Quick Note 64

How to Troubleshoot OpenVPN On TransPort WR Routers

Contents

1 INTRODUCTION

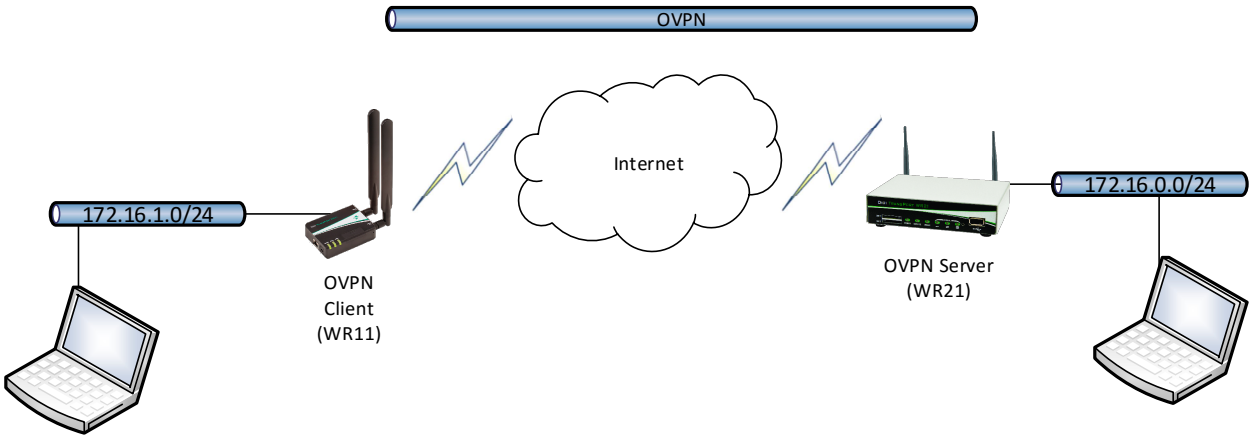
1.1 Outline

This document describes how to troubleshoot an OpenVPN connection on TransPort routers.

OpenVPN is a full-featured SSL VPN which implements OSI layer 2 or 3 secure network extension using the industry standard SSL/TLS protocol, supports flexible client authentication methods based on certificates, smart cards, and/or username/password credentials, and allows user or group-specific access control policies using firewall rules applied to the VPN virtual interface.

When configuring a TransPort router as a OpenVPN Server or Client, users can experience issues in the connection establishment or in the traffic routing through it. This document provide a guide on how collect debug and traces useful to troubleshoot those kind of problems.

In this example will be considered a scenario as the following, where 2 Transport routers are used as OpenVPN Client and Server, but the same troubleshooting guide applies to the case where only one of them is a Transport router and the other one is, for example, a Windows PC.



1.2 Assumptions

This guide has been written for use by technically competent personnel with a good understanding of the communications technologies used in the product and of the requirements for their specific application. It also assumes a basic ability to access and navigate a Digi Transport router and configure it with basic routing functions.

This application note applies to:

Model: Digi Transport WR21 /WR11

Other Compatible Models: All Digi WR Transport models

Firmware versions: 5.077 and later

Configuration: This Application Note assumes the devices are configured to establish an OpenVPN connection.

1.3 Corrections

Requests for corrections or amendments to this application note are welcome and should be addressed to: tech.support@digi.com.

Requests for new application notes can be sent to the same address.

1.4 Version

Version Number	Status
1.0	Published

2 TROUBLESHOOTING THE OPENVPN/SSL CONNECTION ESTABLISHMENT

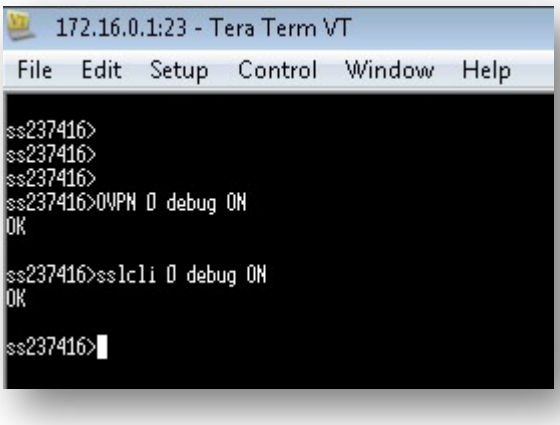
2.1 Enabling SSL and OpenVPN debug

By default, when there is an OpenVPN connection attempt, the result can be seen in the eventlog section (). Anyway, in case of issues, the eventlog will not give many details. To deep analyse what is going on, the SSL and OpenVPN debug can be enabled. The easiest way to do this is to access the router (client or server or both) via a serial/telnet/SSH connection and issue the following CLI commands:

OVPN x debug ON

sslcli x debug ON

Where x is the OpenVPN/SSL instance that must be monitored, so for example if it is 0, it will be as following:



Once the debugs are enabled, in order to see the output of them, there are the following options depending on how the router is accessed by the user:

- **Serial Port:**

Depending on which serial port is used, the debug output will be shown after issuing the following commands:

debug 0 (if ASY 0)

debug 1 (If ASY 1)

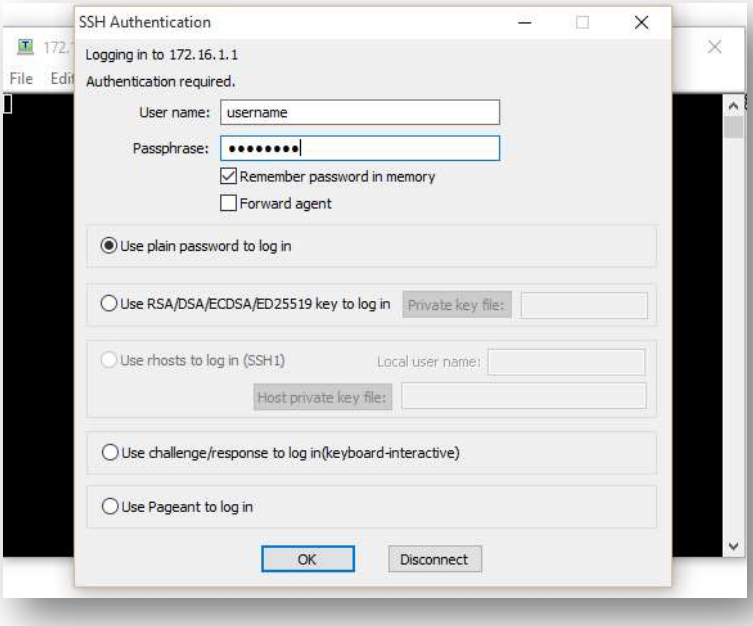
-Telnet:

When using a Telnet connection, the command to issue in order to see the debug output is the following:

