ALG Troubleshooting Guide and Case Study

ALG Introduction

Some applications use multi-channels for data transmission, such as the widely used FTP. In such a condition the control channel and data channel are separated. Hillstone devices under strict security policy control set strict limits on each data channel, for example, only allow FTP data from internal network to external network to transfer on the well-known port TCP 21. Once in the FTP active mode, if an FTP server in the public network tries to initiate a connection to a random port of the host in the internal network, Hillstone devices will reject the connection and the FTP server will not work properly in such a condition. This requires Hillstone devices to be intelligent enough to properly handle the randomness of legitimate applications under strict security policies. In FTP instances, by analyzing the transmission information of the FTP control channel, Hillstone devices will be aware that the server and the client reached an agreement, and open up a temporary communication channel when the server takes the initiative to connect to a port of the client, thus assuring the proper operation of FTP.

StoneOS adopts the strictest NAT mode. Some VoIP applications may work improperly after NAT due to the change of IP address and port number. The ALG mechanism can ensure the normal communication of VoIP applications after the NAT. Therefore, the ALG supports the following functions:

- Under strict security policy rules, ensures the normal communication of multi-channel applications, such as FTP, TFTP, PPTP, RTSP, RSH, MSRPC, SUNRPC and SQLNET.
- Ensures the proper operation of VoIP applications such as SIP and H.323 in NAT mode, and performs monitoring and filtering according to the policies.

- ALG is used to automatically adapt complex protocols.
  - Supports multiple ports/directions of a protocol
  - Enables/Disables pinholes of each session

StoneOS allows you to enable or disable ALG for different applications. Hillstone devices support ALG for the following applications: FTP, HTTP, MSRPC, PPTP, Q.931, RAS, RSH, RTSP, SIP, SQLNetV2, SUNRPC, TFTP, DNS, and H323. You can not only enable or disable ALG for applications, but also specify H323's session timeout.

ALG supports strict mode and non-strict mode. In the strict mode, the newly-created pinhole has the SNAT port which is the same as the SNAT port of the control session. By default, the strict mode is enabled.

Content Analysis、Filter

HTTP: URL Filtering、 attachment download limiting、IPS/AV analysis

Create Pinhole

FTP、TFTP etc.

Modify message content

Modify the IP、port pf APP after NAT

The purpose of ALG Pinhole:

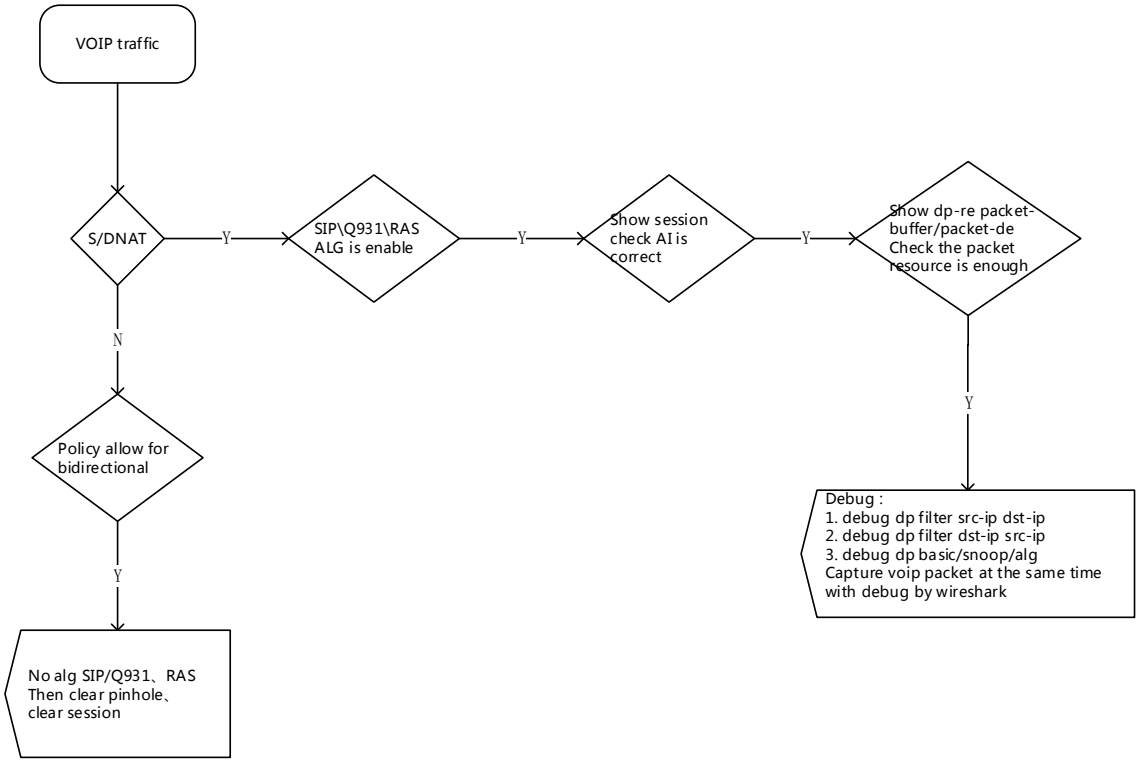
ALG pinhole is a complementary mechanism for FW policy, which is used to allow the protocol traffic of self-negotiation communication port go through FW.

