Probleemoplossing Nexus cheatblad voor beginners

Inhoud

Inleiding

In dit document worden verschillende tools beschreven die beschikbaar zijn voor probleemoplossing bij Nexus-producten die u kunt gebruiken om een probleem te diagnosticeren en op te lossen.

Overzicht

Het is belangrijk om te begrijpen welke hulpmiddelen beschikbaar zijn en in welk scenario u hen voor maximumwinst zou gebruiken. In feite is een bepaald gereedschap soms niet haalbaar simpelweg omdat het is ontworpen om aan iets anders te werken.

Deze tabel compileert de verschillende tools voor probleemoplossing op het Nexus platform en hun mogelijkheden. Raadpleeg voor meer informatie en CLI-voorbeelden het gedeelte Nexus Tools.

TOOLS	FUNCTIE	VOORBEELD VAN GEBRUIKSCASES	Voordelen	NADELEN	PERSISTENTIE	UITGEVOERD VLIEGTUIG	CLI-OPDRACHT GEBRUIKT
Ethanalyzer	Leg verkeer vast dat bestemd is voor of	Problemen met verkeerstraagheid, latentie en congestie	Uitstekend voor traagheid, congestie en latency	Meestal alleen controle vliegtuig verkeer, tarief beperkt	N.v.t.	Bedieningsplane. Kan in sommi	#ethanalyzer lok interface-inband #ethanalyzer Weergavefilter v lokale interface

	afkomstig is van de CPU		problemen			ge scenario's voor gegevensvlak worden gebruikt (SPAN naar CPU)	[interface-ID] [WORD] voorbeeld: #ethanalyzer lo interface Ethern display filter ICM
OVERSPANNING	Leg een hoop pakketten vast en spiegel deze	Mislukt ping s, out-of-order pakketten, enzovoort	Uitstekend voor intermitterend verkeersverlies	Vereist extern apparaat dat snuffelsoftware draait Vereist TCAM-bronnen	SPAN-sessie moet worden geconfigureerd en ingeschakeld/uitgeschakeld	Controle + gegevens	#monitor sessie #description [NA #source interfac [poort-ID] #destination inte [poort-ID] #no gesloten
DM-fout	Leg alleen verkeer vast dat is bestemd voor of afkomstig is van de CPU voor Broadcom Nexus-apparaten	Problemen met verkeerstraagheid, latentie en congestie	Uitstekend voor traagheid, congestie en latency problemen	Alleen voor Broadcom Nexus-apparaten. Snelheid beperkt (CloudScale Nexus 9k heeft wel SPAN-naar-CPU)	N.v.t.	Bedieningsplane. Kan in sommige scenario's voor gegevensvlak worden gebruikt	Afhankelijk van platform, zie ELAM - Overzic Cisco
ELAM	Leg één pakket vast dat de Nexus-switch ingaat [of uitstroomt, indien Nexus 7K] dit wel doet	Controleer of het pakket de Nexus bereikt, controleer de doorsturen beslissingen, controleer het pakket op wijzigingen, controleer de interface/VLAN van het pakket, enzovoort	Uitstekend voor pakketstroom en doorsturen problemen. Niet opdringerig	Vereist een diepgaand begrip van de hardware. Maakt gebruik van unieke triggermechanismen die architectuurspecifiek zijn. Alleen nuttig als u weet welk verkeer u wilt onderzoeken	N.v.t.	Controle + gegevens	# module [MOD NUMMER] # de platform intern <

Nexus 9k pakkettracer	Pad van het pakket detecteren	Connectiviteitsproblemen en pakketverlies	Biedt teller voor stroomstatistieken nuttig voor intermitterend/volledig verlies. Perfect voor lijnkaarten zonder TCAM inkervingen	Kan ARP verkeer niet opnemen. Alleen werken voor Nexus 9k	N.v.t.	Gegevens + controle	# test packet-tra src_IP [SOURC dst_IP [BESTEMMING test pakkettrace # test pakkettrac stop # test pakkettracer sh
traceroute	Detecteer het pad van het pakket met betrekking tot L3 hop	Mislukte pings, kan host/bestemming/ internet niet bereiken, enzovoort	Detecteert de verschillende hops op het pad om L3-storingen te isoleren.	Geeft alleen aan waar de L3-grens is verbroken (identificeert de kwestie zelf niet)	N.v.t.	Gegevens + controle	# traceroute [BESTEMMING De argumenten omvatten: poort, poortnum bron, interface, bron-interface
Ping	Connectiviteit tussen twee punten in een netwerk testen	Bereikbaarheid tussen apparaten testen	Een snel en eenvoudig hulpmiddel om connectiviteit te testen	Identificeert alleen of de host bereikbaar is of niet	N.v.t.	Gegevens + controle	# ping [BESTEMMING De argumenten omvatten: telling, pakketgr broninterface, interval, multica loopback, timeo
PAL/RACL/VACL	Leg verkeer in/uit een bepaalde poort of VLAN	Intermitterend pakketverlies tussen hosts, bevestig als pakketten aankomen/vertrekken bij de Nexus, enzovoort	Uitstekend voor intermitterend verkeersverlies	Vereist TCAM-bronnen. Voor sommige modules is handmatig TCAM-carving vereist	Persistent (toegepast op) running-configuratie)	Gegevens + controle	# ip access-list NAME] # ip poo access-groep [A NAME] # ip acc groep [ACL NAM De argumenten omvatten: ontkennen, fragmenten, nee vergunning, opmerking, tone statistieken, ein uitgang, pop, du waar
LogFlash	Hierop worden historische gegevens voor de switch wereldwijd	Plotseling apparaat herladen / afsluiten, op elk moment dat een apparaat wordt herladen, logbestand flash-gegevens biedt enige informatie	Informatie over het opnieuw laden van het apparaat (permanente opslag) blijft behouden	Extern op Nexus 7K = Moet worden geïnstalleerd/geïntegreerd op het supervisor-platform voor deze logbestanden die moeten	Persistent opnieuw laden	Gegevens + controle	# dir logflash:

	opgeslagen, zoals logboekaccounts, crashbestanden en gebeurtenissen, ongeacht het opnieuw laden van het apparaat	die nuttig kan zijn in analyse		worden verzameld (con is niet van toepassing op 3K/9K aangezien logflash een verdeling van het interne opslagapparaat is)			
OBFL	Slaat historische gegevens op een specifieke module zoals storingen en milieu-informatie	Plotseling apparaat herladen / afsluiten, op elk moment dat een apparaat wordt herladen, logbestand flash-gegevens biedt enige informatie die nuttig kan zijn	Informatie over het opnieuw laden van het apparaat (permanente opslag) blijft behouden	Ondersteunt een beperkt aantal lees- en schrijfbewerkingen	Persistent opnieuw laden	Gegevens + controle	# toont vastlegg aan boord modu [#] De argumenten omvatten: boot-uptime, ka boot-geschieden kaart-first-power contrasistans, apparaat-versie endtime, milieu-geschiedenis, fc stats, uitzonderi log, intern, intern stats, obfl-geschiedenis, st time, status
Geschiedenis van gebeurtenissen	Wanneer u informatie nodig hebt voor een specifiek proces dat momenteel wordt uitgevoerd	Elk proces in nexus heeft zijn eigen gebeurtenisgeschiedenissen zoals CDP, STP, OSPF, EIGRP, BGP, vPC, LACP, etc.	Probleemoplossing voor een specifiek proces op Nexus	Informatie gaat verloren wanneer het apparaat opnieuw wordt geladen (niet-persistent)	niet persistent	Gegevens + controle	# toon [PROCES interne gebeurtenisgesc enis [ARGUMEN De argumenten omvatten: Aangrenzing, cli gebeurtenis, overstroming, ha hallo, ldp, lsa, m objstore, herverdeling, rib segment, spf-trig statistieken,
Debugs	Wanneer u meer gedetailleerde real-time/live-informati	Debug op elk proces in nexus kan worden gedaan zoals CDP, STP, OSPF, IGRP, BGP, vPC, LACP	Probleemoplossing voor een specifiek proces dat op Nexus wordt	Kan van invloed zijn op netwerkprestaties	niet persistent	Gegevens + controle	# debug proces [PROCESSING] voorbeeld: # debug ip ospf

	e nodig hebt voor een specifiek proces	enzovoort	uitgevoerd in realtime voor meer granulariteit				
GOUD	Biedt opstart, uitvoertijd en diagnostiek op aanvraag voor hardwarecomponenten (zoals I/O en Supervisor modules)	Test hardware zoals USB, Bootflash, OBFL, ASIC-geheugen, PCIE, Port loopback, NVRAM, enzovoort	Kan fouten in de hardware detecteren en de nodige corrigerende maatregelen nemen alleen bij release 6(2)8 en hoger	Detecteert alleen hardwareproblemen	niet persistent	N.v.t.	# toon diagnosti inhoudsmodule toon diagnostisc beschrijving mo [#] test alles
EEM	Bewaak gebeurtenissen op het apparaat en neem de nodige maatregelen	Elke apparaatactiviteit waarvoor enige actie/tijdelijke oplossing/kennisgeving nodig is, zoals het uitschakelen van de interface, ventilatorstoring, CPU-gebruik, enzovoort	Ondersteunt Python-scripts	U moet over netwerkbeheerdersrechten beschikken om EEM te kunnen configureren	EM script en trigger bevinden zich in configuratie	N.v.t.	Varieert, zie De ingesloten gebeurtenisbeh configureren

Nexus tools

Als u meer duidelijkheid wilt over verschillende opdrachten en de syntaxis of opties van deze opdrachten, raadpleegt u Cisco Nexus 9000 Series Switches - Opdrachtreferenties - Cisco.