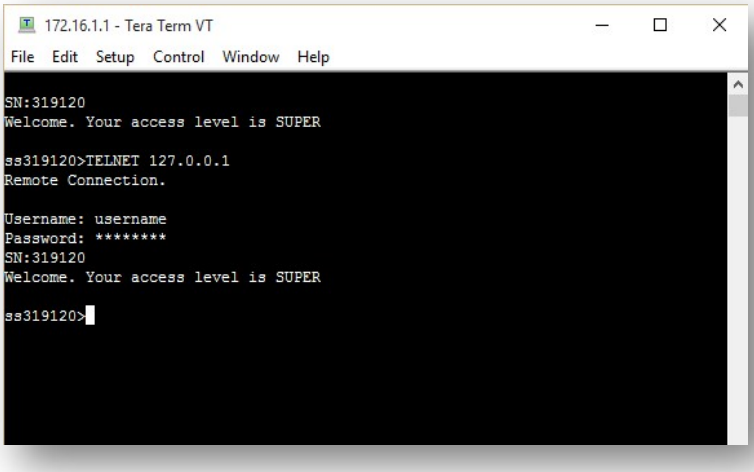
debug t

-SSH:

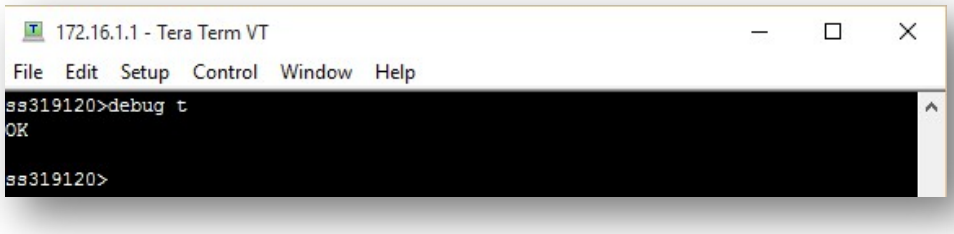
To obtain debug output through SSH connect to the router via an SSH client & log in:



Now, from the command line run the CLI command "*TELNET 127.0.0.1*" & log in:

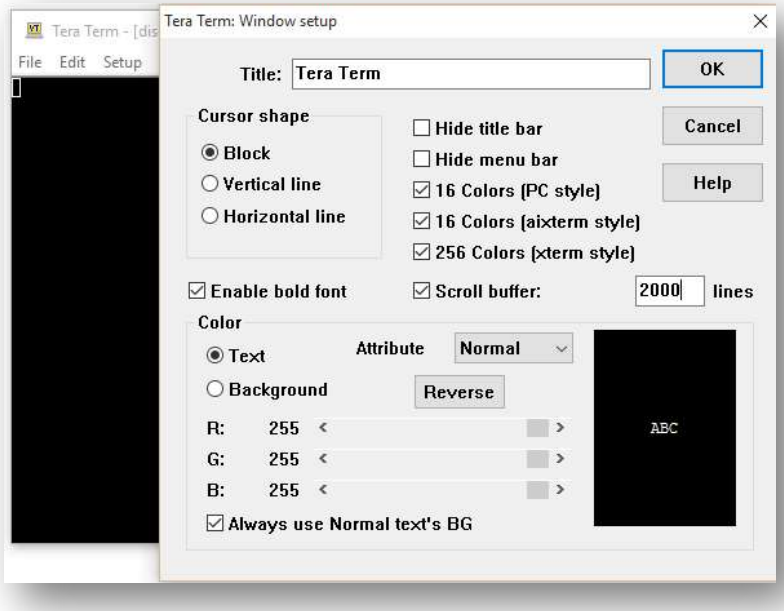


It is now possible to send the debug output to the Telnet port running the CLI command "debug t" to see the debug output**:**

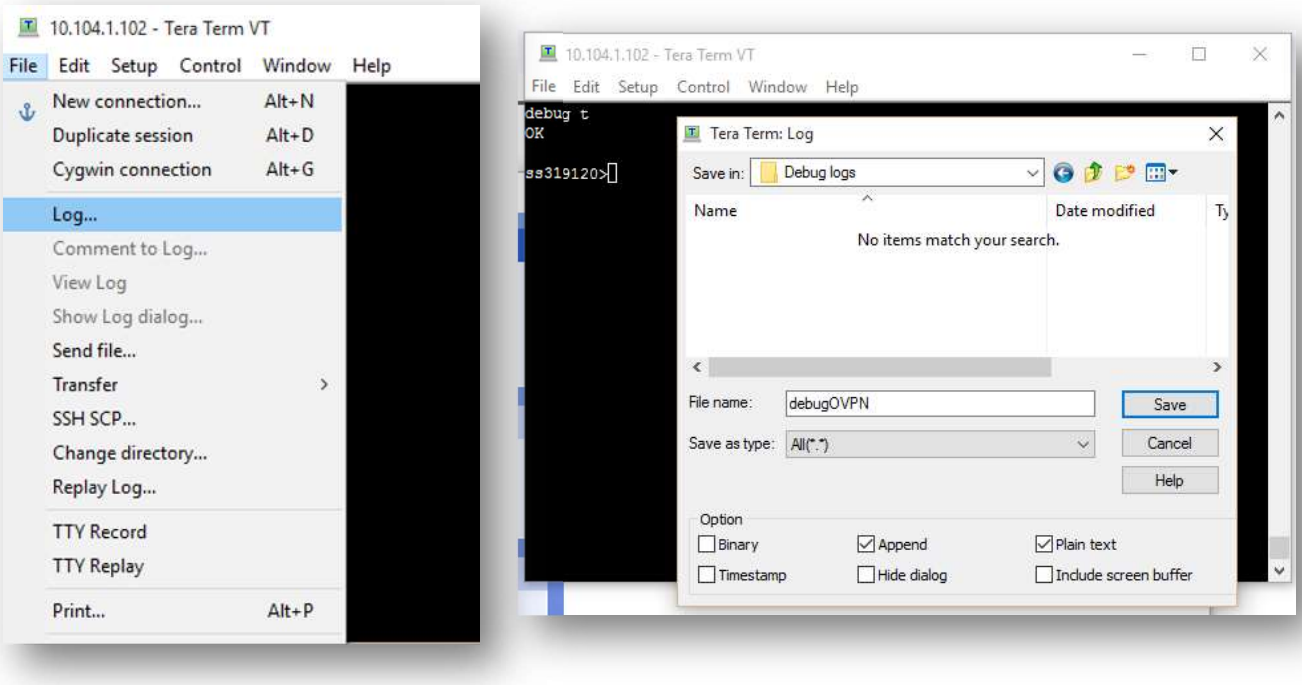


Please note:

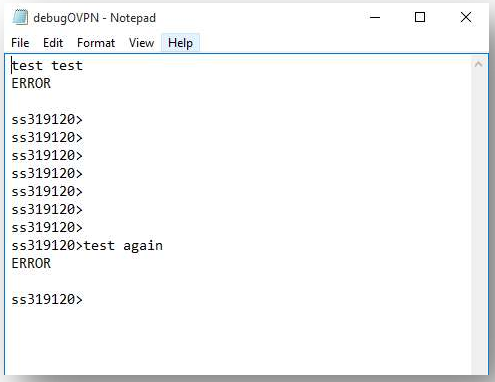
In order to be sure to have all the debug output without losing old logs, please assure that the scroll line limit is high (for example 2000). In Teraterm, for example, the “Scroll buffer” value can be checked and changed in Setup > Window:



Moreover, the logging function of the emulator can be used to collect the debug:



It will generate a log file filled what all the debug shown on the current session:



Once the debugs and their output are enabled, the OpenVPN/SSL connection can be monitored to catch eventual issues. In the following session some example will be shown.

