Configure vManage/vSmart/vEdge TCPDUMP Packet Capture in CLI Mode

Contents

Introduction

This document describes how to configure vManage/vSmart/vEdge **TCPDUMP** Packet Capture in CLI Mode.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Software-defined Wide Area Network (SD-WAN)

Components Used

The information in this document is based on Cisco vManage version 20.9.4

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with cleared (default) configuration. If your network is live,ensure that you understand potential impact of any command.

Background Information

In the Cisco SD-WAN architecture, vManage, vSmart and vEdge respectively play the core roles of management, control and data forwarding. To ensure the stability and security of the network and to troubleshoot network faults, network engineers often need to conduct packet capture and analysis on the traffic flowing through these devices. TCPDUMP is a lightweight and powerful command-line tool that can be used to capture and analyze data packets passing through interfaces.

By configuring and using TCPDUMP in CLI mode, users can directly capture real-time traffic on the device without the need for additional tools or intermediate proxy devices. This is of great significance for locating issues such as routing anomalies, control connection failures, packet loss, and verification of traffic paths. Since Cisco SD-WAN devices (such as vEdge) run customized operating systems (such as Viptela OS), the usage of TCPDUMP can differ slightly from that in traditional Linux environments in some aspects. Therefore, understanding its basic command structure and usage limitations is particularly crucial.

This section explain how to configure and run TCPDUMP in the CLI mode of vManage, vSmart and vEdge devices, in order to assist users in conducting effective network traffic analysis and problem diagnosis.

TCPDUMP(Controllers) Key Points Explanation

```
tcpdump [vpn x | interface x | vpn x interface x] options " "
Usage: tcpdump [-AbdDefhHIJKlLnNOpqStuUv] [ -B size ] [ -c count ]
               [ -E algo:secret ] [ -j tstamptype ] [ -M secret ]
               [ -T type ] [ -y datalinktype ] [ expression ]
```

- Specify an interface (cannot get output specifying vpn only)
- Put options in between quotation marks (**“ ”**) , use **ctrl c** to stop
- Use **–n** to prevent converting ip to hostname and **–nn** to prevent name and port ?
- **-v** shows more detail (IP header information, tos, ttl, offset, flags, protocol)
- **-vv** and **–vvv** show more detail in certain packet types
- Proto ex – udp, tcp icmp pim igmp vrrp esp arp
- Negate **!** or **not**, **&&** or **and**, **| |** or **or**, use with **( )** not (**udp** or **icmp**)

TCPDUMP (cont)

- Adapted from linux tcpdump command but does not support all available options. Snapshots of packets saved to a buffer, cannot export to a PCAP.
- Executes with –p flag, meaning ‘no-promiscuous mode’ – controller only capture packets destined for the controller interface, including control packets, or broadcast pkts. Cannot capture data plane traffic.
- Executed with –s 128, snapshot length in Bytes. First x bytes of packet is captured.

Use the TCPDUMP Command

This section provides examples that illustrate the way in which the**tcpdump**command is used.

```
vmanage# tcpdump ?
Possible completions:
interface  Interface on which tcpdump listens
vpn        VPN ID
```

The output of the **show interface description** command provides precise information about the **vpn/interface** name and number that is currently in use.

```
vmanage# tcpdump vpn 0 interface eth0 ?
Possible completions:
help       tcpdump help
options    tcpdump options or expression
|          Output modifiers
<cr>
```

You can add more conditions for packet capture filtering through the "**options**" keyword.

```
vmanage# tcpdump vpn 0 interface eth0 help

Tcpdump options:
help              Show usage
vpn               VPN or namespace
interface         Interface name
options           Tcpdump options like -v, -vvv, t,-A etc or expressions like port 25 and not host 10.0

e.g., tcpdump vpn 1 interface ge0/4 options "icmp or udp"

Usage: tcpdump [-AbdDefhHIJKlLnNOpqStuUv] [ -B size ] [ -c count ] [ -E algo:secret ] [ -j tstamptype ]
               [ -T type ] [ -y datalinktype ] [ expression ]
```