- **Ethanalyzer**

Ethanalyzer is een NX-OS-tool die is ontworpen om pakketverkeer via CPU's op te nemen. Alles wat de CPU raakt, of er nu toegang of uitgang is, kan met deze tool worden opgenomen. Het is gebaseerd op de veelgebruikte opensource-netwerkprotocolanalyzer Wireshark. Raadpleeg voor meer informatie over deze tool de Ethanalyzer op Nexus 7000 Problemen oplossen Guide - Cisco

Het is belangrijk om op te merken dat Ethanalyzer over het algemeen al het verkeer van en naar

de supervisor opneemt, dat wil zeggen dat het geen interfacespecifieke opnamen ondersteunt. Er zijn specifieke interfaceverbeteringen beschikbaar voor bepaalde platforms in recentere codepunten. Ethanalyzer neemt ook alleen verkeer op dat via CPU is geschakeld, en niet via hardware. U kunt bijvoorbeeld verkeer opnemen via de inband-interface, de beheerinterface of een voorpaneelpoort (indien ondersteund):

```
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
```

Deze output toont weinig van de berichten die met Ethanalyzer kunnen worden gevangen. Houd er rekening mee dat Ethanalyzer standaard maximaal 10 pakketten opneemt. U kunt deze opdracht echter gebruiken om de CLI te vragen pakketten voor onbepaalde tijd op te nemen. Gebruik CTRL+C om de opnamemodus te verlaten.

```
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
```

U kunt filters ook gebruiken met Ethanalyzer om zich te richten op specifiek verkeer. Er zijn twee soorten filters die u kunt gebruiken met ethanalzyer, ze staan bekend als Capture filters en Display filters. Een opnamefilter neemt alleen het verkeer op dat voldoet aan de criteria die in het opnamefilter zijn gedefinieerd. Een weergavefilter vangt nog steeds al het verkeer op, maar alleen het verkeer dat voldoet aan de criteria die in het weergavefilter zijn gedefinieerd, wordt weergegeven.

```
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
```

U kunt ook pakketten opnemen met de detailoptie en ze bekijken in uw terminal, vergelijkbaar met hoe u in Wireshark. Hierdoor kunt u de volledige headerinformatie zien op basis van het pakketdissectorresultaat. Als een frame bijvoorbeeld is versleuteld, kunt u de versleutelde payload niet zien. Zie dit voorbeeld:

```
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
```

Met Ethanalyzer kunt u:

- Schrijf de uitvoer (een PCAP-bestand) naar de opgegeven bestandsnaam op de verschillende doelbestandssystemen: opstartflitser, logflitser, USB, enz... U kon dan het opgeslagen bestand naar buiten het apparaat overbrengen en het in Wireshark bekijken, zoals nodig.
- Lees een bestand van bootflash en weergave op uw terminal. Net zoals wanneer u direct van de CPU-interface leest, kunt u ook de volledige pakketinformatie weergeven als u het detailsleutelwoord gebruikt.

Zie voorbeelden hiervan voor verschillende interfacebronnen en uitvoeropties:

```
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]
```

- OVERSPANNING

SPAN staat voor SwitchPort Analyzer en wordt gebruikt om al het verkeer van een interface en spiegel die verkeer naar een bestemmingshaven op te nemen. De bestemmingshaven verbindt zich typisch met een hulpmiddel van de netwerkanalysator (zoals een PC die Wireshark in werking stellen) dat u toestaat om het verkeer te analyseren dat door die havens oversteekt. U kunt SPAN voor verkeer vanaf één poort of meerdere poorten en VLAN's maken.

SPAN-sessies omvatten een bronpoort en een doelpoort. Een bronpoort kan een Ethernet-poort (geen subinterfaces), poortkanalen, Supervisor Inband-interfaces zijn en kan niet tegelijkertijd een doelpoort zijn. Bovendien worden voor sommige apparaten zoals het 9300- en 9500-platform ook FEX-poorten (Fabric Extender) ondersteund. Een bestemmingshaven kan een Ethernet-poort (Access of Trunk), poortkanaal (Access of Trunk) en voor sommige apparaten, zoals de 9300 uplinkpoorten, worden ook ondersteund, terwijl FEX-poorten niet worden ondersteund.

U kunt meerdere SPAN-sessies configureren als een toegang/uitgang/beide. Er is een limiet aan het totale aantal SPAN-sessies dat een afzonderlijk apparaat kan ondersteunen. Een Nexus 9000 kan bijvoorbeeld maximaal 32 sessies ondersteunen, terwijl een Nexus 7000 alleen 16 sessies kan ondersteunen. U kunt dit controleren op de CLI of verwijzen naar de SPAN-configuratiehandleidingen voor het product dat u gebruikt.