2.2 Good OVPN debug

In case there is no error or issue in the communication between Client and Server, the debug will be like the following (the client one is shown, on the server will be similar):

```
ss319120>debug t
OK
ss319120>
OVPN 0 link socket connected
OVPN 0 init ctx
OVPN 0 start key negotiation
OVPN 0 start negotiation timer
OVPN 0 negotiation waiting on initial ACKS to complete

Client cipher list: SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-
SHA:DH-DSS-AES256-GCM-SHA384:DHE-DSS-AES256-GCM-SHA384:DH-RSA-AES256-GCM-SHA384:DHE-
RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA256:DH-RSA-AES256-
SHA256:DH-DSS-AES256-SHA256:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:DH-RSA-AES256-
SHA:DH-DSS-AES256-SHA:AES256-GCM-SHA384:AES256-SHA256:AES256-SHA:PSK-AES256-CBC-
SHA:SRP-DSS-AES-128-CBC-SHA:SRP-RSA-AES-128-CBC-SHA:SRP-AES-128-CBC-SHA:DH-DSS-
AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DH-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-
GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-DSS-AES128-SHA256:DH-RSA-AES128-SHA256:DH-DSS-
AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:DH-RSA-AES128-SHA:DH-DSS-AES128-
SHA:AES128-GCM-SHA256:AES128-SHA256:AES128-SHA:PSK-AES128-CBC-SHA

SSL_connect:before/connect initialization

SSL_connect:SSLv3 write client hello A

Send DA_RQ to SSL ID 2 len 136
OVPN 0 stop link retransmit timer
OVPN 0 reliability layer timeout
OVPN 0 stop link retransmit timer
OVPN 0 got incoming ciphertext len 1450
Send DA_IN to SSL ID 2 len 1450
SSL_connect:SSLv3 read server hello A

OVPN 0 got incoming ciphertext len 614
Send DA_IN to SSL ID 2 len 614
SSL_connect:SSLv3 read server certificate A

SSL_connect:SSLv3 read server done A

SSL_connect:SSLv3 write client key exchange A

SSL_connect:SSLv3 write change cipher spec A

SSL_connect:SSLv3 write finished A

SSL_connect:SSLv3 flush data

OVPN 0 stop link retransmit timer
OVPN 0 got incoming ciphertext len 226
```

```
Send DA_IN to SSL ID 2 len 226
SSL_connect:SSLv3 read server session ticket A

SSL_connect:SSLv3 read finished A


---
Certificate chain
 0
s:/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=server/name=changeme/emailAddress=support
@digi.com
   i:/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=OpenVPN-
CA/name=changeme/emailAddress=support@digi.com
 1 s:/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=OpenVPN-
CA/name=changeme/emailAddress=support@digi.com
   i:/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=OpenVPN-
CA/name=changeme/emailAddress=support@digi.com
---
Peer certificate
subject=/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=server/name=changeme/emailAddress=s
upport@digi.com
issuer=/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=OpenVPN-
CA/name=changeme/emailAddress=support@digi.com
---
No client certificate CA names sent
---
SSL handshake has read 2290 bytes and written 529 bytes
---
New, TLSv1/SSLv3, Cipher is AES256-GCM-SHA384
Server public key is 1024 bit
SSL-Session:
               Protocol  : TLSv1.2
                                   Cipher    : AES256-GCM-SHA384
                                                                   Session-ID:
1B902762903681F1DABB24DD5C022CF34CD1CFCA90BBF5C7A12E6940A3DC4209

Session-ID-ctx:

Master-Key:
451245BE8480F76E157F8C2310B0CCA4D501B8AFA5FE495B7BC2BC50C92445FC95E608B44AD6781F31EF9
B1A543B2EB2
            Key-Arg   : None
                               PSK identity: None
                                                    PSK identity hint: None
                                                                                SRP
username: None

TLS session ticket lifetime hint: 300 (seconds)

TLS session ticket:

0000 - 6f 2a 8c e3 53 81 2f 87-c4 7a 10 33 f7 fc 57 d9   o*..S./..z.3..W.

0010 - 34 bb eb fb 72 4e 2c 47-2a 87 18 fd c7 2e 07 56   4...rN,G*......V
```

```
                                                        0020 - 01 ed 3b c5 06 30
83 86-3f 3a e9 f4 e9 09 2e 19   ..;..0..?:......

0030 - 3a 61 e4 36 40 5e 04 d4-38 ce 8f b0 14 75 9a 37   :a.6@^..8....u.7

0040 - 34 a1 23 bb 18 66 4a 7d-f7 e2 51 47 1d 53 3e aa   4.#..fJ}..QG.S>.
                    0050 - f8 59 ae 2b 54 23 ac ee-e3 b4 af ab 73 31 25 a3
.Y.+T#......s1

0060 - 5d 1e e9 63 81 75 a8 e3-e8 a4 e9 81 9c dc 8b 16   ]..c.u..........

0070 - 5a ca 30 91 5d 90 f4 68-37 c0 43 8b d8 8f f9 f1   Z.0.]..h7.C.....

0080 - c7 e2 b1 01 fd 55 78 c9-36 34 25 8e 3c 13 04 68   .....Ux.64<..h
                                                       0090 - 7a a1 6f a2 13 6a b8 92-
52 65 0c e6 aa 40 a0 a0   z.o..j..Re...@..


Start Time: 1501594745

Timeout   : 300 (sec)

Verify return code: 0 (ok)

---

OVPN 0 stop link retransmit timer
OVPN 0 got incoming ciphertext len 117
Send DA_IN to SSL ID 2 len 117
Init cipher AES-256-CBC (ctx 4359e9c8)
Cipher 'AES-256-CBC' initialized with 256 bit key
HMAC authentication using 160 bit SHA1
Init cipher AES-256-CBC (ctx 4359eb3c)
Cipher 'AES-256-CBC' initialized with 256 bit key
HMAC authentication using 160 bit SHA1
OVPN 0 KS negotiation complete
PUSH_REQUEST is required
Send DA_RQ to SSL ID 2 len 13
OVPN 0 stop link retransmit timer
OVPN 0 got incoming ciphertext len 104
Send DA_IN to SSL ID 2 len 104
OVPN 0 got PUSH_REPLY
OVPN 0 process option ifconfig 192.168.0.2 192.168.0.1
OVPN 0 got ifconfig option. His IP: 192.168.0.1, our IP 192.168.0.2
OVPN 0 process option route 172.16.0.0 255.255.255.0
OVPN 0 got route option. Add route 172.16.0.0/255.255.255.0
OVPN 0 stop negotiation timer
OVPN 0 UP. IP address: 192.168.0.2
OVPN 0 add route 172.16.0.0/24
```

As shown in the debug output, the SSL and OpenVPN exchange ends successful with the OVPN interface going UP and the route pushed from the server being added to the routing table.

Related to this negotiation, what is shown on the event log section is the following:

```
13:39:06, 01 Aug 2017,OVPN 0 Available,Activation
13:39:06, 01 Aug 2017,OVPN 0 up
```

2.3 Bad OVPN debug

In this section, some examples of debug related to common issues that can be experienced using OpenVPN will be shown.

