

# Utilisation de vemlog pour déboguer les problèmes de Nexus 1000v

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Exécution des commandes vemlog](#)

[Niveaux de débogage du journal virtuel](#)

[Étape 1 : Effacement des paramètres de capture actuels](#)

[Étape 2 : Définition des paramètres de capture du journal virtuel](#)

[Étape 3 : Affichage du résultat](#)

[Étape 4 : Restauration des niveaux de journalisation normaux du journal virtuel](#)

[Informations connexes](#)

## Introduction

La partie Virtual Ethernet Module (VEM) du Nexus 1000v comporte des commandes **de débogage** intégrées que vous pouvez utiliser pour faciliter le dépannage. Ces commandes vemlog vous permettent d'avoir une idée de ce que font certains processus du VEM, des commandes qu'ils envoient et du type de réponse qu'ils reçoivent. Par exemple, vous pouvez les utiliser dans les situations suivantes :

- Un canal de port LACP ne monte pas entre le commutateur Nexus 1000v et le commutateur en amont. Vous utiliseriez une combinaison de “ ” vemlog sur l'hôte et “ ” de débogage sur le commutateur en amont pour résoudre le problème.
- QoS ne fonctionne pas correctement et vous voulez vérifier que le VEM définit correctement les valeurs.
- Vous voulez voir les modifications apportées à la communication VSM-VEM.

Le cas d'utilisation le plus courant pour exécuter ces commandes est le débogage de LACP. En utilisant les modules debug « sflACP » et « debug lACP » sur le commutateur en amont, vous pouvez détecter quel périphérique est défectueux, ou si la liaison entre les deux périphériques est en train de supprimer les BPDU LACP.

Les niveaux par défaut ont généralement assez d'informations pour que le journal puisse être collecté et analysé sans activer de débogages supplémentaires.

## Conditions préalables

## Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

## Components Used

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

## Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

## Exécution des commandes vemlog

Si vous avez des questions sur la syntaxe de ces commandes, vous pouvez taper "vemlog" dans l'interface de ligne de commande ESXi et recevoir une sortie d'aide. Aucun onglet n'est terminé ou « ? » pour ces commandes. Vous devez les taper complètement ou ils ne fonctionneront pas correctement. Pour exécuter les commandes **vemlog**, vous devez utiliser l'une des commandes suivantes :

- Accès par console SSH/local à l'hôte ESXi en question. L'accès SSH est préférable car vous pouvez facilement faire défiler vers le haut.
- L'accès à la console SSH/locale au VSM et le module doit être actif dans le VSM. Encore une fois, l'accès SSH est préférable. Si vous utilisez cette méthode, vous devez préfacer chaque commande avec **module vem # execute**, où # est le numéro de module du VEM.

## Niveaux de débogage du journal virtuel

Vemlog sur le VEM est similaire au Syslog sur un commutateur normal. Comme un commutateur normal, il existe des niveaux de gravité :

e - Error  
n - Notification  
w - Warning  
i - Information  
d - Debug  
p - Print  
t - Temporary

Cette valeur est classée de la plus grave au moins (les messages les moins sévères étant beaucoup plus nombreux). Par défaut, le vemlog est défini sur ENW (Erreur, Notification, Avertissement).

## Étape 1 : Effacement des paramètres de capture actuels

Pour résoudre un problème particulier, il est utile de désactiver les captures par défaut.

```
vemlog stop  
vemlog clear  
vemlog debug all none
```

Ces commandes arrêtent toutes les commandes **vemlog** actuelles qui se produisent et réinitialisent les paramètres de capture vierges. Vous pouvez le vérifier à l'aide de la commande suivante :