Houd er rekening mee dat voor elke NX-OS release en het producttype de ondersteunde interfacetypen en functionaliteit verschillen. Raadpleeg de meest recente configuratierichtlijnen en -beperkingen voor het product en de versie die u gebruikt. Hier zijn de links voor respectievelijk Nexus 9000 en Nexus 7000:

Cisco Nexus 9000 Series NX-OS configuratiehandleiding voor systeembeheer, release 9.3(x) - SPAN configureren [Cisco Nexus 9000 Series Switches] - Cisco

Cisco Nexus 7000 Series NX-OS configuratiehandleiding voor systeembeheer - SPAN configureren [Cisco Nexus 7000 Series Switches] - Cisco

Er zijn verschillende typen SPAN-sessies. Enkele van de meest voorkomende types worden hier vermeld:

- Local SPAN: een type SPAN-sessie waarbij de bron en de doelhost lokaal zijn voor de switch. Met andere woorden, alle configuratie die nodig is om de SPAN-sessie in te stellen wordt toegepast op één switch, dezelfde switch waar de bron- en doelhostpoorten zich bevinden.
- Remote SPAN (RSPAN): een type SPAN-sessie waarbij de bron en de doelhost niet lokaal zijn voor de switch. Met andere woorden, u vormt bron RSPAN-sessies op één switch switch en doelmap RSPAN op de doelmap en breidt de connectiviteit met RSPAN VLAN uit.

Opmerking: RSPAN wordt niet ondersteund op Nexus

- Extended Remote SPAN (ERSPAN): De switch kapselt het gekopieerde frame in met een GRE-tunnelkop (Generic Routing Encapsulation) en routeert het pakket naar de ingestelde bestemming. U vormt bron- en doelsessies op de inkapselings- en decapsulatie-switches (twee verschillende apparaten). Dit geeft ons de mogelijkheid om verkeer te SPAN via een Layer 3 netwerk.
- SPAN-to-CPU: een naam die wordt gegeven aan een speciaal type SPAN-sessie waarbij uw bestemmingshaven de supervisor of CPU is. Het is een vorm van lokale SPAN-sessie en kan worden gebruikt in gevallen waarin u geen standaard SPAN-sessie kunt gebruiken. Enkele gemeenschappelijke redenen zijn: geen beschikbare of geschikte SPAN-bestemmingspoorten, site niet toegankelijk of onbeheerde site, geen apparaat beschikbaar dat kan verbinden met SPAN-bestemmingspoorten, enzovoort. Raadpleeg voor meer informatie deze link naar de Nexus 9000 Cloud Scale ASIC NX-OS SPAN-to-CPU procedure - Cisco. Het is belangrijk om te onthouden dat SPAN-to-CPU snelheid wordt beperkt door CoPP (Control Plane Policing), dus sniffing een of meer broninterfaces die de policer overschrijden, kunnen druppels voor de SPAN naar CPU-sessie opleveren. Als dit gebeurt, zijn de gegevens niet 100% weerspiegelend van wat op de draad is, zodat is SPAN aan CPU niet altijd geschikt voor het oplossen van problemen scenario's met hoge gegevenssnelheid en/of intermitterend verlies. Zodra u een SPAN naar CPU-sessie configureert en deze administratief inschakelt, moet u Ethanalyzer uitvoeren om het verkeer te zien dat naar de CPU wordt verzonden om de analyse dienovereenkomstig uit te voeren.

Dit is een voorbeeld van hoe u een eenvoudige lokale SPAN-sessie kunt configureren op een Nexus 9000 switch:

```
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
```

Dit voorbeeld toont de configuratie van een SPAN-naar-CPU sessie die is opgestart, en vervolgens het gebruik van Ethanalyzer om het verkeer op te nemen:

```
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 is een type SPAN-TO-CPU sessie voor Broadcom-gebaseerde Nexus-platforms. Het concept is hetzelfde als bij SPAN-to-CPU en is beperkt tot 50 pps (pakketten per seconde). De functie is geïmplementeerd om het interne gegevenspad met de bcm-shell CLI te debuggen. Wegens de bijbehorende beperkingen is er geen NX-OS CLI om gebruikers toe te staan om sessies van SPAN aan de Opstelling te configureren omdat het controleverkeer kan beïnvloeden en CoPP-klassen kan verbruiken.

- ELAM

ELAM staat voor Embedded Logic Analyzer Module. Het biedt de mogelijkheid om in de ASIC te kijken en te bepalen welke doorsturen beslissingen worden genomen voor **ÉÉN** pakket. Zo, met ELAM kunt u identificeren of het pakket de Forwarding Engine en op welke Poorten/VLAN informatie bereikt. U kunt ook controleren op L2 - L4 pakketstructuur en of er wijzigingen in het pakket zijn aangebracht of niet.

Het is belangrijk om te begrijpen dat ELAM architectuurafhankelijk is en dat de procedure om een

pakket op te nemen van platform tot platform verschilt op basis van de interne architectuur. U moet de ASIC-toewijzingen van de hardware kennen om het gereedschap correct te kunnen toepassen. Voor Nexus 7000 worden twee opnamen gemaakt voor één pakket, één voordat de beslissing wordt genomen **Data BUS (DBUS)** en andere nadat de beslissing is genomen **Result BUS (RBUS)**. Wanneer u de DBUS-informatie bekijkt, kunt u zien wat/waar het pakket is ontvangen, evenals de Layer 2 tot 4-informatie. De resultaten in de RBUS laten zien waar het pakket naar doorgestuurd wordt en of het frame gewijzigd is. U moet triggers instellen voor DBUS en RBUS, zorgen dat ze klaar zijn en vervolgens proberen om het pakket in real time op te nemen. De procedures voor verschillende lijnkaarten zijn als volgt:

Zie voor meer informatie over de verschillende ELAM-procedures de links in deze tabel:

ELAM - OVERZICHT	ELAM - Overzicht - Cisco
Nexus 7K F1 module	Nexus 7000 F1 module - ELAM-procedure - Cisco
Nexus 7K F2-module	Nexus 7000 F2 module - ELAM-procedure - Cisco
Nexus 7K F3 module	F3- ELAM Voorbeeld
Nexus 7K M3-module	Nexus 7000 M-Series module - ELAM-procedure - Cisco
Nexus 7K M1/M2 en F2 module	Nexus 7K ELAM voor M1/M2 en F2 en Ethanalyzer
Nexus 7K M3-module	Nexus 7000 M3 module - ELAM-procedure - Cisco

ELAM voor Nexus 7000 - M1/M2 (Eureka-platform)

- Controleer het modulenummer met de opdracht **toon module**.
- Bevestig de module met **module x**, waarbij x het modulenummer is.
- Controleer of interne ASIC-mapping met de opdracht **hardware interne dev-port-map toont** en controleer of L2LKP en L3LKP aanwezig zijn.

