

# Utilisant LIBfc avec la carte d'interface virtuelle de VMware et de Cisco pour dépanner le demandeur/visez la transmission

## Contenu

[Introduction](#)

[Configuration prise en charge](#)

[Identifier la configuration actuelle](#)

[Configuration debug\\_logging de LIBfc de modification](#)

[Modification LIBfc debug\\_logging de nouveau à la configuration d'origine :](#)

## Introduction

Ce document décrit comment utiliser le libfc masqué met au point pour gagner la visibilité inférieure dans le processus de la procédure de connexion de port (PLOGI) utilisé dans la transmission de la Manche de fibre (FC) dans ESXi. Par activant debug\_logging nous pouvons voir les informations de l'adaptateur de réseau convergent (POUVEZ) sur les trames étendues du service de lien (ELS) telles que la procédure de connexion de matrice (FLOGI), la procédure de connexion de port (PLOGI), cette nous normalement ne pourrions pas voir. Ceci peut être utile s'il n'y a pas un Finisar à portée de la main ou une ENVERGURE et vous veulent assurer ce qui l'hôte is/is ne se terminant pas dans la pile FC.

Contribué par Brian Hopkins, ingénieur TAC Cisco.

## Configuration prise en charge

Actuellement ceci est seulement pris en charge sur ESX avec une carte d'interface virtuelle de Cisco (carte d'interface virtuelle), d'autres adaptateurs dans la mesure où je sais ne prends en charge pas cette caractéristique.

## Identifier la configuration actuelle

Vous pouvez utiliser la commande suivante sur l'hôte d'ESXi de s'assurer que cette valeur n'est pas déjà placée :

From the CLI of ESXi:

```
esxcli system module parameters list -m libfc_92
```

```
esxcli system module parameters list -m libfcoe_92
```

La sortie devrait ressembler au suivant, avis comment la valeur n'est pas configurée pour **debug\_logging**, qui est la valeur que nous changerons dans les étapes suivantes.

```

~ # cat /var/log/vmkernel.log | grep <6>
~ # esxcli system module parameters list -m libfc_92
Name          Type  Value  Description
-----
debug_logging  int   a bit mask of logging levels
heap_initial   int   Initial heap size allocated for the driver.
heap_max       int   Maximum attainable heap size for the driver.
min_exch_pool_elem int   Minimum number of elements guaranteed to be allocated for exchange pool.
rec_tov       int   REC timeout value
skb_mpool_initial int   Driver's minimum private socket buffer memory pool size.
skb_mpool_max  int   Maximum attainable private socket buffer memory pool size for the driver.
~ # esxcli system module parameters list -m libfc0e_92
Name          Type  Value  Description
-----
debug_logging  int   a bit mask of logging levels
heap_initial   int   Initial heap size allocated for the driver.
heap_max       int   Maximum attainable heap size for the driver.
skb_mpool_initial int   Driver's minimum private socket buffer memory pool size.
skb_mpool_max  int   Maximum attainable private socket buffer memory pool size for the driver.
~ # _

```

## Configuration debug\_logging de LIBfc de modification

Afin d'obtenir les informations complémentaires pour révéler dans le fichier de /var/log/vmkernel.log sur ESXi que nous devons activer debug\_logging et devons redémarrer l'hôte :

```
esxcli system module parameters set -p debug_logging=0xf -m libfc_92
```

```
esxcli system module parameters set -p debug_logging=0xf -m libfc0e_92
```

Après que vous entriez ceci vous commande peut vérifier de nouveau pour s'assurer que la valeur est maintenant placée à 0xf :

```

~ # esxcli system module parameters set -p debug_logging=0xf -m libfc_92
~ # esxcli system module parameters set -p debug_logging=0xf -m libfc0e_92
~ # esxcli system module parameters list -m libfc0e_92
Name          Type  Value  Description
-----
debug_logging  int   0xf    a bit mask of logging levels
heap_initial   int   Initial heap size allocated for the driver.
heap_max       int   Maximum attainable heap size for the driver.
skb_mpool_initial int   Driver's minimum private socket buffer memory pool size.
skb_mpool_max  int   Maximum attainable private socket buffer memory pool size for the driver.
~ # esxcli system module parameters list -m libfc_92
Name          Type  Value  Description
-----
debug_logging  int   0xf    a bit mask of logging levels
heap_initial   int   Initial heap size allocated for the driver.
heap_max       int   Maximum attainable heap size for the driver.
min_exch_pool_elem int   Minimum number of elements guaranteed to be allocated for exchange pool.
rec_tov       int   REC timeout value
skb_mpool_initial int   Driver's minimum private socket buffer memory pool size.
skb_mpool_max  int   Maximum attainable private socket buffer memory pool size for the driver.