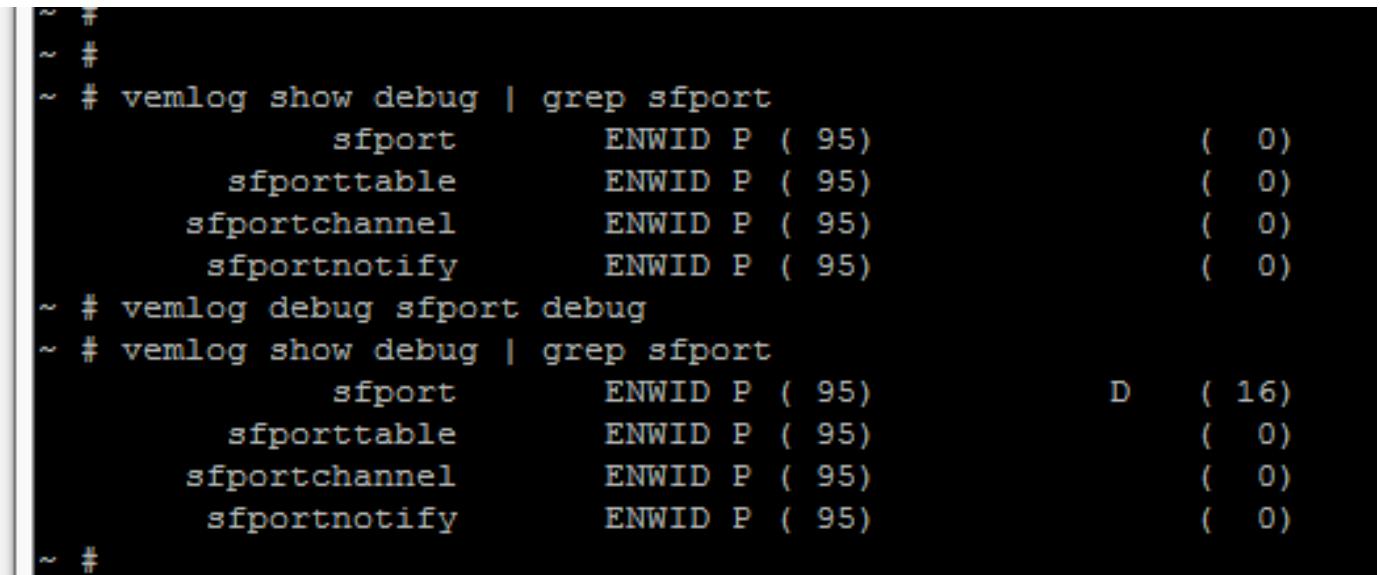


Module	Available	Printing
all	ENWID P ( 95)	( 0)
sfheapshow	ENWID P ( 95)	( 0)
sfheap	ENWID P ( 95)	( 0)
sfmempoolshow	ENWID P ( 95)	( 0)
sfmempool	ENWID P ( 95)	( 0)
if_bridge_rt	ENWID P ( 95)	( 0)
sfint32list	ENWID P ( 95)	( 0)
sflist	ENWID P ( 95)	( 0)
cbridge_proc	ENWID P ( 95)	( 0)
vssnet	ENWID P ( 95)	( 0)
sfpacket	ENWID P ( 95)	( 0)
sfbase	ENWID P ( 95)	( 0)
sfscchedlog	ENWID P ( 95)	( 0)
sfscched	ENWID P ( 95)	( 0)
sfbd	ENWID P ( 95)	( 0)

Comme vous le voyez ici, la commande **vemlog show debug** affiche les paramètres de débogage actuels. La valeur par défaut de tous les modules pour “ Impression ” est 0 ou non.

## Étape 2 : Définition des paramètres de capture du journal virtuel

Maintenant que vous avez une ardoise propre à commencer, il est temps de configurer les cookies pour la capture. Pour cet exemple, les journaux de port généraux sont capturés :