2.3.1 Server not responding

Below is shown an example of OpenVPN negotiation ending with failure due to timeout. Actually, the negotiation never starts as the client keeps waiting for the initial ACKS exchange with the server:

```
OVPN 0 init ctx
OVPN 0 start key negotiation
OVPN 0 start negotiation timer
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
```

```
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 reliability layer timeout
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation waiting on initial ACKS to complete
OVPN 0 negotiation timeout
OVPN 0 DOWN
OVPN 0 stop link retransmit timer
OVPN 0 closing link socket
```

```
OVPN 0 link socket disconnected
OVPN 0 start socket idle timer
```

From this output is clear that the Client is not receiving any reply from the Server and the OpenVPN negotiation ends due to timeout expired, so, in such cases, first thing to check is that the OpenVPN Server address configured in the Client is correct and reachable from the Client.

What is shown in the eventlog section is simply an OVPN interface down events:

```
14:15:21, 01 Aug 2017,OVPN 0 down
```

2.3.2 OVPN Configuration error

The example below shows a case where there is an error in the configuration of the OpenVPN interface. In particular, the Cipher and Digest were different between Client and Server:

```
ss319120>
OVPN 0 link socket connected
OVPN 0 init ctx
OVPN 0 start key negotiation
OVPN 0 start negotiation timer
OVPN 0 negotiation waiting on initial ACKS to complete
Client cipher list: SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-
SHA:DH-DSS-AES256-GCM-SHA384:DHE-DSS-AES256-GCM-SHA384:DH-RSA-AES256-GCM-SHA384:DHE-
RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA256:DH-RSA-AES256-
SHA256:DH-DSS-AES256-SHA256:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:DH-RSA-AES256-
SHA:DH-DSS-AES256-SHA:AES256-GCM-SHA384:AES256-SHA256:AES256-SHA:PSK-AES256-CBC-
SHA:SRP-DSS-AES-128-CBC-SHA:SRP-RSA-AES-128-CBC-SHA:SRP-AES-128-CBC-SHA:DH-DSS-
AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DH-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-
GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-DSS-AES128-SHA256:DH-RSA-AES128-SHA256:DH-DSS-
AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:DH-RSA-AES128-SHA:DH-DSS-AES128-
SHA:AES128-GCM-SHA256:AES128-SHA256:AES128-SHA:PSK-AES128-CBC-SHA


SSL_connect:before/connect initialization

SSL_connect:SSLv3 write client hello A

Send DA_RQ to SSL ID 35 len 136
OVPN 0 stop link retransmit timer
OVPN 0 stop link retransmit timer
OVPN 0 got incoming ciphertext len 1450
Send DA_IN to SSL ID 35 len 1450
SSL_connect:SSLv3 read server hello A

OVPN 0 got incoming ciphertext len 614
Send DA_IN to SSL ID 35 len 614
```

```
SSL_connect:SSLv3 read server certificate A

SSL_connect:SSLv3 read server done A

SSL_connect:SSLv3 write client key exchange A

SSL_connect:SSLv3 write change cipher spec A

SSL_connect:SSLv3 write finished A

SSL_connect:SSLv3 flush data

OVPN 0 stop link retransmit timer
OVPN 0 got incoming ciphertext len 226
Send DA_IN to SSL ID 35 len 226
SSL_connect:SSLv3 read server session ticket A

SSL_connect:SSLv3 read finished A


---
Certificate chain
 0
s:/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=server/name=changeme/emailAddress=support
@digi.com
   i:/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=OpenVPN-
CA/name=changeme/emailAddress=support@digi.com
 1 s:/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=OpenVPN-
CA/name=changeme/emailAddress=support@digi.com
   i:/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=OpenVPN-
CA/name=changeme/emailAddress=support@digi.com
---
Peer certificate
subject=/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=server/name=changeme/emailAddress=s
upport@digi.com
issuer=/C=DE/ST=BY/L=Munich/O=Digi/OU=support/CN=OpenVPN-
CA/name=changeme/emailAddress=support@digi.com
---
No client certificate CA names sent
---
SSL handshake has read 2290 bytes and written 529 bytes
---
New, TLSv1/SSLv3, Cipher is AES256-GCM-SHA384
Server public key is 1024 bit
SSL-Session:
                Protocol  : TLSv1.2
                                    Cipher    : AES256-GCM-SHA384
                                                                    Session-ID:
5AC99A6068E027B1692ADA5CE3F66F3376ACDF433F501EC9655F26DF05920D7D

Session-ID-ctx:

Master-Key:
```

```
42A8DB3D0B6F7C3D42EF882694B23812F8686F063BCF73A473D72A569CB5A29C40FB84BC370BFF38774BC
93495B3F6A5
             Key-Arg   : None
                                PSK identity: None
                                                      PSK identity hint: None
                                                                                    SRP
username: None

TLS session ticket lifetime hint: 300 (seconds)

TLS session ticket:

0000 - 6f 2a 8c e3 53 81 2f 87-c4 7a 10 33 f7 fc 57 d9   o*..S./..z.3..W.

0010 - 31 1f ec 2c 3e 90 01 e9-a8 d6 00 45 df 4f 78 47   1..,>......E.OxG
                                                             0020 - ca 72 89 86 85 a2
e9 79-82 72 38 fa 4b b1 9b 61   .r.....y.r8.K..a

0030 - dd ae 42 5c 49 3d db 71-a0 b6 a1 0a e5 34 91 cd   ..B\I=.q.....4..

0040 - b4 03 f1 39 40 c7 8f 9d-ff e8 3d b9 1a 7f 8d 74   ...9@.....=....t
                      0050 - 7e df d2 e6 34 75 cc 83-0f 59 69 d7 44 f1 b3 89
~...4u...Yi.D...

0060 - 87 be 9c cf 3a 84 dc 30-a7 a2 30 f3 fe 93 2b 26   ....:..0..0...+&

0070 - 35 89 43 f1 e0 11 b2 fd-0d 90 8e 8c 48 c2 f5 65   5.C.........H..e

0080 - da 10 ce 1d 05 40 6d 72-00 3d 1f 8f ca ac ff ae   .....@mr.=......
                                                          0090 - 2b b6 c5 f2 ea ef b7
e5-ca 18 60 01 32 ca 0e 4f   +.........`.2..O


Start Time: 1501597976

Timeout   : 300 (sec)

Verify return code: 0 (ok)

---

OVPN 0 stop link retransmit timer
OVPN 0 got incoming ciphertext len 117
Send DA_IN to SSL ID 35 len 117
Init cipher AES-256-CBC (ctx 4359e9c8)
Cipher 'AES-256-CBC' initialized with 256 bit key
HMAC authentication using 128 bit MD5
Init cipher AES-256-CBC (ctx 4359eb3c)
Cipher 'AES-256-CBC' initialized with 256 bit key
HMAC authentication using 128 bit MD5
OVPN 0 KS negotiation complete
PUSH_REQUEST is required
Send DA_RQ to SSL ID 35 len 13
OVPN 0 stop link retransmit timer
```

```
OVPN 0 got incoming ciphertext len 104
Send DA_IN to SSL ID 35 len 104
OVPN 0 got PUSH_REPLY
OVPN 0 process option ifconfig 192.168.0.2 192.168.0.1
OVPN 0 got ifconfig option. His IP: 192.168.0.1, our IP 192.168.0.2
OVPN 0 process option route 172.16.0.0 255.255.255.0
OVPN 0 got route option. Add route 172.16.0.0/255.255.255.0
OVPN 0 stop negotiation timer
OVPN 0 UP. IP address: 192.168.0.2
OVPN 0 add route 172.16.0.0/24
OVPN 0 RX pkt HMAC error
OVPN 0 decrypt error
OVPN 0 decrypt failed
OVPN 0 attempted decrypt using key id 0 at index 0 failed
OVPN 0 key not ready using key id 0 at index 1 (state 0)
OVPN 0 key not ready using key id 0 at index 2 (state 0)
OVPN 0 no matching session decrypt key ID found for key ID 0
OVPN 0 unable to process RX Data packet
OVPN 0 RX data pkt error
OVPN 0 remove route 172.16.0.0/24
OVPN 0 DOWN
deact SSL 35
Cleanup cipher AES-256-CBC (ctx 4359e9c8)
Cleanup cipher AES-256-CBC (ctx 4359eb3c)
OVPN 0 closing link socket
OVPN 0 link socket disconnected
OVPN 0 start socket idle timer
```