```

Nous ne sommes toujours pas terminés, vous ne verrez pas les nouveaux logs apparaître jusqu'à ce que vous **redémarrerez l'hôte d'ESXi**. Après que vous ayez redémarré l'hôte d'ESXi vous pouvez vous vérifier voyez que ces nouvelles données mises à jour dans le fichier de vmkernel.log en exécutant le suivant commandez :

```
cat /var/log/vmkernel.log | grep "<6>"
```

Puisque toutes les commandes ont cette en-tête <6> qu'il les rend faciles à trouver, j'ai inclus un bout ci-dessous de ces nouvelles informations utiles affichant les états FLOGI et PLOGI :

```

2016-04-01T16:12:39.672Z cpu21:8803<6>fnic : 3 :: vNIC flags 0x8 luns per tgt 256
2016-04-01T16:12:39.672Z cpu21:8803<6>fnic : 3 :: vNIC flogi_retries 8 flogi timeout 4000
2016-04-01T16:12:39.672Z cpu21:8803<6>fnic : 3 :: vNIC plogi_retries 8 plogi timeout 20000
2016-04-01T16:12:39.672Z cpu21:8803<6>fnic : 3 :: vNIC io throttle count 16 link dn timeout 30000
2016-04-01T16:12:39.672Z cpu21:8803<6>fnic : 3 :: vNIC port dn io retries 30 port dn timeout 30000
2016-04-01T16:12:39.673Z cpu21:8803<6>fnic : 3 :: vNIC interrupt mode: MSI-X
2016-04-01T16:12:39.673Z cpu21:8803<6>fnic : 3 :: vNIC resources avail: wq 2 cp_wq 1 raw_wq 1 rq 1 cq 3 intr 4
2016-04-01T16:12:39.673Z cpu21:8803<6>fnic : 3 :: firmware uses non-FIP mode
2016-04-01T16:12:39.680Z cpu21:8803<6>host3: lport ffffffff: Entered RESET state from reset state
<6>Broadcom NetXtreme II CNIC Driver cnic v1.74.04.v50.1 (September 11, 2012)
<6>bnx2fc: Broadcom NetXtreme II FCoE Driver bnx2fc v1.74.02.v50.2 (Aug 28, 2012)
2016-04-01T16:12:40.341Z cpu1:8761<6>host2: libfc: Link up on port ( 0)
2016-04-01T16:12:40.341Z cpu1:8761<6>host2: lport 0: Entered FLOGI state from reset state
2016-04-01T16:12:40.354Z cpu2:8763<6>host2: lport 0: Received a FLOGI accept
2016-04-01T16:12:40.354Z cpu2:8763<6>host2: Assigned Port ID 10003
2016-04-01T16:12:40.354Z cpu2:8763<6>host2: fip: received FLOGI LS_ACC using non-FIP mode
2016-04-01T16:12:40.354Z cpu2:8763<6>host2: lport 10003: Entered DNS state from FLOGI state
2016-04-01T16:12:40.354Z cpu2:8763<6>host2: rport fffffc: Login to port
2016-04-01T16:12:40.354Z cpu2:8763<6>host2: rport fffffc: Port entered PLOGI state from Init state
2016-04-01T16:12:40.356Z cpu18:8733<6>host2: rport fffffc: Received a PLOGI accept
2016-04-01T16:12:40.357Z cpu18:8733<6>host2: rport fffffc: Port is Ready
2016-04-01T16:12:40.357Z cpu18:8733<6>host2: rport fffffc: work event 1
2016-04-01T16:12:40.357Z cpu18:8733<6>host2: rport fffffc: callback ev 1
2016-04-01T16:12:40.357Z cpu18:8733<6>host2: lport 10003: Received a 1 event for port (fffffc)

```

## Modification LIBfc debug\_logging de nouveau à la configuration d'origine :

Vous pouvez changer ceci de nouveau au par défaut en insérant les 2 commandes ci-dessous et en redémarrant l'hôte d'ESXi. Nous mettons fondamentalement juste la modification à zéro d'avant pour placer ceci de nouveau au par défaut :

```
esxcli system module parameters set -p debug_logging= -m libfc_92
```

```
esxcli system module parameters set -p debug_logging= -m libfcoc_92
```

Vous pouvez exécuter les mêmes commandes de s'assurer de nouveau que la modification est réussie :

From the CLI of ESXi:

```
esxcli system module parameters list -m libfc_92
```

```
esxcli system module parameters list -m libfcoc_92
```

Ils devraient chacun des deux ressembler à ce qui suit :

```

~ # esxcli system module parameters list -m libfc_92
Name                Type  Value  Description
-----
debug_logging       int   a bit mask of logging levels
heap_initial        int   Initial heap size allocated for the driver.
heap_max            int   Maximum attainable heap size for the driver.
min_exch_pool_elem int   Minimum number of elements guaranteed to be allocated for exchange pool.
rec_tov             int   REC timeout value
skb_mpool_initial   int   Driver's minimum private socket buffer memory pool size.
skb_mpool_max       int   Maximum attainable private socket buffer memory pool size for the driver.
~ # esxcli system module parameters list -m libfcoc_92
Name                Type  Value  Description
-----
debug_logging       int   a bit mask of logging levels
heap_initial        int   Initial heap size allocated for the driver.
heap_max            int   Maximum attainable heap size for the driver.
skb_mpool_initial   int   Driver's minimum private socket buffer memory pool size.
skb_mpool_max       int   Maximum attainable private socket buffer memory pool size for the driver.
~ # _

```

Après réinitialisation de l'hôte ESX vous pouvez s'assurer que l'élimination des imperfections est entrée dans le log en vérifiant avec cette commande :

```
tail /var/log/vmkernel.log | grep "<6>"
```