|  |  |
| --- | --- |
|  |  |

Collection Guide for CSR-3342729 - FFM-ECAB010 Terminating Invite Issue EVIP and Linux

# Abstract

Troubleshooting commands for dropped outgoing packets when evip or linux is the suspicious point.

# pre-condition

Make sure that the packet is not captured on external network.

Confirm that the packet is sent out from sip transport;

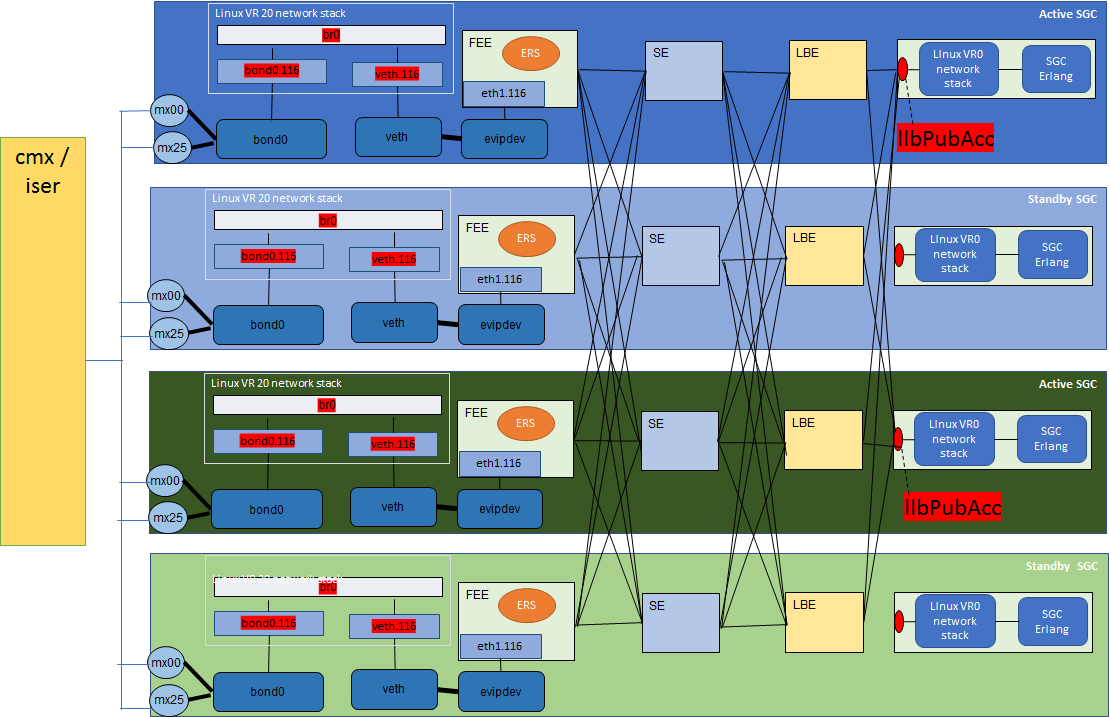
# Basic Check

blade\_0\_11:~# uname -a

blade\_0\_11:~# /opt/vip/bin/evipc -v

check if ip fragmentation is allowed in firewall profile.

# tcpdump



The picture shows IS network; VSBG doesn’t have bond mechanism.

## Starting tcpdump

In each terminal / putty-session that handles one blade each for evip traffic. (If multiple interfaces are captured, multiple terminals / putty-sessions are needed towards the same blade, for simultaneous captures)

* Capture will be done towards /tmp on SGC. Run this command to check how much available diskspace is left. If it’s more then 2GB it will be ok.

df -h /tmp

* Find the interface owning the VIP address with “ifconfig”, like llbPubAcc in the example.

The commands to be run on the blades. We suggest you prepare the commands first, with different filenames for each blade, and keep it ready in the terminal / putty-session. Then trigger all of them as close to each other as possible.

**Linux VR XX (all SGC):**

vrctl -c xx tcpdump -i bond0.131 -vvv -s 0 'udp and host 217.0.21.64’ -c 1000000 -w /tmp/bond0\_capture\_blade\_X.pcap

vrctl -c xx tcpdump -i veth.131 -vvv -s 0 'udp and host 217.0.21.64’ -c 1000000 -w /tmp/veth\_capture\_blade\_X.pcap

vrctl -c xx tcpdump -i br0 -vvv -s 0 'udp and host 217.0.21.64’ -c 1000000 -w /tmp/br0\_capture\_blade\_X.pcap

Note, change the bond0.131 to eth1.131 on VSBG; change the VIP address; change the UDP;

**LLB (active SGC):**

vrctl -c 0 tcpdump -i llbPubAcc -vvv -s 0 'udp and host 217.0.21.64’ -c 1000000 -w /tmp/LLB\_capture\_blade\_X.pcap

Note, change the interface name to LLB interface; change the VIP address; change the UDP;

The number of packets captured (in this case 1000000) will limit the time the capture runs. If all packets are 1500 bytes, this will give a maximum capture of 1.5GB.

To interrupt the capture if you want to stop it early, use ctrl+c.

## copy tcpdump

scp homer@[SGC]:/tmp/bond0\* /tmp

scp homer@[SGC]:/tmp/veth\* /tmp

scp homer@[SGC]:/tmp/br0\* /tmp

scp homer@[SGC]:/tmp/LLB\* /tmp

Login to each SGC and remove the files.

rm -f /tmp/\*.pcap

# Check the return value from OTP.

OTP also shows the APIs returns true.