So, the OVPN seems going UP at the beginning but as soon as it receives the first packet it got a decrypt error and it goes down.

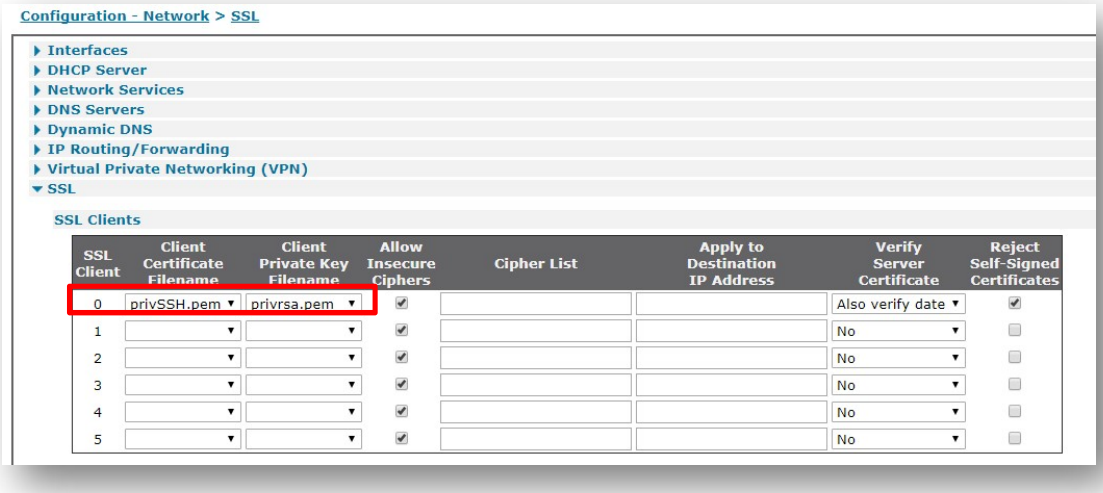
On the eventlog what shown is only the OVPN going UP/DOWN:

```
14:33:08, 01 Aug 2017,OVPN 0 Out Of Service,Activation
14:33:08, 01 Aug 2017,OVPN 0 down
14:32:56, 01 Aug 2017,OVPN 0 Available,Activation
14:32:56, 01 Aug 2017,OVPN 0 up
```

2.3.3 SSL Certs error

Another error that can be done when configuring an OpenVPN connection, is to use wrong certificates/keys files.

In the example below, on the Client has been selected the wrong files:

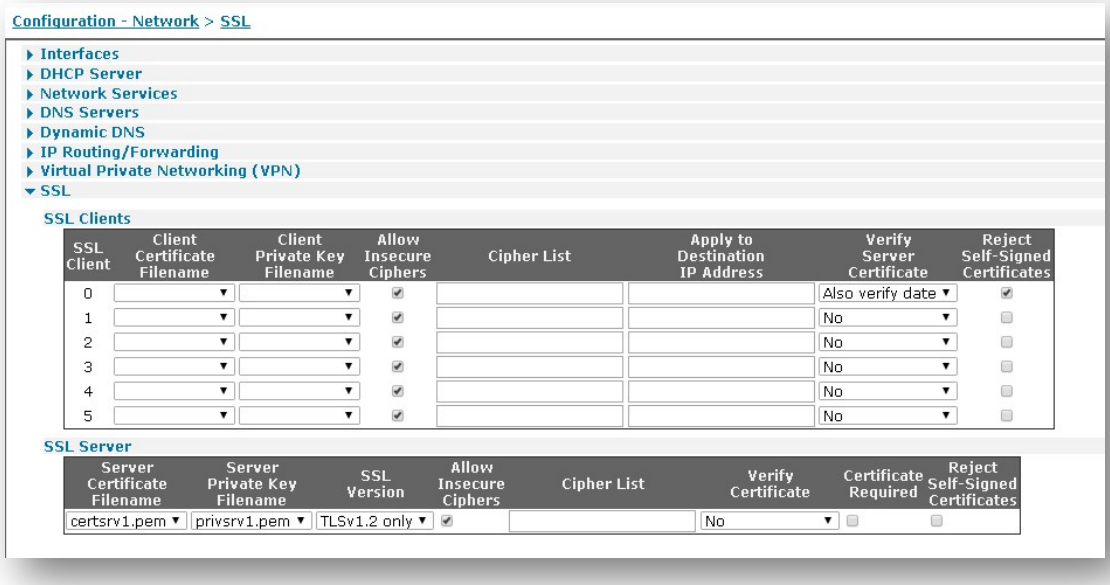



```
OVPN 0 init ctx
OVPN 0 start key negotiation
OVPN 0 start negotiation timer
OVPN 0 negotiation waiting on initial ACKS to complete
SSL error
 Lib: SSL routines
 Func: SSL_CTX_use_certificate_file
 Reason: BUF lib
unable to get certificate from 'privSSH.pem'
Send DA_RQ to SSL ID 3 len 136
OVPN 0 stop link retransmit timer
OVPN 0 unable to process control packet
OVPN 0 RX data pkt error
OVPN 0 DOWN
OVPN 0 stop negotiation timer
OVPN 0 closing link socket
OVPN 0 link socket disconnected
```

The SSL debug will show an error in the certificate usage. In this case, almost the same is shown also in the eventlog:

```
09:17:45, 02 Aug 2017,OVPN 0 down
09:17:45, 02 Aug 2017,Certificate Code Error
                      Lib: SSL routines
                      Func: SSL_CTX_use_certificate_file
                      Reason: BUF lib
```

Another case can be that on the Server is used a Certifcates/Key pair not corresponding to the CA used from the client:



The debug on the client will show the following:

```
OVPN 0 link socket connected
OVPN 0 init ctx
OVPN 0 start key negotiation
OVPN 0 start negotiation timer
OVPN 0 negotiation waiting on initial ACKS to complete
Client cipher list: SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-
CBC-SHA:DH-DSS-AES256-GCM-SHA384:DHE-DSS-AES256-GCM-SHA384:DH-RSA-AES256-GCM-SHA
384:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA256:DH-RSA
-AES256-SHA256:DH-DSS-AES256-SHA256:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:DH-RSA
-AES256-SHA:DH-DSS-AES256-SHA:AES256-GCM-SHA384:AES256-SHA256:AES256-SHA:PSK-AES
256-CBC-SHA:SRP-DSS-AES-128-CBC-SHA:SRP-RSA-AES-128-CBC-SHA:SRP-AES-128-CBC-SHA:
DH-DSS-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DH-RSA-AES128-GCM-SHA256:DHE-
```

```
RSA-AES128-GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-DSS-AES128-SHA256:DH-RSA-AES128-
SHA256:DH-DSS-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:DH-RSA-AES128-
SHA:DH-DSS-AES128-SHA:AES128-GCM-SHA256:AES128-SHA256:AES128-SHA:PSK-AES128-CBC-
SHA
SSL_connect:before/connect initialization
SSL_connect:SSLv2/v3 write client hello A
Send DA_RQ to SSL ID 1 len 136
OVPN 0 stop link retransmit timer
OVPN 0 stop link retransmit timer
OVPN 0 got incoming ciphertext len 1114
Send DA_IN to SSL ID 1 len 1114
SSL_connect:SSLv3 read server hello A
SSL error
 Lib: SSL routines
 Func: ssl3_get_server_certificate
 Reason: certificate verify failed
SSL3 alert write:fatal:unknown CA
SSL_connect:error in error
SSL_connect:error in error
OVPN 0 unable to process control packet
OVPN 0 RX data pkt error
OVPN 0 DOWN
OVPN 0 stop negotiation timer
OVPN 0 stop link retransmit timer
OVPN 0 closing link socket
OVPN 0 link socket disconnected
OVPN 0 start socket idle timer
OVPN 0 link socket idle timeout
OVPN 0 connect socket
OVPN 0 link socket connected
OVPN 0 init ctx
OVPN 0 start key negotiation
OVPN 0 start negotiation timer
OVPN 0 link socket disconnected
OVPN 0 DOWN
OVPN 0 stop negotiation timer
OVPN 0 stop link retransmit timer
OVPN 0 start socket idle timer
OVPN 0 link socket idle timeout
OVPN 0 connect socket
OVPN 0 link socket connected
OVPN 0 init ctx
OVPN 0 start key negotiation
OVPN 0 start negotiation timer
OVPN 0 link socket disconnected
OVPN 0 DOWN
OVPN 0 stop negotiation timer
OVPN 0 stop link retransmit timer
OVPN 0 start socket idle timer
OVPN 0 link socket idle timeout
OVPN 0 connect socket
OVPN 0 link socket connected
OVPN 0 init ctx
OVPN 0 start key negotiation
```

```
OVPN 0 start negotiation timer
OVPN 0 link socket disconnected
OVPN 0 DOWN
OVPN 0 stop negotiation timer
OVPN 0 stop link retransmit timer
OVPN 0 start socket idle timer
OVPN 0 link socket idle timeout
```

So, a “certificate verify alert” is shown (also in the eevntlog) and the reason sent by the server is “Uknown CA”

```
10:21:27, 02 Aug 2017,OVPN 0 down
10:21:24, 02 Aug 2017,OVPN 0 down
10:21:20, 02 Aug 2017,OVPN 0 down
10:21:17, 02 Aug 2017,OVPN 0 down
10:21:14, 02 Aug 2017,OVPN 0 down
10:21:10, 02 Aug 2017,OVPN 0 down
10:21:06, 02 Aug 2017,OVPN 0 down
10:21:03, 02 Aug 2017,OVPN 0 down
10:21:00, 02 Aug 2017,OVPN 0 down
10:20:56, 02 Aug 2017,OVPN 0 down
10:20:52, 02 Aug 2017,OVPN 0 down
10:20:49, 02 Aug 2017,OVPN 0 down
10:20:45, 02 Aug 2017,OVPN 0 down
10:20:42, 02 Aug 2017,OVPN 0 down
10:20:38, 02 Aug 2017,OVPN 0 down
10:20:34, 02 Aug 2017,OVPN 0 down
10:20:31, 02 Aug 2017,OVPN 0 down
10:20:27, 02 Aug 2017,OVPN 0 down
10:20:27, 02 Aug 2017,Certificate Code Error
                      Lib: SSL routines
                      Func: ssl3_get_server_certificate
                      Reason: certificate verify failed
```

In this case in useful also to check the server debug that will show the “tlsv1 alert unknown ca” error:

```
SSL error
 Lib: SSL routines
 Func: ssl3_read_bytes
 Reason: tlsv1 alert unknown ca
```

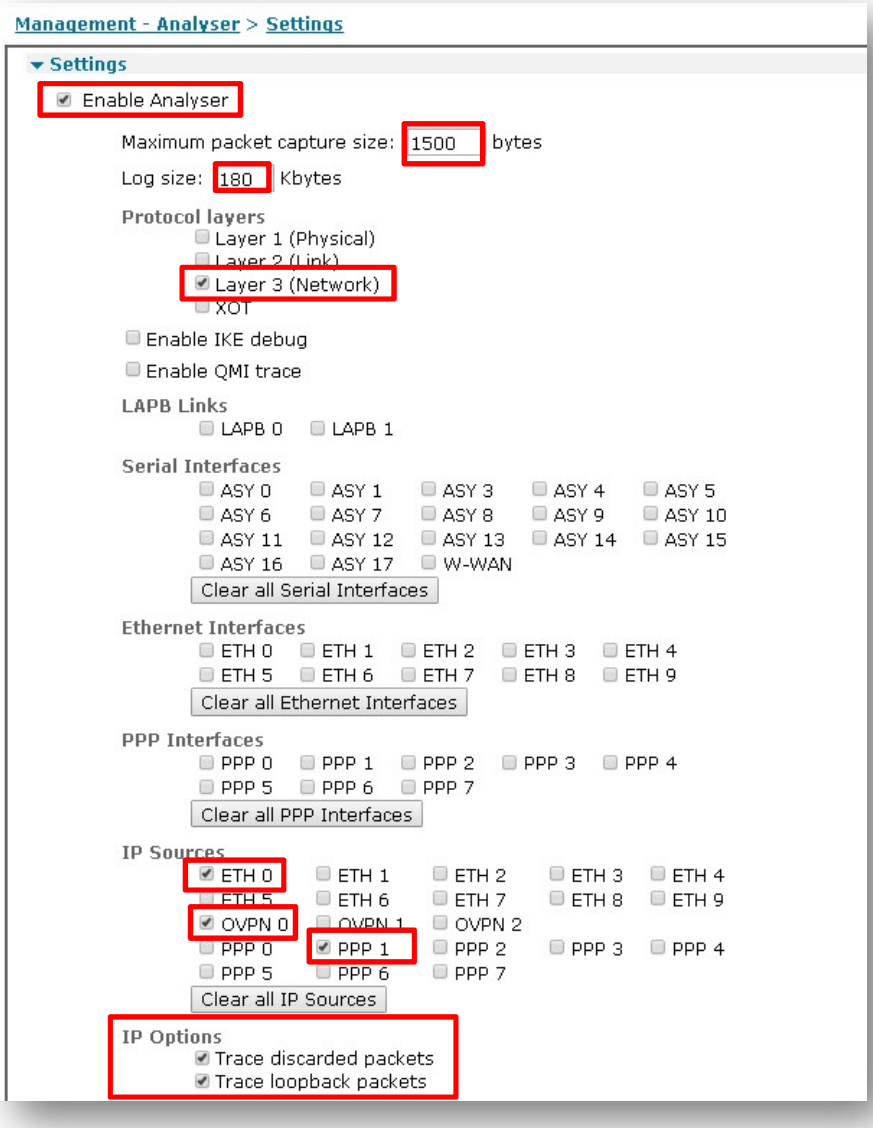
```
10:21:26, 02 Aug 2017,OVPN 0 down
10:21:10, 02 Aug 2017,GP socket connected: 37.83.252.71:22 ->
193.201.224.199:34900
10:20:43, 02 Aug 2017,GP socket connected: 37.83.252.71:22 ->
193.201.224.199:46635
10:20:27, 02 Aug 2017,Certificate Code Error
                      Lib: SSL routines
                      Func: ssl3_read_bytes
                      Reason: tlsv1 alert unknown ca
```

