tests the packet path from
        the Supervisor card to the physical port in ADMIN DOWN state
        on Linecards.

L2ACLRedirect :
        A health monitoring test, enabled by default, that does a
        non disruptive loopback for TAHOE asics to check the ACL Sup
        redirect with the CPU port.

BootupPortLoopback :
        A Bootup test that tests the packet path from the Supervisor
        card to all of the physical ports at boot time.
```

- **EEM**

EEM代表嵌入式事件管理器。它是一种强大的工具，允许您对设备进行编程，以便在发生特定事件时执行特定任务。它会监视设备上的各种事件，然后采取必要的措施来解决问题并可能恢复。
EEM包括三个主要组件，下面分别简要介绍：

- **事件语句：**这些是您想要监控并希望Nexus执行某些操作的事件，例如采取解决方法或只通知SNMP服务器或显示CLI日志等。
- **操作语句：**这些是EEM在事件触发后采取的措施。这些操作可能只是禁用接口或执行一些show命令，并将输出复制到ftp服务器上的文件，发送电子邮件等。
- **策略：**它基本上是一个事件，它与可通过CLI或bash脚本在Supervisor上配置的一个或多个action语句相结合。您还可以使用python脚本调用EEM。一旦在管理引擎上定义了策略，它就会将该策略推送到相关模块。

有关EEM的详细信息，请参阅链接Cisco Nexus 9000系列NX-OS系统管理配置指南9.2(x)版 — 配置嵌入式事件管理器[Cisco Nexus 9000系列交换机] — 思科。

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。