

# Usando el vemlog para hacer el debug de los problemas del nexa 1000v

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Comandos corrientes del vemlog](#)

[Niveles de debugging de Vemlog](#)

[Paso 1: Borrar las configuraciones actuales de la captura](#)

[Paso 2: Determinación de los parámetros de la captura del vemlog](#)

[Paso 3: Ver la salida](#)

[Paso 4: Restablecer el vemlog a los niveles de registro normales](#)

[Información Relacionada](#)

## [Introducción](#)

La porción virtual del módulo Ethernet (VEM) del nexa 1000v tiene los **comandos debug** incorporados que usted puede utilizar para ayudar al troubleshooting. Estos comandos del vemlog no le prohíben la penetración a qué ciertos procesos del VEM están haciendo, a qué comandos están enviando, y qué clase de respuesta están recibiendo. Por ejemplo, usted puede ser que los utilice en estas situaciones:

- Un canal del puerto LACP no está subiendo entre el nexa 1000v y el Switch por aguas arriba. Usted utilizaría una combinación de “vemlog” en el host y el “debug” en el Switch por aguas arriba para estrechar abajo el problema.
- QoS no está trabajando correctamente y usted quiere verificar que el VEM está fijando los valores correctamente.
- Usted quiere ver los cambios a la comunicación VSM-VEM.

El caso más de uso común para funcionar con estos comandos está haciendo el debug del LACP. Usando el módulo “sflacp” y “lacp del debug del debug” en el Switch por aguas arriba usted puede detectar está funcionando incorrectamente qué dispositivo, o si el link entre los dos dispositivos está cayendo el LACP BPDU.

Los niveles predeterminados tienen generalmente bastante información así que el registro puede ser recogido y ser analizado sin girar los debugs adicionales.

## [prerrequisitos](#)

## Requisitos

No hay requisitos específicos para este documento.

## Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

## Convenciones

Consulte [Convenciones de Consejos Técnicos Cisco](#) para obtener más información sobre las convenciones del documento.

## Comandos corrientes del vemlog

Si usted tiene cualesquiera preguntas sobre el sintaxis de estos comandos usted puede teclear el “vemlog” en el ESXi CLI y recibir una salida de la ayuda. No hay lengüeta completa o “?” para estos comandos. Usted debe escribirlos a máquina completamente o no trabajarán correctamente. Para ejecutar el **vemlog** le ordena necesitar uno del siguiente:

- Acceso a la consola SSH/local al host de ESXi en la pregunta. El acceso SSH es preferible porque usted puede navegar fácilmente la salvaguardia.
- El acceso a la consola SSH/local al VS y al módulo tiene que ser activo en el VS. El acceso de SSH es de nuevo preferible. Si usted está utilizando este método que usted necesitará introducir cada comando con el **vem del módulo # ejecuta**, donde # está el número de módulo del VEM.

## Niveles de debugging de Vemlog

Vemlog en el VEM es similar al Syslog en un Switch normal. Como un Switch normal, hay niveles de gravedad:

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

Esto se alinea de la más severo a lo menos (con los mensajes menos severos haciendo salir lejos más información). Por abandono el vemlog se fija a ENW (error, notificación, advirtiendo).

## Paso 1: Borrar las configuraciones actuales de la captura

Para resolver problemas un problema determinado tiene valor inhabilitar las capturas predeterminadas.

```
vemlog stop
```

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

Estos comandos pararán cualquier comando actual del **vemlog** que esté sucediendo, y reajustarán el vemlog para esconder las configuraciones de la captura. Usted puede verificar esto con el siguiente comando:

```
~ #
~ #
~ # vemlog show debug
      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)
      sfschedlog            ENWID P ( 95)                ( 0)
      sfsched                ENWID P ( 95)                ( 0)
      sfbid                  ENWID P ( 95)                ( 0)
```

Como usted ve aquí, el **comando show debug del vemlog** visualizará las configuraciones actuales del debug. El valor por defecto para todos los módulos para “imprimir” es 0, o impresión.