Control connection matches configured known service policy, before data transition the control connection negotiates a pair of communication ports for data connection, but the negotiated ports may failed to be matched by policy, for this case we need create a pinhole for data connection.

When the first message of Data connection arriving at FW, session will be created based on pinhole without checking the policy.

ALG Troubleshooting Process

Take the VOIP as an example:



1. For audio and video communication, please check if NAT was used between both sides, if not, please disable the corresponding ALG function and clear session/pin, then try to reconnect;
2. Check if ALG function is enabled、 **ALG AUTO** is enabled for application with nonstandard port;
3. Application identification is enabled, check if there is user-defined app-signature; ***show session*** to check if APP is identified correctly;
4. ***show dp-re packet-buffer*** to check if buffer number is normal, ***show dp-r packet-descriptor*** to check if Free number is normal;
5. If the issue related to video/audio slower buffering, please check the device configuration to see if it was limited by some functions, such as qos，session-limit，attack defense etc.；
6. Some SIP vender was using non-standard protocol message, please enable the ***icmp-unreachable-session-keep*** under flow
7. Configure the bidirectional (source and destination) ***debug filter*** , ***debug dp basic/snoop/alg*** to check if there is packet drop

8. Collect related information and send to Hillstone TAC for further analysis: configuration file、 network topology、 ***debug dp basic/snoop/alg*** information of service initial connection and the packet capture of port mirroring during that time、 ***show version*** information.

Case Study (Common issues):

Issue 1：

The client software could not be updated after firewall deployed, the update was successful if remove the firewall. Tried to update software in PC by firewall but failed either.

Topology:
Internet---FW---SW---(PC)

Analysis and Solution：

1、***debug…*** - check the data and make sure no data drop

2、port 6001 is used, identify the X-WINDOW service

nqzy[DBG](config)# sh session dst-ip 10.186.169.253
session: id 100, proto 6, flag a, flag1 20000, created 1178871, life 1799, policy 9,app 122(X-WINDOW) flag 0x0, auth_user_id 0, reverse_auth_user_id 0
flow0(8/40300b10): 192.168.1.13:52434->10.186.169.253:6001
flow1(9/300b10): 10.186.169.253:6001->10.186.215.10:52434

3、Capture packet from client PC, track the tcp flow:

220 Microsoft FTP Service
USER qrsreckoner
331 Password required for qrsreckoner.
PASS 123.com
230 User qrsreckoner logged in.
CWD interface
250 CWD command successful.
PWD
257 "/interface" is current directory.
TYPE A

200 Type set to A.
PORT 192,168,1,13,203,21
500 Invalid PORT Command.
LPRT 6,16,0,0,0,0,0,0,0,0,220,190,24,0,19,224,128,119,2,203,21
500 'LPRT 6,16,0,0,0,0,0,0,0,0,220,190,24,0,19,224,128,119,2,203,21': command not understood
421 Timeout (120 seconds): closing control connection.
421 Terminating connection.

We can find the port 6001 (ftp server) error

4、Now we can assure it is alg and application identification issue.