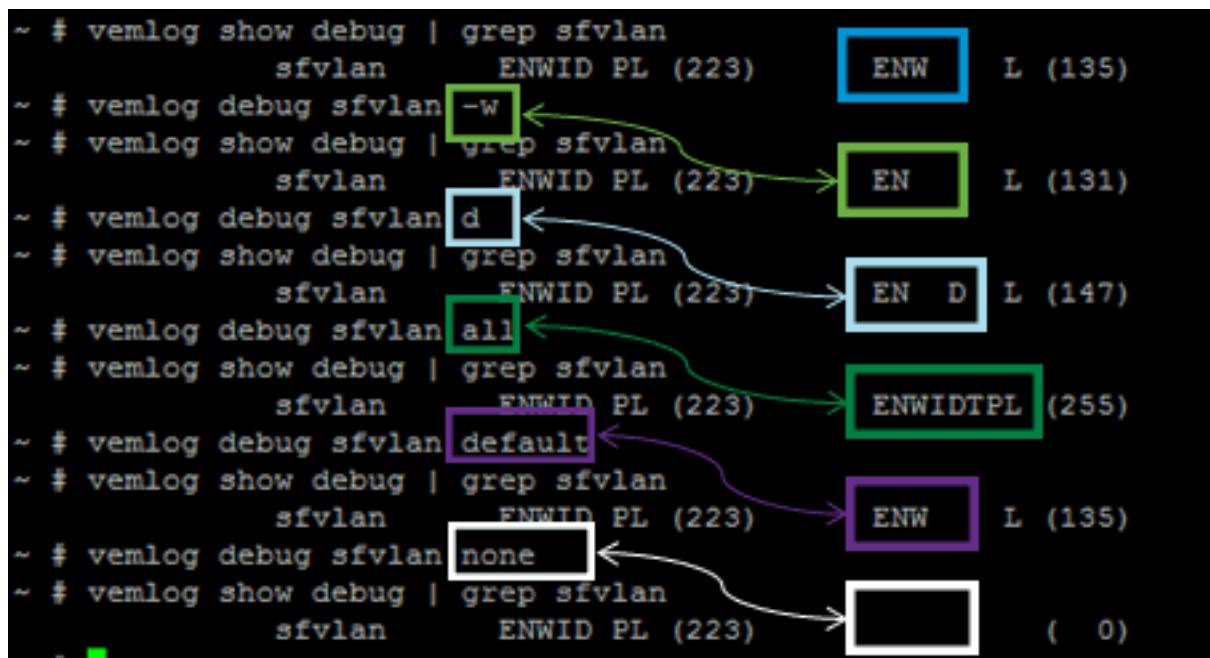
```
~ #  
~ #  
~ # vemlog show debug | grep sfport  
      sfport      ENWID P ( 95)          ( 0)  
      sfporttable ENWID P ( 95)          ( 0)  
      sfportchannel ENWID P ( 95)         ( 0)  
      sfportnotify ENWID P ( 95)          ( 0)  
~ # vemlog debug sfport debug  
~ # vemlog show debug | grep sfport  
      sfport      ENWID P ( 95)          D  ( 16)  
      sfporttable ENWID P ( 95)          ( 0)  
      sfportchannel ENWID P ( 95)         ( 0)  
      sfportnotify ENWID P ( 95)          ( 0)  
~ #
```

La deuxième commande présente la syntaxe suivante :

```
vemlog debug [module|all] [-][all|none|default|e|w|n|i|d|p|t]
```

Où “debug\_module” est le paquet vemlog que vous voulez capturer. Généralement, “ ” de débogage est suffisant pour le dépannage.

Si vous ajoutez [-] à la commande, le niveau de débogage sera supprimé pour ce module. Cette capture d'écran présente des exemples de ceci :



```
~ # vemlog show debug | grep sfvlan
      sfvlan      ENWID PL (223)      ENW      L (135)
~ # vemlog debug sfvlan -w
~ # vemlog show debug | grep sfvlan
      sfvlan      ENWID PL (223)      EN      L (131)
~ # vemlog debug sfvlan d
~ # vemlog show debug | grep sfvlan
      sfvlan      ENWID PL (223)      EN D    L (147)
~ # vemlog debug sfvlan all
~ # vemlog show debug | grep sfvlan
      sfvlan      ENWID PL (223)      ENWIDTPL (255)
~ # vemlog debug sfvlan default
~ # vemlog show debug | grep sfvlan
      sfvlan      ENWID PL (223)      ENW      L (135)
~ # vemlog debug sfvlan none
~ # vemlog show debug | grep sfvlan
      sfvlan      ENWID PL (223)      ( 0)
```

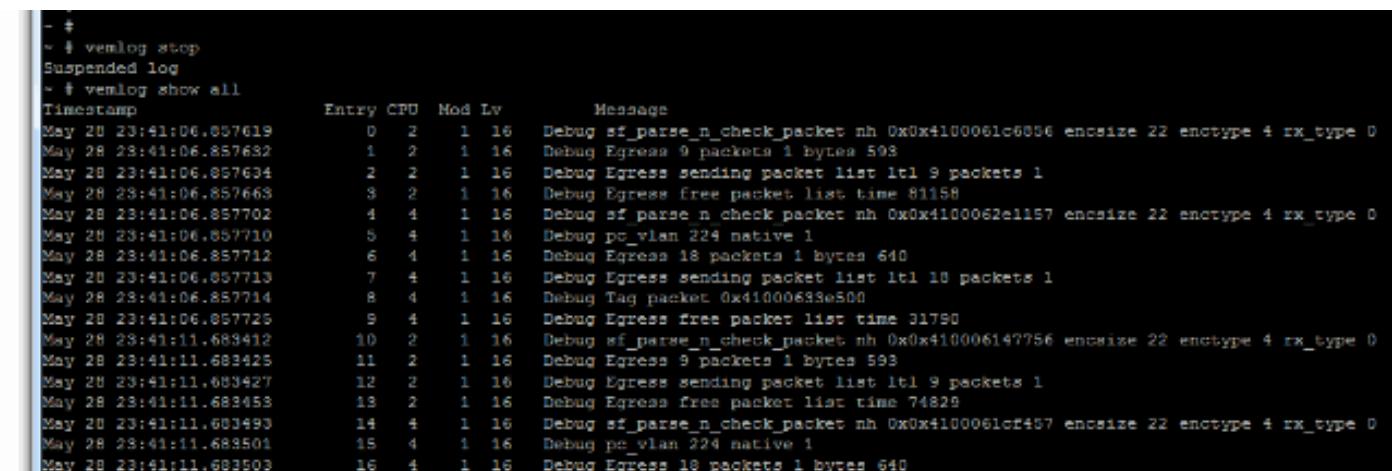
Vous pouvez afficher tous les modules de débogage disponibles avec une commande **vemlog show debug**.