2.4 Troubleshooting the Traffic through an OpenVPN connection

If the OpenVPN connection is UP and stable, the desired traffic should pass through it without issues.

In case this doesn't happen, the traffic flow can be checked configuring the analyser as shown below. Also in this case, the client side will be shown, but it will be the same on the server side:

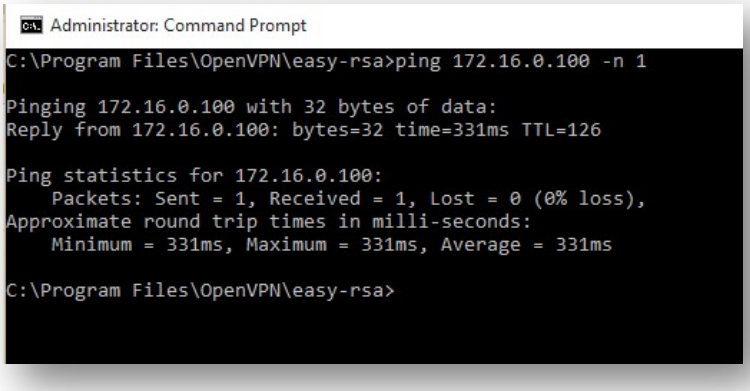
MANAGEMENT - ANALYSER > SETTINGS



Where:

Parameter	Setting	Description
Enable Analyser	✓	If ticked will reveal all Analyser settings options
Maximum packet capture size	1500	The number of bytes that are captured and stored for each packet. If the packet is bigger than the configured size, the packet is truncated. The usual value used is 1500
Log Size	180	The maximum size of the pseudo file **ana.txt** for storing the captured data packets. Once the maximum size is reached, the oldest captured data packets are overwritten when new packets are captured. Usually the maximum value is chosen: 180Kb (but the data is compressed so more than 180Kb of trace data will be captured)
Protocol layers	Layer 3 (Network)	The check-boxes under this heading specify which protocol layers are captured and included in the Analyser trace. In this case the the Network Layer (Layer 3) is chosen.
IP Sources		Selects the IP sources over which packets are captured and included in the Analyser trace. These sources include IP packet transmitted and received over Ethernet, PPP and OpenVPN (OVPN) interfaces.
ETH 0	✓	LAN Interface , in this example ETH 0 is used
OVPN 0	✓	Open VPN interface used, in this case in “0”
PPP 1	✓	WAN Interface, in this example PPP 1 is used
IP Options		Enabling the options below is useful in order to check if the traffic is discarded for any reason
Trace discarded packets	✓	Enables or disables the capture of packets that are discarded by an interface, and displays a reason for why the packet was discarded.
Trace loopback packets	✓	Enables or disables the capture of IP loopback packets

Apply the changes and make a ping from a laptop in Client LAN to a device in the Server LAN:



If all is configured correctly on both sides and network elements, the ping should success, in case of issues, the **ana.txt** can be checked by downloading it from the router or checking its content in Management - Analyser > Trace section of the WEB UI.

Below, a “good”trace is shown as well as troubleshooting tips in case of not expected behaviour:

- A Packet arrives on the OVPN interface via PPP connection, and it is revealed as an ECHO REQ for the client Laptop from a server LAN one:

```
-----   2-8-2017  11:53:24.610   -------
 45 00 00 91 02 AE 00 00 F4 11 EA B2 25 53 FC 47     E...........%S.G
 25 54 92 0C 04 AA 1C 14 00 7D BB 89 30 74 CC 9E     %T.......}..0t..
 7E 4C BF 45 0E BB 70 02 37 DE 71 CF 5B 06 41 B1     ~L.E..p.7.q.[.A.
 A4 38 49 6E 96 ED D5 E3 2A 49 80 7C 97 9E AF 0F     .8In....*I.|....
 7A 88 CD 26 16 56 50 B4 E9 13 03 8D 31 4D 8E E7     z..&.VP.....1M..
 B4 07 EF 2C 4F 3B C8 0F 18 8F D1 39 9B 65 9C 9D     ...,O;.....9.e..
 06 36 51 86 12 C5 C6 BD 30 E0 29 36 91 6A 5B B6     .6Q.....0.)6.j[.
 26 FD EE F9 23 26 A5 0A 4B 38 FE 5D 1A AB 0C EB     &...#&..K8.]....
 C2 EF 9B 1D A3 40 2A 23 A2 99 F5 B3 38 F6 A8 A1     .....@*#....8...
 59                                                  Y

 IP (In) From REM TO LOC       IFACE: PPP 1
 45             IP Ver:        4
                Hdr Len:       20
 00             TOS:           Routine
                Delay:         Normal
                Throughput:    Normal
                Reliability:   Normal
 00 91          Length:        145
 02 AE          ID:            686
 00 00          Frag Offset:   0
                Congestion:    Normal
                               May Fragment
                               Last Fragment
 F4             TTL:           244
 11             Proto:         UDP
 EA B2          Checksum:      60082
 25 53 FC 47    Src IP:        37.83.252.71
 25 54 92 0C    Dst IP:        37.84.146.12
UDP:
 04 AA          SRC Port:      ??? (1194)
 1C 14          DST Port:      ??? (7188)
 00 7D          Length:        125
 BB 89          Checksum:      48009
 ----------
 -----   2-8-2017  11:53:24.610   -------
 45 00 00 3C 02 BC 00 00 7F 01 DF 1C AC 10 00 64     E..<...........d
 AC 10 01 64 08 00 4D 37 00 01 00 24 61 62 63 64     ...d..M7...$abcd
 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74     efghijklmnopqrst
 75 76 77 61 62 63 64 65 66 67 68 69                 uvwabcdefghi

 IP (In) From REM TO LOC       IFACE: OVPN 0
```

```
   45             IP Ver:        4
                  Hdr Len:       20
   00             TOS:           Routine
                  Delay:         Normal
                  Throughput:    Normal
                  Reliability:   Normal
   00 3C          Length:        60
   02 BC          ID:            700
   00 00          Frag Offset:   0
                  Congestion:    Normal
                                 May Fragment
                                 Last Fragment
   7F             TTL:           127
   01             Proto:         ICMP
   DF 1C          Checksum:      57116
   AC 10 00 64    Src IP:        172.16.0.100
   AC 10 01 64    Dst IP:        172.16.1.100
   ICMP:
   08             Type:          ECHO REQ
   00             Code:          0
   4D 37          Checksum:      14157
   ----------
```

Please note: If the above packet is not received (with analyser correctly configured), the Client-Server conenction should be checked on both sides.