5、SNAT is configured for Internet, so we enable the ftp alg, and define 6001 port as ftp

```
nqzy[DBG](config)#  sh  app-signature  static
Total  signature  count:  1
S:  Signature  Status  (E  -  Enabled;    D  -  Disabled)
Flag  :  C  -  Continue  Application  identify
================================================================================
S      ID  Src-zone      Src-addr                  Dst-addr                  Pr
otocol      Dst-Port/Type    Src-Port/Code    Application        Flag
--------------------------------------------------------------------------------
E        1  Any                Any                              Any
              TCP                6001                       Any
        FTP
================================================================================
```

6、The app is identified as ftp and update is successful now

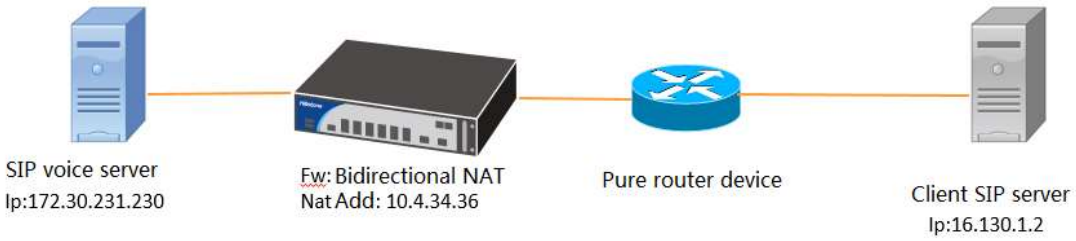
```
nqzy[DBG](config)#  sh  session  dst-ip    10.186.169.253
session:  id  100,  proto  6,  flag  a,  flag1  20000,  created  1178871,  life
  1799,  policy  9,app  4(FTP)  flag  0x0,  auth_user_id  0,  reverse_aut
h_user_id  0
    flow0(8/40300b10):  192.168.1.13:52434->10.186.169.253:6001
    flow1(9/300b10):  10.186.169.253:6001->10.186.215.10:52434
session:  id  1999,  proto  6,  flag  a,  flag1  20000,  created  1178750,  lif
e  1706,  policy  9,app  4(FTP)  flag  0x0,  auth_user_id  0,  reverse_au
th_user_id  0
    flow0(8/40300b10):  192.168.1.13:52391->10.186.169.253:6001
    flow1(9/300b10):  10.186.169.253:6001->10.186.215.10:52391
```

Configuration file (optional): the final configuration in FW.
Other information (optional): captured packets etc.

Issue 2：

FW was configured with bidirectional NAT to isolate internal SIP server and external SIP server. Ping or Telnet to port was normal, but SIP protocol had issue and caused communication disconnected.

Network topology:



Troubleshooting steps：

1. Firstly, test the SIP voice communication in routing mode without FW, and no problem for SIP connection. Use Wireshark to capture the packet.

Client SIP server IP: 16.130.1.2

Sip voice server IP: 16.12.18.151

Normal message without FW.pcap

2. Connect to FW, test with bidirectional NAT, SIP voice connection is abnormal under NAT. Capture packet by FW and Wireshark