In case of UDP:

gen\_udp:send/2; gen\_udp:send/4

inet\_udp:send/2; inet\_udp:send/4

inet6\_udp:send/2; inet6\_udp:send/4

For TCP case:

gen\_tcp:send/2; inet\_tcp:send/2; inet6\_tcp:send/2;

# Monitor return value from Linux System Call

From one linux shell on Active blades, execute " ps -ef|grep beam.smp " to find the pid of beam.smp progress, such as:

blade\_0\_15:homer# ps -ef |grep beam.smp

homer 3642 3603 0 Jun05 pts/1 00:05:23 /blade/active/GBS-SGC\_CXP9014753\_R99A634/OTP3\_CXC138863/otp3-R99A06/priv/pkg/erts-7.2.1/bin/beam.smp

then, trace “sendto” for udp and “send” for tcp with “strace”:

blade\_0\_15:homer# strace -p 3642 -f -e trace=sendto

Note, this is suitable to few traffics.

## Commands to be run before and after the issue

### On each active SGC blade that is handling evip traffic in the test.

Login to Linux on the active SGC blade and collect the output of these commands:

ip -s link

ss -s

ifconfig -a

Ipv4:

netstat -s

cat /proc/net/snmp

iptables -nvL

ipv6:

netstat --inet6 -s

cat /proc/net/snmp6

ip6tables -nvL

### Login to Linux on all SGC blades and collect the output of these commands:

vrctl -c xx ss -s

ipv4:

vrctl -c xx ip -s link

vrctl -c xx netstat -s

sudo vrctl -c xx iptables -nvL

sudo vrctl -c xx iptables -nvL -t raw

vrctl -c xx cat /proc/net/snmp

ipv6:

vrctl -c xx ip -6 -s link

vrctl -c xx netstat --inet6 -s

vrctl -c xx cat /proc/net/snmp6

vrctl -c xx ifconfig -a

vrctl -c xx ip6tables -nvL

vrctl -c xx ip6tables -nvL -t raw

### Commands on FEE elements on all blades running FEE

* IS:

### ssh root@$(ps aux |grep fee |grep lxc-execute | awk '{print $21}')

password:ncc1701D

### and collect the output of below commands:

netstat -s

netstat –inet6 -s

cat /proc/net/snmp

cat /proc/net/snmp6

iptables -nvL -t mangle

ip6tables -nvL -t mangle

ip -s link

ip -6 -s link

ss -s

* VSBG:

sudo ip netns exec `sudo ip netns list | grep FEE` netstat -s

sudo ip netns exec `sudo ip netns list | grep FEE` netstat –inet6 -s

sudo ip netns exec `sudo ip netns list | grep FEE` cat /proc/net/snmp

sudo ip netns exec `sudo ip netns list | grep FEE` cat /proc/net/snmp6

sudo ip netns exec `sudo ip netns list | grep FEE` iptables -nvL -t mangle

sudo ip netns exec `sudo ip netns list | grep FEE` ip6tables -nvL -t mangle

sudo ip netns exec `sudo ip netns list | grep FEE` ip -s link

sudo ip netns exec `sudo ip netns list | grep FEE` ss -s

### Commands on LBE elements on all blades running LBE

* IS:

### ssh root@$(ps aux |grep lbe |grep lxc-execute | awk '{print $21}')

password:ncc1701D

### and collect the output of below commands:

netstat -s

netstat –inet6 -s

cat /proc/net/snmp

cat /proc/net/snmp6

iptables -nvL -t mangle

ip6tables -nvL -t mangle

ip -s link

ss -s

* VSBG:

sudo ip netns exec `sudo ip netns list | grep LBE` netstat -s

sudo ip netns exec `sudo ip netns list | grep LBE` netstat –inet6 -s

sudo ip netns exec `sudo ip netns list | grep LBE ` cat /proc/net/snmp

sudo ip netns exec `sudo ip netns list | grep LBE ` cat /proc/net/snmp6

sudo ip netns exec `sudo ip netns list | grep LBE ` iptables -nvL -t mangle

sudo ip netns exec `sudo ip netns list | grep LBE ` ip6tables -nvL -t mangle

sudo ip netns exec `sudo ip netns list | grep LBE ` ip -s link

sudo ip netns exec `sudo ip netns list | grep LBE ` ss -s

### Commands on SE elements on all blades running SE

* IS:

### ssh root@$(ps aux |grep se |grep lxc-execute | awk '{print $21}')

password:ncc1701D

### and collect the output of below commands:

netstat -s

netstat –inet6 -s

cat /proc/net/snmp

cat /proc/net/snmp6

iptables -nvL -t mangle

ip6tables -nvL -t mangle

ip -s link

ss -s

* VSBG:

sudo ip netns exec `sudo ip netns list | grep SE` netstat -s

sudo ip netns exec `sudo ip netns list | grep SE` netstat –inet6 -s

sudo ip netns exec `sudo ip netns list | grep SE ` cat /proc/net/snmp

sudo ip netns exec `sudo ip netns list | grep SE ` cat /proc/net/snmp6

sudo ip netns exec `sudo ip netns list | grep SE ` iptables -nvL -t mangle

sudo ip netns exec `sudo ip netns list | grep SE ` ip6tables -nvL -t mangle

sudo ip netns exec `sudo ip netns list | grep SE ` ip -s link

sudo ip netns exec `sudo ip netns list | grep SE ` ss -s

## Collecting external trace

Collect the external trace for Core side SIP traffic for the time of the test.

## Collecting ASI