Application Note

OpenVPN Tunnel

Advantech Czech s.r.o., Sokolska 71, 562 04 Usti nad Orlici, Czech Republic
Document No. APP-0007-EN, revised on March 26, 2026.

© 2026 Advantech Czech s.r.o. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photography, recording, or any information storage and retrieval system, without prior written consent. Information in this manual is subject to change without notice and does not represent a commitment by Advantech.

Advantech Czech s.r.o. shall not be liable for any incidental or consequential damages arising from the use, performance, or furnishing of this manual.

All brand names used in this manual are registered trademarks of their respective owners. The use of trademarks or other designations in this publication is for reference purposes only and does not imply endorsement by the trademark holder.

Used symbols

Important

Important — Indicates a risk to personal safety or potential damage to the router. Follow these instructions precisely to prevent injury or equipment damage.

Warning

Warning — Highlights conditions that may cause malfunction, loss of data, or unexpected behavior in specific situations. Read carefully before proceeding.

Info

Info — Provides helpful tips, context, or references that improve understanding but are not strictly required to complete the task.

Code Example

```
Code Example — Copy-pasteable configuration snippets or CLI commands.
```

Contents

1. OpenVPN protocol **1**

1.1 Compatibility Notes . . . 2
1.2 Restrictions in Advantech routers . . . 2

2. Configuration of OpenVPN tunnel **3**

3. Router on both sides of tunnel **7**

3.1 OpenVPN tunnel without authentication . . . 7
3.2 OpenVPN tunnel with pre-shared secret authentication . . . 10
3.3 OpenVPN tunnel with username/password authentication . . . 13
3.4 OpenVPN tunnel with X.509 certificate authentication . . . 16

4. Tunnel against WIN/Linux CLIENT **19**

4.1 OpenVPN tunnel configuration on the router . . . 19
4.2 OpenVPN tunnel configuration on WIN/Linux client . . . 22

5. Tunnel against WIN/Linux SERVER **23**

5.1 OpenVPN tunnel configuration on the router . . . 23
5.2 Tunnel configuration on Computer 1 – Server . . . 26

6. Multiclient-Server – Router (CLIENT) **27**

6.1 OpenVPN tunnel configuration on Advantech routers . . . 28
6.2 OpenVPN server configuration . . . 29

7. Multiclient-Server – Router (CL. to CL.) **30**

7.1 OpenVPN server configuration . . . 31
7.2 OpenVPN tunnel configuration on Advantech routers . . . 32

8. Creation of pre-shared key in Windows **34**

9. Creation of certificates in Windows **35**

9.1 Introduction . . . 35
9.2 Generating of certificates . . . 35
9.3 Overview of the generated files . . . 36

10. Related Resources **37**

Appendix A: Installation of OpenVPN Windows **A1**

Appendix B: Installation of Easy-RSA on Windows **B1**

List of Figures

1 Basic scheme . . . 1
2 Configuration form for OpenVPN tunnel . . . 6
3 Router on both sides of tunnel . . . 7
4 Configuration of the first router – SERVER (no authentication) . . . 8
5 Network Status . . . 9
6 System log . . . 9
7 Configuration of the first router – SERVER (pre-shared secret) . . . 11
8 Network Status . . . 12
9 System log . . . 12
10 Configuration of the first router – SERVER (username/password) . . . 14
11 Network Status . . . 15
12 System log . . . 15
13 Configuration of the first router – SERVER (X.509 certificate) . . . 17
14 Network Status . . . 18
15 System log . . . 18
16 OpenVPN tunnel against Windows/Linux CLIENT . . . 19
17 Router configuration . . . 20
18 Network Status . . . 21
19 System log . . . 21
20 OpenVPN tunnel against Windows/Linux SERVER . . . 23
21 Router configuration . . . 24
22 Network Status . . . 25
23 System log . . . 25
24 OpenVPN Multiserver – Advantech router (CLIENT) . . . 27
25 Configuration of Advantech router . . . 28
26 OpenVPN client to client . . . 30
27 Advantech router configuration . . . 32
28 Network Status . . . 33
29 System log . . . 33
30 Generating a pre-shared key . . . 34
31 Installation of OpenVPN – basic information . . . A1
32 Installation of OpenVPN – components . . . A2
33 Installation of OpenVPN – progress . . . A2
34 Installation of OpenVPN – complete . . . A3