```
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
+-------------------------------------------------------------------------+
+-------------------------------------------------------------------------+
```

- Eerst, vangt u het pakket in L2 en ziet of is het het door:sturen besluit juist. Om dit te doen, kijk je in de kolom L2LKP-toewijzingen en identificeer je de ASIC-instantie # die overeenkomt met de poort.
- Vervolgens voert u ELAM op deze instantie uit met de commando **elam asic eureka instantie x**waarin x het ASIC-instantienummer is en onze triggers voor DBUS en RBUS configureren. Controleer de status van de triggers met de opdrachtstatus en bevestig dat de triggers zijn geconfigureerd.

```
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
```

- Activeer de triggers met de commando **start** en controleer dat de status van de triggers met de commando **status** om te bevestigen dat de triggers gewapend zijn.

```
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
```

- Zodra de status laat zien dat de triggers gewapend zijn, zijn ze klaar om te vangen. Op dit moment moet je het verkeer doorsturen en de status opnieuw controleren om te zien of je triggers daadwerkelijk zijn geactiveerd.

```
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
```

- Controleer na activering het pakketvolgnummer voor zowel rbus als dbus om te bevestigen dat beide hetzelfde pakket hebben opgenomen. Dit kan worden gedaan met de opdracht **show dbus | i seq ; show rbus | i seq**. Als het volgnummer overeenkomt, kunt u de inhoud van de dbus en rbus bekijken. Als dit niet het geval is, voert u de opname opnieuw uit totdat u hetzelfde pakket kunt opnemen.

  Opmerking: Voor een grotere nauwkeurigheid, altijd meerdere ELAM draaien om het doorsturen van problemen te bevestigen.

- U kunt de inhoud van rbus en dbus bekijken met de opdrachten **tonen dbus** en **tonen rbus**. Het belangrijke ding in de opname is de reeks # en de bron/bestemmingsindex. Dbus toont u de bronindex die u vertelt de poort die het pakket ontvangen heeft. Rbus toont u de bestemmingsindex van de haven waaraan het pakket door:sturen aan. Daarnaast kunt u ook kijken naar de IP/MAC-adressen van de bron en de bestemming en naar VLAN-informatie.
- Met de bron- en doelindex (ook bekend als LTL-index), kunt u de bijbehorende voorpaneelpoort controleren met de opdracht **systeem interne pixelinfo ltl #**.

ELAM voor Nexus 7000 - M1/M2 (Lamira platform)

De procedure is ook voor het platform van Lamira hetzelfde, maar er zijn enkele verschillen:

- U draait ELAM met het sleutelwoord Lamira **elam asic lamira instantie x**.
- De opdrachten om de ELAM te activeren zijn:

```
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
```

- U controleert op status met de opdracht **status** en zorgt ervoor dat ze bewapend zijn voordat u verkeer verstuurt en geactiveerd nadat u het hebt opgenomen.
- U kunt dan de uitgangen van dbus interpreteren en bus tonen op dezelfde manier als getoond voor Eureka.

ELAM voor Nexus 7000 - F2/F2E (Clipper-platform)

Opnieuw, de procedure is gelijkaardig, slechts zijn de triggers verschillend. De enkele verschillen zijn:

- U voert ELAM uit met het sleutelwoord Clipper **elam asic clipper instantie x** en specificeert Layer 2 of Layer 3 modus.

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

- De opdrachten om de ELAM te activeren zijn als volgt:

```
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
```

- U controleert op status met de opdracht **status** en zorgt ervoor dat ze bewapend zijn voordat u verkeer verstuurt en geactiveerd nadat u het hebt opgenomen.
- U kunt dan de uitgangen van dbus interpreteren en bus tonen op dezelfde manier als getoond voor Eureka.

ELAM voor Nexus 7000 - F3 (Flanker Platform)

Opnieuw, de procedure is gelijkaardig, slechts zijn de triggers verschillend. De enkele verschillen zijn:

- U voert ELAM uit met het sleutelwoord Flanker **Elam asic flanker instantie x** en specificeert Layer 2 of Layer 3-modus.

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

- De opdrachten om de ELAM te activeren zijn als volgt:

```
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
```

- U controleert op status met de **status** opdracht en zorgt ervoor dat ze bewapend zijn voordat u verkeer verstuurt en geactiveerd nadat u het hebt opgenomen.
- U kunt dan de uitgangen van dbus en rbus op dezelfde manier interpreteren zoals getoond voor Eureka.

ELAM voor Nexus 9000 (platform met tahoe)

Bij Nexus 9000 is de procedure iets anders dan bij Nexus 7000. Raadpleeg voor Nexus 9000 de link [Nexus 9000 Cloud Scale ASIC (Tahoe) NX-OS ELAM - Cisco]

- Controleer eerst of interface mapping met de opdracht **hardware interne tah interface #**. De belangrijkste informatie in deze uitvoer is **ASIC #, Slice # en source ID (srcid) #.**
- Daarnaast kunt u deze informatie ook dubbel controleren met de commando **show systeem interne ethpm info interface # | i src**. Het belangrijke ding hier naast wat eerder werd vermeld is de dpid en dmod waarden.
- Controleer het modulenummer met de opdracht **toon module**.
- Bevestig de module met **module x**, waarbij x het modulenummer is.
- Start ELAM op de module met de opdrachtmodule **module-1# debug platform interne tah elam asic #**
- Configureer de binnen- of buitentrikker op basis van het type verkeer dat u wilt opnemen (L2, L3, ingesloten verkeer zoals GRE of VXLAN, enzovoort):

```
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
```

- Zodra de triggers zijn ingesteld, start ELAM met de opdracht **start**, verstuur verkeer en bekijk de uitvoer met het opdrachtrapport. De output van het rapport toont u de uitgaande en inkomende interfaces samen met het adres van VLAN-id, bron en bestemming IP/MAC.

```
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
```

ELAM voor Nexus 9000 (NorthStar-platform)

De procedure voor het NorthStar-platform is hetzelfde als het Tahoe-platform, het enige verschil is dat het trefwoord **ns** wordt gebruikt in plaats van **tah** wanneer ELAM-modus wordt ingevoerd:

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

- N9K-pakkettracer

Nexus 9000 pakkettracer tool kan worden gebruikt om het pad van het pakket te volgen en met zijn ingebouwde tellers voor stroomstatistieken maakt het een waardevol gereedschap voor intermitterende / volledige verkeersverlies scenario's. Het zou zeer nuttig zijn wanneer TCAM-bronnen beperkt zijn of niet beschikbaar zijn om andere tools te gebruiken. Bovendien, kan dit

hulpmiddel geen ARP verkeer vangen en toont geen details van de pakketinhoud zoals Wireshark.

Gebruik deze opdrachten om pakkettracer te configureren:

```
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
```

Zie voor meer informatie de link Nexus 9000: Packet Tracer tool uitgelegd - Cisco

- ## Traceroute en pings

Deze opdrachten zijn de twee meest nuttige opdrachten waarmee u snel connectiviteitsproblemen kunt identificeren.

Ping gebruikt Internet Control Message Protocol (ICMP) om ICMP-echoberichten naar de specifieke bestemming te verzenden en wacht op ICMP-echoantwoorden van die bestemming. Als het pad tussen de host werkt prima zonder problemen, kunt u de antwoorden terugkomen en pings zijn succesvol. Het ping-commando verstuurt standaard 5x ICMP echo-berichten (gelijke grootte in beide richtingen) en als alles goed werkt, kunt u 5x ICMP echo-antwoorden zien. Soms mislukt het eerste echoverzoek wanneer switches het MAC-adres leren tijdens het ARP-verzoek (Address Resolution Protocol). Als u de ping direct daarna opnieuw uitvoert, is er geen eerste ping-verlies. Bovendien kunt u met deze trefwoorden het aantal pings-, pakketgrootte-, bron-, broninterface- en time-outintervallen instellen:

```
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 wordt gebruikt om de verschillende hops te identificeren die een pakket neemt voordat het zijn 'bestemming bereikt. Het is een zeer belangrijk hulpmiddel omdat het helpt om de L3 grens te identificeren waar het falen gebeurt. U kunt de poort, bron en broninterface ook gebruiken met de volgende trefwoorden:

```
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
```

- ## PAL/RACL/VACL

ACL staat voor Access Control List. Het is een belangrijk hulpmiddel dat u toelaat om verkeer te filteren op basis van een relevant gedefinieerd criterium. Zodra ACL met ingangen voor overeenkomende criteria wordt gevuld, kan het worden toegepast om of inkomend of uitgaand verkeer te vangen. Een belangrijk aspect van ACL is zijn capaciteit om tellers voor stroomstatistieken te verstrekken. De termen PACL/RACL/VACL verwijzen naar verschillende implementaties van deze ACL's die u in staat stellen om ACL te gebruiken als een krachtig probleemoplossingsgereedschap, met name voor intermitterend verkeersverlies. Deze termen worden hier kort beschreven:

- PAL staat voor Port Access Control List: Wanneer u een toegangslijst toepast op een L2-switchpoort/interface, wordt die toegangslijst aangeduid als PACL.
- RACL staat voor Router Access Control List: Wanneer u een toegangslijst op een L3 routed poort/interface toepast, wordt die toegangslijst RACL genoemd.
- VACL staat voor VLAN Access Control List: U kunt VACL's configureren om van toepassing te zijn op alle pakketten die in of uit een VLAN worden gerouteerd of binnen een VLAN worden overbrugd. VACL's zijn strikt bedoeld voor security pakketfilters en om verkeer om te leiden naar specifieke fysieke interfaces. VACL's worden niet gedefinieerd door richting (ingress of egress).