- The ECHO REQ is routed to the laptop via the ETH 0 interface:

```
-----   2-8-2017  11:53:24.610   -------
45 00 00 3C 02 BC 00 00 7E 01 E0 1C AC 10 00 64     E..<....~......d
AC 10 01 64 08 00 4D 37 00 01 00 24 61 62 63 64     ...d..M7...$abcd
65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74     efghijklmnopqrst
75 76 77 61 62 63 64 65 66 67 68 69                 uvwabcdefghi

IP (Final) From LOC TO REM    IFACE: ETH 0
45             IP Ver:        4
               Hdr Len:       20
00             TOS:           Routine
               Delay:         Normal
               Throughput:    Normal
               Reliability:   Normal
00 3C          Length:        60
02 BC          ID:            700
00 00          Frag Offset:   0
               Congestion:    Normal
                              May Fragment
                              Last Fragment
7E             TTL:           126
01             Proto:         ICMP
```

```
E0 1C          Checksum:      57372
AC 10 00 64    Src IP:        172.16.0.100
AC 10 01 64    Dst IP:        172.16.1.100
ICMP:
08             Type:          ECHO REQ
00             Code:          0
4D 37          Checksum:      14157
----------
```

Please note: If the above packet is not shown, the ETH 0 connection on the TransPort should be checked, it can be down, disconnected or wrongly configured.

- The ECHO REPLY from the Client laptop arrives on ETH 0 interface:

```
-----   2-8-2017  11:53:24.610   -------
45 00 00 3C 6D 54 00 00 80 01 73 84 AC 10 01 64     E..<mT....s....d
AC 10 00 64 00 00 55 37 00 01 00 24 61 62 63 64     ...d..U7...$abcd
65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74     efghijklmnopqrst
75 76 77 61 62 63 64 65 66 67 68 69                 uvwabcdefghi

IP (In) From REM TO LOC       IFACE: ETH 0
45             IP Ver:        4
               Hdr Len:       20
00             TOS:           Routine
               Delay:         Normal
               Throughput:    Normal
               Reliability:   Normal
00 3C          Length:        60
6D 54          ID:            27988
00 00          Frag Offset:   0
               Congestion:    Normal
                              May Fragment
                              Last Fragment
80             TTL:           128
01             Proto:         ICMP
73 84          Checksum:      29572
AC 10 01 64    Src IP:        172.16.1.100
AC 10 00 64    Dst IP:        172.16.0.100
ICMP:
00             Type:          ECHO REPLY
00             Code:          0
55 37          Checksum:      14165
----------
```

Please note: If the above packet is not shown, then the client laptop and routing configuration should be checked. Most probably can be a switch/ conenction issue on the LAN or a default gateway missing on the client laptop configuration.

- The ECHO REPLY is then sent out to the OVPN 0 interface via the PPP connection:

```
-----   2-8-2017  11:53:24.610   -------
45 00 00 3C 6D 54 00 00 7F 01 74 84 AC 10 01 64     E..<mT....t....d
AC 10 00 64 00 00 55 37 00 01 00 24 61 62 63 64     ...d..U7...$abcd
65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74     efghijklmnopqrst
75 76 77 61 62 63 64 65 66 67 68 69                 uvwabcdefghi

IP (Final) From LOC TO REM    IFACE: OVPN 0
45             IP Ver:        4
               Hdr Len:       20
00             TOS:           Routine
               Delay:         Normal
               Throughput:    Normal
               Reliability:   Normal
00 3C          Length:        60
6D 54          ID:            27988
00 00          Frag Offset:   0
               Congestion:    Normal
                              May Fragment
                              Last Fragment
7F             TTL:           127
01             Proto:         ICMP
74 84          Checksum:      29828
AC 10 01 64    Src IP:        172.16.1.100
AC 10 00 64    Dst IP:        172.16.0.100
ICMP:
00             Type:          ECHO REPLY
00             Code:          0
55 37          Checksum:      14165
----------
-----   2-8-2017  11:53:24.610   -------
45 00 00 91 05 9E 00 00 FA 11 E1 C2 25 54 92 0C     E...........%T..
25 53 FC 47 1C 14 04 AA 00 7D 80 02 30 9B 23 24     %S.G.....}..0.#$
40 55 4B 8F 37 CE B3 4E 0F 6A F5 BD 59 BB A2 9A     @UK.7..N.j..Y...
34 BE 01 1E FB D9 A0 D6 9A E7 F1 EB 6E AF 67 78     4...........n.gx
1D 1C 73 27 F3 3A CA B2 F6 AB 79 22 47 A3 D1 EF     ..s'.:....y"G...
1F 65 43 35 74 1C 2B 69 47 9A DA 64 2F 2B 36 B9     .eC5t.+iG..d/+6.
10 2E 8B CF B3 78 6B 6F D4 3F D4 8A B5 44 40 58     .....xko.?...D@X
FD 55 34 25 75 C8 34 9D 84 28 45 89 E4 80 4B 20     .U4%u.4..(E...K
AC EE 9E 1E 70 17 0D 6E AC 8B A8 27 4C 2D D7 E5     ....p..n...'L-..
B2                                                  .

IP (Final) From LOC TO REM    IFACE: PPP 1
45             IP Ver:        4
               Hdr Len:       20
00             TOS:           Routine
               Delay:         Normal
```

```
               Throughput:    Normal
               Reliability:   Normal
00 91          Length:        145
05 9E          ID:            1438
00 00          Frag Offset:   0
               Congestion:    Normal
                              May Fragment
                              Last Fragment
FA             TTL:           250
11             Proto:         UDP
E1 C2          Checksum:      57794
25 54 92 0C    Src IP:        37.84.146.12
25 53 FC 47    Dst IP:        37.83.252.71
UDP:
1C 14          SRC Port:      ??? (7188)
04 AA          DST Port:      ??? (1194)
00 7D          Length:        125
80 02          Checksum:      32770
----------
```

Please note: If the above packets are not shown, the PPP connection/OVPN interface status and the routing table should be checked on the client router.

3 CONFIGURATION FILES

The main configuration part added to the Client router in order to debug the OpenVPN connection establishment and traffic is shown below:

```
eth 0 ipanon ON
ppp 1 ipanon ON
ana 0 anon ON
ana 0 l2on OFF
ana 0 xoton OFF
ana 0 lapdon 0
ana 0 lapbon 0
ana 0 loopon ON
ana 0 discardson ON
ana 0 maxdata 1500
ana 0 logsize 180

sslcli 0 debug ON
ovpn 0 debug ON
```