## [Paso 2: Determinación de los parámetros de la captura del vemlog](#)

Ahora que usted tiene una cuenta nueva a comenzar con, es hora de fijar los Cookie para capturar. Por este ejemplo, se capturan los registros generales del puerto:

```
~ #
~ #
~ # 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)
~ #
```

El comando second allí tiene este sintaxis:

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

Donde está el paquete el “debug\_module” del vemlog usted quiere capturar. El “debug” es generalmente bastante para resolver problemas.

Si usted agrega el [-]al comando que el nivel de debug será quitado para ese módulo. Este tiro de pantalla tiene ejemplos de esto:

```
~ # 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)
```

Usted puede ver todos los módulos disponibles del debug con un comando show debug del vemlog.

Una vez que usted ha especificado qué paquetes para monitorear, después comienza el vemlog con un comando start del vemlog:

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

### Paso 3: Ver la salida

```
~ #
~ # vemlog stop
Suspended log
~ # vemlog show all
Timestamp      Entry CPU  Mod Lv      Message
May 28 23:41:06.857619 0 2 1 16 Debug sf_parse_n_check_packet nh 0x0x4100061c6856 encsize 22 enctype 4 rx_type 0
May 28 23:41:06.857632 1 2 1 16 Debug Egress 9 packets 1 bytes 593
May 28 23:41:06.857634 2 2 1 16 Debug Egress sending packet list ltl 9 packets 1
May 28 23:41:06.857663 3 2 1 16 Debug Egress free packet list time 81158
May 28 23:41:06.857702 4 4 1 16 Debug sf_parse_n_check_packet nh 0x0x4100062e1157 encsize 22 enctype 4 rx_type 0
May 28 23:41:06.857710 5 4 1 16 Debug pc_vlan 224 native 1
May 28 23:41:06.857712 6 4 1 16 Debug Egress 18 packets 1 bytes 640
May 28 23:41:06.857713 7 4 1 16 Debug Egress sending packet list ltl 18 packets 1
May 28 23:41:06.857714 8 4 1 16 Debug Tag packet 0x41000633e500
May 28 23:41:06.857725 9 4 1 16 Debug Egress free packet list time 31750
May 28 23:41:11.683412 10 2 1 16 Debug sf_parse_n_check_packet nh 0x0x410006147756 encsize 22 enctype 4 rx_type 0
May 28 23:41:11.683425 11 2 1 16 Debug Egress 9 packets 1 bytes 593
May 28 23:41:11.683427 12 2 1 16 Debug Egress sending packet list ltl 9 packets 1
May 28 23:41:11.683453 13 2 1 16 Debug Egress free packet list time 74829
May 28 23:41:11.683493 14 4 1 16 Debug sf_parse_n_check_packet nh 0x0x4100061cf457 encsize 22 enctype 4 rx_type 0
May 28 23:41:11.683501 15 4 1 16 Debug pc_vlan 224 native 1
May 28 23:41:11.683503 16 4 1 16 Debug Egress 18 packets 1 bytes 640
```

Una vez que usted está listo para ver el registro, publique un **comando all de la demostración del vemlog**. Esto imprime todas las entradas de registro a la consola (o a la sesión SSH). Si usted está conectado directamente con el host usted puede también enviar esto a un archivo de modo que usted pueda copiarlo apagado y analizarlo más adelante:

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

Si usted mueve este archivo al directorio del [datastore\_name] de /vmfs/volumes/ usted puede moverlo del host que usa al navegador del datastore en ESXi.

El vemlog también se salva en el registro del vmkernel.

## [Paso 4: Restablecer el vemlog a los niveles de registro normales](#)

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

Estos comandos restablecerán el vemlog a la configuración estándar. Esto es importante asegurarse de que la información de debugging está capturada correctamente por problemas futuros.

Porque el vemlog se salva en los registros del vmkernel y el buffer de los vemlog, un debug determinado hablador puede inundar ambos con la información innecesaria, especialmente si se dejan habilitados.

## [Información Relacionada](#)

- [Soporte Técnico y Documentación - Cisco Systems](#)