Une fois que vous avez spécifié les packages à surveiller, démarrez le vemlog avec une commande **vemlog start** :



```
~ #
~ # vemlog start
Started log
~ #
```

## Étape 3 : Affichage du résultat



```
~ #
~ # vemlog stop
Suspended log
~ # vemlog show all
Timestamp      Entry CPU  Mod Lv      Message
May 20 23:41:06.857619      0  2    1  16  Debug sf_parse_n_check_packet nh 0x0x4100061c6056 encsize 22 enctype 4 rx_type 0
May 20 23:41:06.857632      1  2    1  16  Debug Egress 9 packets 1 bytes 598
May 20 23:41:06.857634      2  2    1  16  Debug Egress sending packet list ltl1 9 packets 1
May 20 23:41:06.857663      3  2    1  16  Debug Egress free packet list time 81158
May 20 23:41:06.857702      4  4    1  16  Debug sf_parse_n_check_packet nh 0x0x4100062e1157 encsize 22 enctype 4 rx_type 0
May 20 23:41:06.857710      5  4    1  16  Debug pc_vlan 224 native 1
May 20 23:41:06.857712      6  4    1  16  Debug Egress 18 packets 1 bytes 640
May 20 23:41:06.857713      7  4    1  16  Debug Egress sending packet list ltl1 18 packets 1
May 20 23:41:06.857714      8  4    1  16  Debug Tag packet 0x41000633e500
May 20 23:41:06.857725      9  4    1  16  Debug Egress free packet list time 31790
May 20 23:41:11.683412     10  2    1  16  Debug sf_parse_n_check_packet nh 0x0x410006147756 encsize 22 enctype 4 rx_type 0
May 20 23:41:11.683425     11  2    1  16  Debug Egress 9 packets 1 bytes 598
May 20 23:41:11.683427     12  2    1  16  Debug Egress sending packet list ltl1 9 packets 1
May 20 23:41:11.683453     13  2    1  16  Debug Egress free packet list time 74829
May 20 23:41:11.683493     14  4    1  16  Debug sf_parse_n_check_packet nh 0x0x4100061cf457 encsize 22 enctype 4 rx_type 0
May 20 23:41:11.683501     15  4    1  16  Debug pc_vlan 224 native 1
May 20 23:41:11.683503     16  4    1  16  Debug Egress 18 packets 1 bytes 640
```

Une fois que vous êtes prêt à afficher le journal, émettez une commande **vemlog show all**. Cela imprime toutes les entrées du journal sur la console (ou la session SSH). Si vous êtes directement connecté à l'hôte, vous pouvez également l'envoyer à un fichier afin de pouvoir le copier et l'analyser ultérieurement :

```
~ #  
~ # vemlog show all > /tmp/vemlog  
~ # ls /tmp  
dpafifo  scratch  vemlog  
~ #
```

Si vous déplacez ce fichier vers le répertoire `/vmfs/volumes/[datastore_name]`, vous pouvez le déplacer de l'hôte à l'aide du navigateur de data store dans ESXi.

Le vemlog est également stocké dans le journal vmkernel.

## Étape 4 : Restauration des niveaux de journalisation normaux du journal virtuel

```
~ # vemlog stop  
Suspended log  
~ # vemlog clear  
Cleared log  
~ # vemlog debug all default  
~ # vemlog start  
Started log
```

Ces commandes restaureront la configuration standard du journal virtuel. Ceci est important pour s'assurer que les informations de débogage sont capturées correctement pour les problèmes futurs.

Comme le vemlog est stocké à la fois dans les journaux vmkernel et dans la mémoire tampon de vemlog, un débogage particulièrement chatty peut inonder ces deux fichiers d'informations inutiles, surtout s'ils sont laissés activés.

## Informations connexes

- [Support et documentation techniques - Cisco Systems](#)