List of Tables

1 OpenVPN Configuration . . . 3
2 Configuration of the first router (no authentication) . . . 7
3 Configuration of the second router (no authentication) . . . 7
4 Configuration of the first router (pre-shared secret) . . . 10
5 Configuration of the second router (pre-shared secret) . . . 10
6 Configuration of the router (username/password) . . . 13
7 Configuration of the first router (X.509 certificate) . . . 16
8 Configuration of the second router (X.509 certificate) . . . 16

9 Router configuration 19
10 Router configuration 23
11 Overview of the generated files 36

1. OpenVPN protocol

OpenVPN (Open Virtual Private Network) is a means of interconnecting several computers over an untrusted public network. It enables connected computers to communicate with each other as if they were connected in a single closed private network (this network is consequently trusted). Using client-server architecture, OpenVPN is capable of ensuring a direct connection between computers behind NAT without any need to configure NAT. It has a few ways to authenticate clients – using a pre-shared key, a certificate or a username and password.

OpenVPN uses the officially assigned port 1194, which is applied as a default in newer versions. It offers two types of network interfaces (Universal TUN and TAP driver), which enable creation of an IP tunnel (TUN) on the third layer of the ISO/OSI or on the second layer (layer-2 Ethernet TAP), which is able to transmit any type of data. OpenVPN uses common network protocols (TCP and UDP) and thus creates an alternative to IPsec protocol.

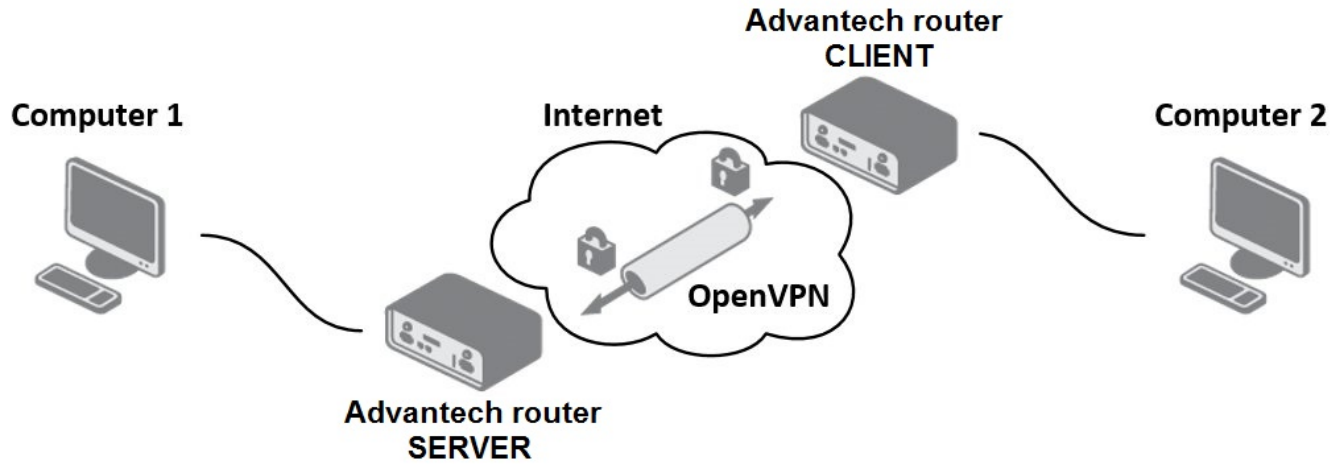

Figure 1: Basic scheme

1.1 Compatibility Notes

In firmware version 6.4.0, we updated the OpenVPN software from version 2.4.12 to 2.6.6. Additionally, the OpenSSL library has been upgraded from the previous version 1.1.1, which will no longer be supported after September 11, 2023, to the new version 3.0.11. These updates entail the following compatibility notes:

- If you are using version 2.4 of OpenVPN on the remote tunnel side, you may encounter issues. We recommend upgrading these clients to OpenVPN version 2.6 or newer. For older versions, consider adding a specific configuration in *Extra Options*, such as `--cipher AES-256-CBC`. If unsure, use the AES-256-CBC cipher on the remote tunnel side.
- A general overview of deprecated features is available on the *Deprecated Options in OpenVPN* website.
- For details on expected behavior in OpenVPN Cipher Negotiation between common configurations of OpenVPN servers and clients, visit the *OpenVPN Cipher Negotiation* page.
- **Option –comp-lzo**: Compression is not recommended and should be avoided. This option is discouraged and considered deprecated. Starting from version 2.5, this option will no longer enable compression, only the compression framing, to allow for receiving compressed packets.
- There are two possible solutions if you encounter the "OpenSSL: error:0A00018E:SSL routines::ca md too weak" issue:
  1. **Secure Option**: Regenerate the certificate using the Signature Hash Algorithm SHA256 or better. OpenSSL 3's default settings are adequate for this process. For more detailed information, refer to Chapter 9.
  2. **Less Secure Option**: Add a specific configuration in *Extra Options*: `--tls-cipher "DEFAULT:@SECLEVEL=0"`.

1.2 Restrictions in Advantech routers

- Routers allow to create up to four OpenVPN tunnels simultaneously
- Routers support TUN and TAP adapters
- Routers can not be used as a multiclient server

2. Configuration of OpenVPN tunnel

Warning

The OpenVPN server must have a known IP address accessible by the client(s).

OpenVPN tunnel allows protected connection of four networks LAN to the one network. To open the *OpenVPN* tunnel configuration page, click *OpenVPN* in the *Configuration* section of the main menu. The menu item will expand and you will see four separate configuration pages: *1st Tunnel*, *2nd Tunnel*, *3rd Tunnel* and *4th Tunnel*. Description of all items is listed in following table.

Info

In v3, v4 and v4i routers, the IPv4 and IPv6 tunnels are supported. In v2 routers, only IPv4 tunnels are supported.

Item	Description
Create 1st|2nd|3rd|4th OpenVPN tunnel	If enabled, the tunnel is activated.
Description	Specifies the description or name of tunnel.
Interface Type	TAP is basically at the Ethernet level (layer 2) and acts as a switch, whereas TUN works at the network level (layer 3) and routes packets on the VPN. TAP is bridging, whereas TUN is routing. • **TUN** – Choose the TUN mode. • **TAP** – Choose the TAP mode, but remember first to **configure the bridge** on the ethernet interface.
Protocol	Specifies the communication protocol. • **UDP** – The OpenVPN communicates using UDP. • **TCP server** – The OpenVPN communicates using TCP in server mode. • **TCP client** – The OpenVPN communicates using TCP in client mode. • **UDPv6** – The OpenVPN communicates using UDP over IPv6. • **TCPv6 server** – The OpenVPN communicates using TCP over IPv6 in server mode. • **TCPv6 client** – The OpenVPN communicates using TCP over IPv6 in client mode.
UDP/TCP port	Specifies the port of the relevant protocol (UDP or TCP).
1st Remote IP Address	Specifies the first IPv4, IPv6 address or domain name of the opposite side of the tunnel.
2nd Remote IP Address	Specifies the second IPv4, IPv6 address or domain name of the opposite side of the tunnel.
Remote Subnet	IPv4 address of a network behind opposite side of the tunnel.
Remote Subnet Mask	IPv4 subnet mask of a network behind opposite tunnel's side.
Redirect Gateway	Adds (rewrites) the default gateway. All the packets are then sent to this gateway via tunnel, if there is no other specified default gateway inside them.