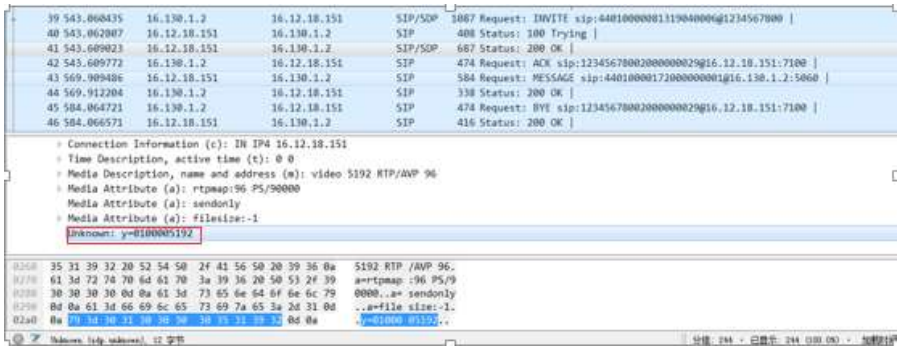
Client SIP server IP: 16.130.1.2

SIP voice server IP: 172.30.231.230 WAN bidirectional NAT IP: 10.4.34.36

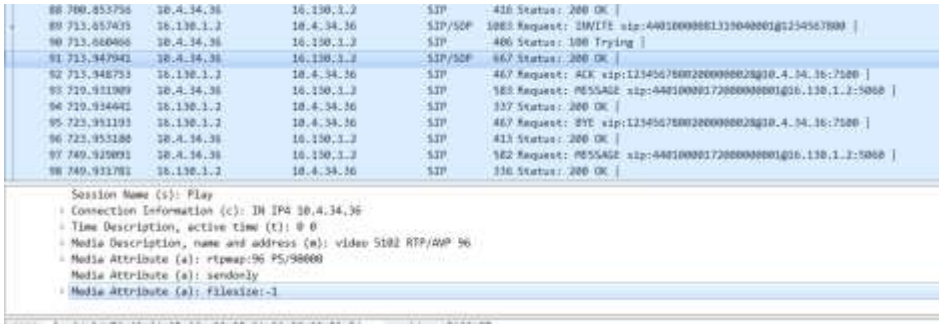
pacap with FW.pcap SIP voice debug in FW.txt

3. Compare the two Wireshark packet capture to find the difference, and the analyze.

Message without FW（part）:



Message with FW:



Based on those information, we can find the returned packet via FW is missing Y value

4. Check the message information of FW

2017-04-05 15:15:18, DEBUG@FLOW: core 16 (sys up 0x2f3738f4b ms): 12674371: (i) len=690 0023.89a1.813f->001c.5463.c9f4/800

172.30.231.230->16.130.1.2/17

vhl=45, tos=00, id=12842, frag=0000, ttl=127, tlen=676

udp:ports 7100->5060, len=656

rx_handle_prepare: 0023.89a1.813f->001c.5463.c9f4, size 690, type 0x800, vid 0, port ethernet0/9

dp_prepare_if_for_pak

Switchid is 35(interface redundant1) port redundant1 ,pak iif=ethernet0/9

Not from apm packet, return.

Not ha apm heart beat message.

rx_handle_prepare i_if is redundant1

Start l3 forward

Packet: 172.30.231.230 -> 16.130.1.2, id: 12842, ip size 676, prot: 17(UDP): 7100 -> 5060

ad_vector_for_fast_flow: zonename inside, proto_flag[1] 0, proto 17

dp_prepare_pak_lookup srcip: 172.30.231.230, dstip: 16.130.1.2, src-port:7100, dst-port:5060, prot 17

Found the session 2414

session: id 2414, prot 17, flag0 2,flag1 20000, created 12673358, life 60

flow0(if id: 35 flow id: 4828 flag: 40200910):172.30.231.230:7100
->16.130.1.2:5060
flow1(if id: 12 flow id: 4829 flag: 200910): 16.130.1.2:5060
->10.4.34.36:7100
dp_app_proc: 0x0x8000000048fa4f00
NEED get session lock.
try lock SUCCESS.
Enter UPD process
Received a SIP over (UDP) Packet with (648) bytes.
SIP/2.0 200 OK
Via: SIP/2.0/UDP 16.130.1.2:5060;rport=5060;branch=z9hG4bKPj6d71eab5-0fd2-4cc6-8394-b9e9d2d84c93
From: <sip:44010000172000000001@4401000017>;tag=8bce6556-a946-4cf0-95b4-06265e6a69b4
To: <sip:44010000081319040001@1234567800>;tag=1942750434
Call-ID: 9160eb59-db67-4891-9554-8528135eefe9
CSeq: 4762 INVITE
Contact: <sip:12345678002000000028@172.30.231.230:7100>
Content-Type: Application/SDP
User-Agent: NCG V2.6.0.299938

Content-Length: 180
v=0
o=44010000081319040001 0 0 IN IP4 16.12.49.121
s=Play
c=IN IP4 172.30.231.230
t=0 0
m=video 5112 RTP/AVP 96
a=rtpmap:96 PS/90000
a=sendonly
a=filesize:-1
y=0106005112

Terminator found, effectlen=648
The message(INVITE) hash = 11.
Message(INVITE, 200, 2XX Response) drive the IVT-TA(ID=11) state from (INVITE Proceeding) to (Terminated).
from_tag: 8bce6556-a946-4cf0-95b4-06265e6a69b4, dlg->tag[0]: 8bce6556-a946-4cf0-95b4-06265e6a69b4, dlg->tag[1]: (null)
The dialog(8bce6556-a946-4cf0-95b4-06265e6a69b4, (null), (9160eb59-db67-4891-9554-8528135eefe9@(null))) is matched.
SIP process invite 2xx response, dlg->pri_state 2
get first invite 2xx response message.
There is no or more Record-Route headers.