Deze tabel biedt een vergelijking tussen de versies van ACL's.

ACL-TYPE	PACL	RACL	VACL
FUNCTIE	Filterverkeer ontvangen op een L2-interface.	Filterverkeer ontvangen op een L3-interface	Filter VLAN-verkeer
TOEGEPAST OP	- L2-interfaces/poorten. - L2 poort-kanaal interfaces. - Indien toegepast op een trunkpoort, ACL-filters verkeer op alle VLAN's toegestaan op die trunkpoort.	- VLAN-interfaces. - Fysieke L3-interfaces. - L3-subinterfaces. - L3 poort-kanaal interfaces. - Beheerinterfaces.	Zodra toegelaten wordt A toegepast op alle havens VLAN (omvat boomstamhavens).
TOEGEPASTE RICHTING	Alleen inkomend.	Inkomend of Uitgaand	-

Hier is een voorbeeld van hoe u een toegangslijst kunt configureren. Raadpleeg voor meer informatie de link naar de [Cisco Nexus 9000 Series NX-OS security configuratiegids, release 9.3(x) - IP-ACL's configureren [Cisco Nexus 9000 Series Switches] - Cisco]

```
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 is een type permanente opslag beschikbaar op Nexus-platforms als een externe compacte flitser, een USB-apparaat of een ingesloten schijf in de supervisor. Indien verwijderd uit de switch, meldt het systeem de gebruiker periodiek dat LogFlash ontbreekt. Logflash is geïnstalleerd op de supervisor en houdt historische gegevens zoals boekhoudingslogboeken, syslog berichten, debugs en Ingesloten Event Manager (EEM) outputs. EEM wordt besproken later in dit artikel. U kunt de inhoud van de LogFlash controleren met deze opdracht:

```
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
```

In het geval dat een gebruiker het apparaat opnieuw laadde of het plotseling op zijn eigen opnieuw laadde als gevolg van een gebeurtenis, zou alle loginformatie verloren gaan. In dergelijke scenario's kan LogFlash historische gegevens verstrekken die kunnen worden herzien om een waarschijnlijke oorzaak van het probleem te identificeren. Uiteraard is verder zorgvuldigheid vereist om de basisoorzaak te identificeren die u voorzien van tips over wat te zoeken in het geval deze gebeurtenis zich opnieuw voordoet.

Raadpleeg voor informatie over het installeren van logflash op het apparaat de link Nexus 7000 Logging Capabilities - Cisco.

- **OBFL**

OBFL staat voor OnBoard Failure Logging. Het is een type permanente opslag dat beschikbaar is voor zowel Nexus Top of Rack als modulaire switches. Net als de LogFlash, wordt de informatie behouden zodra het apparaat wordt herladen. OBFL slaat informatie op zoals storingen en milieugegevens. De informatie varieert voor elk platform en module, maar hier is een steekproefoutput van module 1 van Nexus 93108 platform (dat is een vast chassis met slechts één module):

```
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
```

Opnieuw, is deze informatie nuttig in het geval van een apparaat dat of opzettelijk door de gebruiker of wegens een gebeurtenis wordt herladen wordt herladen. In dit geval kan OBFL-informatie helpen om vast te stellen wat er verkeerd is gegaan vanuit het perspectief van een lijnkaart. De opdracht **toont vastlegging aan boord** is een goede plek om te beginnen. Vergeet niet dat je van binnen de module context moet vastleggen om alles wat je nodig hebt te krijgen. Zorg ervoor dat u **toont logboekregistratie aan boord module x** gebruikt of **mod x vastmaken; logboekregistratie aan boord tonen.**

- ## Geschiedenis van gebeurtenissen

Event-histories zijn een van de krachtige tools die u informatie kunnen geven over verschillende gebeurtenissen die plaatsvinden voor een proces dat loopt op Nexus. Met andere woorden, elk proces dat op een Nexus-platform loopt heeft gebeurtenisgeschiedenissen die op de achtergrond draaien en informatie opslaan over verschillende gebeurtenissen van dat proces (denk aan hen als debugs die constant lopen). Deze gebeurtenisgeschiedenissen zijn niet-persistent en alle opgeslagen informatie gaat verloren bij het opnieuw laden van het apparaat. Deze zijn zeer nuttig wanneer u een probleem met een bepaald proces hebt geïdentificeerd en dat proces zou willen problemen oplossen. Als uw OSPF-routeringsprotocol bijvoorbeeld niet goed werkt, kunt u gebeurtenisgeschiedenissen gebruiken die aan OSPF zijn gekoppeld om te bepalen waar het OSPF-proces mislukt. U kunt geschiedenissen van gebeurtenissen vinden die worden geassocieerd met bijna elk proces op het Nexus-platform zoals CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP, enzovoort.

Dit is hoe u typisch gebeurtenisgeschiedenissen voor een proces met verwijzingsvoorbeelden zou controleren. Elk proces heeft meerdere opties dus gebruik **?** om te controleren of er verschillende opties beschikbaar zijn in een proces.