You can indicate the specific package count by options "**-c count**" command. If you do not indicate a specific package count, a continuous capture is run with no limit.

```
vmanage# tcpdump vpn 0 interface eth0 options "-c 10 "
tcpdump -p -i eth0 -s 128 -c 10 in VPN 0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 128 bytes
04:56:55.797308 IP 50.128.76.22.12746 > softbank219168102002.bbtec.net.12366: UDP, length 237
04:56:55.797371 IP 50.128.76.22.12746 > softbank219168102002.bbtec.net.12366: UDP, length 205
04:56:55.797554 IP 50.128.76.22.12746 > softbank219168102002.bbtec.net.12366: UDP, length 173
04:56:55.797580 IP 50.128.76.22.12746 > softbank219168102002.bbtec.net.12366: UDP, length 173
04:56:55.808036 IP 50.128.76.22.12746 > softbank219168102002.bbtec.net.12366: UDP, length 173
04:56:55.917567 ARP, Request who-has 50.128.76.31 (Broadcast) tell 50.128.76.1, length 46
04:56:55.979071 IP 50.128.76.22.12346 > 50.128.76.25.12346: UDP, length 182
04:56:55.979621 IP 50.128.76.25.12346 > 50.128.76.22.12346: UDP, length 146
04:56:56.014054 IP 50.128.76.22.12746 > softbank219168102002.bbtec.net.12366: UDP, length 237
04:56:56.135636 IP 50.128.76.32.12426 > 50.128.76.22.12546: UDP, length 140
10 packets captured
1296 packets received by filter
0 packets dropped by kernel
```

You can also add filter conditions about the **host address** and **protocol type** in the options.

```
vmanage# tcpdump vpn 0 interface eth0 options "-n host 50.128.76.27 and icmp"
tcpdump -p -i eth0 -s 128 -n host 50.128.76.27 and icmp in VPN 0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 128 bytes
05:21:31.855189 IP 50.128.76.27 > 50.128.76.22: ICMP echo reply, id 34351, seq 29515, length 28
```

```
05:21:34.832871 IP 50.128.76.22 > 50.128.76.27: ICMP echo request, id 44520, seq 29516, length 28
05:21:34.859655 IP 50.128.76.27 > 50.128.76.22: ICMP echo reply, id 44520, seq 29516, length 28
05:21:37.837244 IP 50.128.76.22 > 50.128.76.27: ICMP echo request, id 39089, seq 29517, length 28
05:21:37.866201 IP 50.128.76.27 > 50.128.76.22: ICMP echo reply, id 39089, seq 29517, length 28
05:21:40.842214 IP 50.128.76.22 > 50.128.76.27: ICMP echo request, id 24601, seq 29518, length 28
05:21:40.870203 IP 50.128.76.27 > 50.128.76.22: ICMP echo reply, id 24601, seq 29518, length 28
05:21:43.847548 IP 50.128.76.22 > 50.128.76.27: ICMP echo request, id 42968, seq 29519, length 28
05:21:43.873016 IP 50.128.76.27 > 50.128.76.22: ICMP echo reply, id 42968, seq 29519, length 28
05:21:46.852305 IP 50.128.76.22 > 50.128.76.27: ICMP echo request, id 23619, seq 29520, length 28
05:21:46.880557 IP 50.128.76.27 > 50.128.76.22: ICMP echo reply, id 23619, seq 29520, length 28
^C                                                                                   <<<< Ctrl + c can inter
11 packets captured
11 packets received by filter
0 packets dropped by kernel
```

Note: On Cisco IOS XE SD-WAN software, you can use Embedded Packet Capture (EPC) instead of TCPDUMP.

TCPDUMP Examples

Listening general UDP packet：

```
tcpdump vpn 0 options "-vvv -nnn udp"
```

Note: This can be applied for other protocols too... eg.: icmp, arp, etc

---

Listening for a specific port with ICMP and UDP:
```
tcpdump vpn 0 interface ge0/4 options "icmp or udp"
```

Listening on a specific port number(Listening on TLS Port):
```
tcpdump vpn 0 interface ge0/4 options “-vvv –nn port 23456”
```

Listening on a specific port number(Listening on DTLS Port):
```
tcpdump vpn 0 interface ge0/4 options “-vvv –nn port 12346”
```

Listening for a specific host (to/from that host): -e prints link-level header
```
tcpdump vpn 0 interface ge0/4 options “host 64.100.103.2 –vvv –nn –e”
```

Listening for a specific host with ICMP only
tcpdump vpn 0 interface ge0/4 options “host 64.100.103.2 && icmp”

Filtering by Source and/or Destination
tcpdump vpn 0 interface ge0/4 options “src 64.100.103.2 && dst 64.100.100.75”

Filter on GRE-encapsulated traffic
tcpdump vpn 0 interface ge0/4 options “-v –n proto 47 "

Related Documents

- Troubleshoot SD-WAN Control Connections
- Cisco SD-WAN: The Usual Suspects
- TCPDUMP MAN PAGE