Process the media(video) of SDP.
Media port is 5112.
pinhole_dup_dst_check_for_multi: pinhole hash(2887671692) search key: dst_ip: 172.30.231.230, dst_port: 5112, prot: 17
Pinhole1/2 not found by dst chash
pinhole_dup_src_check_for_multi: pinhole hash(2887671692) search key: src_ip: 172.30.231.230, src_port: 5112, prot: 17
Pinhole1/2 not found by src chash
Create 2 pinholes from flow0
A new pinhole is allocated, sequence 296984, pinhole_seq 296984, g_pinhole_count 24
A new pinhole is allocated, sequence 296985, pinhole_seq 296985, g_pinhole_count 25
SRC NAT in session .. newip=10.4.34.36
Changing connection address from (172.30.231.230) to (10.4.34.36)
Processed the response's SDP successfully.
pinhole dst hash:168073807
pinhole src hash:2887671693
Insert the pinhole into hash tables.
Protocol(17) AppID:875 PH_Flags:0x2300104.
reference 4
Ingress-Side(0.0.0.0:0 --> 10.4.34.36:5113 flow_token:21)
Egress-Side(172.30.231.230:5113 --> 0.0.0.0:0)
sess_flag0:0x4, reverse_sess_flag0:0x2
tunnel_id:0, seq: 296985, ref_cnt=4, life before hit:120,life after hit:600,appid:875,policy id:7,flow0ifid:0,flow1ifid:35,refflow:4
828
Init pinhole timer to 120
pinhole dst hash:168073806
pinhole src hash:2887671692
Insert the pinhole into hash tables.
Protocol(17) AppID:875 PH_Flags:0x2300104.
reference 4
Ingress-Side(0.0.0.0:0 --> 10.4.34.36:5112 flow_token:21)
Egress-Side(172.30.231.230:5112 --> 0.0.0.0:0)
sess_flag0:0x4, reverse_sess_flag0:0x2
tunnel_id:0, seq: 296984, ref_cnt=4, life before hit:120,life after hit:600,appid:875,policy id:7,flow0ifid:0,flow1ifid:35,refflow:4
828
Init pinhole timer to 120
The AoR(44010000081319040001) is not registed.
Changing uri's host from (172.30.231.230) to (10.4.34.36)
TA is not processed since created or terminated.
Destroy the transaction(ID=11).
Forward the SIP message.
Xlate SIP Message:

SIP/2.0 200 OK
Via: SIP/2.0/UDP 16.130.1.2:5060;rport=5060;branch=z9hG4bKPj6d71eab5-0fd2-4cc6-8394-b9e9d2d84c93
From: <sip:44010000172000000001@4401000017>;tag=8bce6556-a946-4cf0-95b4-06265e6a69b4
To: <sip:44010000081319040001@1234567800>;tag=1942750434
Call-ID: 9160eb59-db67-4891-9554-8528135eefe9
CSeq: 4762 INVITE
Contact: <sip:12345678002000000028@10.4.34.36:7100>
Content-Type: Application/SDP
User-Agent: NCG V2.6.0.299938

Content-Length: 163
v=0
o=44010000081319040001 0 0 IN IP4 16.12.49.121
s=Play
c=IN IP4 10.4.34.36
t=0 0
m=video 5112 RTP/AVP 96
a=rtpmap:96 PS/90000
a=sendonly
a=filesize:-1

End the SIP packet process

Change requested
Last offset 0 req->offset 0
Last offset in changebuffer 0
pak marked to dirty
shrinking app for 21 bytes
shrinked, now 0 needs
content dirty, modify checksum
unlock
The cached packet number is -1555235647
The processed cached packet number is -1555235647
The discarded packet number is 227
flow mac_copy
L3 forward, out if is ethernet0/2, mtu 1500
12674371: (o) len=669 001c.5463.c9c4->5439.df1a.af5f/800
    10.4.34.36->16.130.1.2/17
    vhl=45, tos=00, id=12842, frag=0000, ttl=126, tlen=655
    udp:ports 7100->5060, len=635

Based on the ***debug alg*** information, we can find before the ALG, the message from client has the

Y parameter, but after ALG, there is no Y parameter. FW changed the parameters and caused Y value was lost.