```
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
```

- Debugs

Debugs zijn krachtige tools binnen NX-OS die u in staat stellen om real-time probleemoplossing gebeurtenissen uit te voeren en ze te registreren naar een bestand of weergave in CLI. Het is sterk aanbevolen om de debug-uitgangen op een bestand te registreren omdat ze invloed hebben op de CPU-prestaties. Gebruik voorzichtigheid voordat u een debug direct op de CLI uitvoert.

Debugs worden meestal alleen uitgevoerd wanneer u een probleem hebt geïdentificeerd om één enkel proces te zijn en wil controleren hoe dit proces zich in real-time gedraagt met echt verkeer in het netwerk. U moet een debug-functie inschakelen op basis van de gedefinieerde gebruikersaccountrechten.

Net als gebeurtenishistoriën kunt u debugs uitvoeren voor elk proces op een Nexus-apparaat zoals CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP enzovoort.

Dit is hoe u typisch debug voor een proces zou in werking stellen. Elk proces heeft meerdere opties dus gebruik **?** om te controleren of er verschillende opties beschikbaar zijn in een proces.

```
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
```

- GOUD

GOLD staat voor Generic OnLine Diagnostics. Zoals de naam al aangeeft, worden deze tests over het algemeen gebruikt als systeemgezondheidscontrole en worden ze gebruikt om de hardware in kwestie te controleren of te verifiëren. Er zijn verschillende online tests die worden uitgevoerd en gebaseerd op het platform in gebruik, sommige van deze tests verstoren terwijl sommige niet verstorend zijn. Deze onlinetests kunnen als volgt worden gecategoriseerd:

- **Opstartdiagnostiek:** Deze tests zijn de tests die worden uitgevoerd wanneer het apparaat opstart. Zij controleren ook op connectiviteit tussen de supervisor en de modules, die connectiviteit tussen gegevens en controlevliegtuig voor alle ASIC's omvat. Tests zoals ManagementPortLoopback en EOBLoopback zijn verstorend, terwijl de testen voor OBFL en USB niet storend zijn.
- **Run-time of Health Monitoring Diagnostics:** Deze tests leveren informatie op over de gezondheid van het apparaat. Deze tests zijn niet storend en worden op de achtergrond uitgevoerd om de stabiliteit van de hardware te waarborgen. U kunt deze tests in- en uitschakelen wanneer dit nodig is of voor probleemoplossing.
- **On-demand diagnostiek:** Alle genoemde tests kunnen op aanvraag worden herhaald om een probleem te lokaliseren.

U kunt met deze opdracht controleren op de verschillende soorten online tests die voor uw switch beschikbaar zijn:

```
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-
```

Om weer te geven wat elk van de 17 genoemde tests doet, kunt u deze opdracht gebruiken:

```
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 staat voor Embedded Event Manager. Het is een krachtig hulpmiddel dat u toestaat om uw apparaat te programmeren om specifieke taken uit te voeren in het geval dat een bepaalde gebeurtenis gebeurt. Het controleert verschillende gebeurtenissen op het apparaat en neemt dan noodzakelijke actie om het probleem op te lossen en mogelijk te herstellen. EEM bestaat uit drie hoofdcomponenten, die hier kort worden beschreven:

- **Event Statement:** Dit zijn de gebeurtenissen die u wilt controleren en willen dat Nexus een bepaalde actie uitvoert zoals het doen van een tijdelijke oplossing of eenvoudig een SNMP-server op de hoogte stellen of een CLI-logboek weergeven, enzovoort.
- **Actieverklaringen:** Dit zouden de stappen zijn die EEM zou nemen zodra een gebeurtenis wordt geactiveerd. Deze acties zouden eenvoudig kunnen zijn om een interface uit te schakelen of sommige showbevelen en kopieeroutput uit te voeren naar een bestand op FTP server, een e-mail te versturen, etc.
- **Beleid:** Het is in principe een gebeurtenis in combinatie met een of meerdere actieverklaringen die u op de supervisor via CLI of een bash script kunt configureren. Je kunt EEM ook aanhalen met een python script. Zodra het beleid is gedefinieerd over de toezichthouder, duwt het het beleid naar de relevante module.

Raadpleeg voor meer informatie over EEM de link Cisco Nexus 9000 Series NX-OS Systeembeheerconfiguratiehandleiding, release 9.2(x) - Ingesloten Event Manager [Cisco Nexus 9000 Series Switches] - Cisco configureren.

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.