5. The purpose of Y value:

Y is a decimal integer string, which is called SSRC. Format is ddddddddd.
SSRC value is generated by the SIP monitoring domain in which the media transport device is located, it is used as an identifier for the media stream.

So if Y value missed, the media stream can't be identified, which cause the failure of media play.

Solution:
This value is the new checksum parameter, it will be supported at below firmware version and later release: 5.5R1P14, 5.5R2P7, 5.5R3P5, 5.5R4P3

Issue 3：

Policy mode IPsec VPN, video、audio communication is abnormal (ALG related apps)

Key Log：

Some pinhole message was sent from FW without encryption. Such as below syn ack message:

2015-06-15 19:22:04, DEBUG@FLOW: core 1 (sys up 0x3046cca75 ms): Finish decap

Packet: 10.110.107.199 -> 192.168.109.8, id: 50465, ip size 44, prot: 6(TCP): 1040 -> 15032

dp_prepare_pak_lookup srcip: 10.110.107.199, dstip: 192.168.109.8,prot 6

No session found, try to create session

————–First path creating new session————–

Found a pinhole

Connection route.

Session created from pinhole

The following session is installed

session: id 972260, prot 6, flag0 8000000,flag1 0, created 12959140, life 20

flow0(if id: 27 flow id: 1944520 flag: 40300a00):10.110.107.199:1040

->192.168.109.8:15032

flow1(if id: 34 flow id: 1944521 flag: 100a00): 192.168.109.8:15032

->10.110.107.199:1040

Session installed successfully

————————–First path over————————

Found the session 972260

session: id 972260, prot 6, flag0 8100040,flag1 0, created 12959140, life 20

flow0(if id: 27 flow id: 1944520 flag: 40300a10):10.110.107.199:1040

->192.168.109.8:15032

flow1(if id: 34 flow id: 1944521 flag: 300a10): 192.168.109.8:15032

->10.110.107.199:1040

DP filter remathed after app identification? Yes

TCP seqence handling

L3 forward, out if is redundant2.109

Send sub or vsi interface packet on interface 22, tag is 109

begin to tx pak in redundant_drv: ethernet0/2

tcp:ports 1040->15032, seq=5771430, ack=0, flag=24578/SYN

_msw_ext_dsa_tag_encap_from_cpu_with_port_: TX packet from dev 31 to dev 0 port 14 vid 109 ivid 0 cos 0.

2015-06-15 19:22:04, DEBUG@FLOW: core 1 (sys up 0x3046cca75 ms): tcp:ports 15032->1040, seq=732832404, ack=5771431, flag=245

94/SYN/ACK

000c.29cd.e2ad->001c.54ff.0811, size 60, type 0x800, vid 109, port ethernet0/2

vlan get sub if redundant2.109

Switchid is 34(interface redundant2.109) port redundant2.109

Switchid is 34(interface redundant2.109) port redundant2.109 ,pak iif=ethernet0/2

dp_prepare_if_for_pak i_if is redundant2.109

Start l3 forward

Packet: 192.168.109.8 -> 10.110.107.199, id: 0, ip size 44, prot: 6(TCP): 15032 -> 1040

dp_prepare_pak_lookup srcip: 192.168.109.8, dstip: 10.110.107.199,prot 6

Found the session 972260

session: id 972260, prot 6, flag0 8100040,flag1 0, created 12959140, life 20

flow0(if id: 27 flow id: 1944520 flag: 40300a10):10.110.107.199:1040

->192.168.109.8:15032

flow1(if id: 34 flow id: 1944521 flag: 300a10): 192.168.109.8:15032

->10.110.107.199:1040

DP filter remathed after app identification? Yes

TCP seqence handling

L3 forward, out if is redundant1.3

Send sub or vsi interface packet on interface 21, tag is 3

begin to tx pak in redundant_drv: ethernet0/0

tcp:ports 15032->1040, seq=732832404, ack=5771431, flag=24594/SYN/ACK

Analysis：

When creating pinhole, there was no record for policy id and tunnel id, so the created session policy was default based on pinhole, it would not be encrypted according to the tunnel id of original message.

Workaround & Solution：

1. Disable the corresponding alg（clear session、pinhole）, VPN policy set as any permit

2. **Or**, change VPN to routing mode, when creating session for pinhole, it will also check route, the encryption will be triggered by route.