Table 1: OpenVPN Configuration

Item	Description
Local Interface IP Address	Specifies the IPv4 address of a local interface. **For proper routing it is recommended to fill-in any IPv4 address from local range even if you are using IPv6 tunnel only.**
Remote Interface IP Address	Specifies the IPv4 address of the interface of opposite side of the tunnel. **For proper routing it is recommended to fill-in any IPv4 address from local range even if you are using IPv6 tunnel only.**
Remote IPv6 Subnet	IPv6 address of the remote IPv6 network. Equivalent of the *Remote Subnet* in IPv4 section.
Remote IPv6 Prefix	IPv6 prefix of the remote IPv6 network. Equivalent of the *Remote Subnet Mask* in IPv4 section.
Local Interface IPv6 Address	Specifies the IPv6 address of a local interface.
Remote Interface IPv6 Address	Specifies the IPv6 address of the interface of opposite side of the tunnel.
Ping Interval	Time interval after which the router sends a message to opposite side of tunnel to verify the existence of the tunnel.
Ping Timeout	Specifies the time interval the router waits for a message sent by the opposite side. For proper verification of the OpenVPN tunnel, set the *Ping Timeout* to greater than the *Ping Interval*.
Renegotiate Interval	Specifies the renegotiate period (reauthorization) of the OpenVPN tunnel. You can only set this parameter when the *Authenticate Mode* is set to *username/-password* or *X.509 certificate*. After this time period, the router changes the tunnel encryption to keep the tunnel secure.
Max Fragment Size	Maximum size of a sent packet.
Compression	Compression of the data sent: • **none** – No compression is used. • **LZO** – A lossless compression is used, use the same setting on both sides of the tunnel. **Deprecated scheduled for removal!**
NAT Rules	Activates/deactivates the NAT rules for the OpenVPN tunnel: • **not applied** – NAT rules are not applied to the tunnel. • **applied** – NAT rules are applied to the OpenVPN tunnel.
Authenticate Mode	Specifies the authentication mode: • **none** – No authentication is set. **Deprecated scheduled for removal!** • **Pre-shared secret** – Specifies the shared key function for both sides of the tunnel. **Deprecated scheduled for removal!** • **Username/password** – Specifies authentication using a CA Certificate, Username and Password. **Deprecated scheduled for removal!** • **X.509 Certificate (multiclient)** – Activates the X.509 authentication in multi-client mode. • **X.509 Certificate (client)** – Activates the X.509 authentication in client mode. • **X.509 Certificate (server)** – Activates the X.509 authentication in server mode.

Table 1: OpenVPN Configuration (continued)

Item	Description
Security Mode	Choose the security mode, *tls-auth* or *tls-crypt*. We recommend to use the *tls-crypt* mode for the security reasons. In this mode, all the data is encrypted with a pre-shared key. Moreover, this mode is more robust against the TLS denial of service attacks.
Pre-shared Secret	Specifies the pre-shared secret which you can use for every authentication mode.
CA Certificate	Specifies the CA Certificate which you can use for the username/password and X.509 Certificate authentication modes.
DH Parameters	Specifies the protocol for the DH parameters key exchange which you can use for X.509 Certificate authentication in the server mode.
Local Certificate	Specifies the certificate used in the local device. You can use this authentication certificate for the X.509 Certificate authentication mode.
Local Private Key	Specifies the key used in the local device. You can use the key for the X.509 Certificate authentication mode.
Local Passphrase	Passphrase used during private key generation.
Username	Specifies a login name which you can use for authentication in the username/-password mode.
Password	Specifies a password which you can use for authentication in the username/-password mode. Enter valid characters only, see chap. 6!
User's Up Script[1]	Custom script, executed when the OpenVPN tunnel is established.
User's Down Script[1]	Custom script, executed when the OpenVPN tunnel is closed.
Extra Options	Specifies additional parameters for the OpenVPN tunnel, such as DHCP options. The parameters are preceded by two dashes. For possible parameters see the help text in the router using SSH – run the `openvpnd ---help` command.

Table 1: OpenVPN Configuration (continued)

The changes in settings will be applied after pressing the *Apply* button.

Info

Tips for working with the configuration form:

- CLIENT routers must have filled in *Remote IP Address* item.
- For SERVER routers we recommend not filling in *Remote IP Address* item!
- If two routers are situated against each other, one of them is CLIENT and the other is SERVER.
- **It is always recommended to set *Ping Interval* and *Ping Timeout* items.**

[1] Parameters passed to the script are `cmd tun_dev tun_mtu link_mtu ifconfig_local_ip ifconfig_remo-te_ip [ init | restart` see Reference manual for OpenVPN, option `--up cmd`.

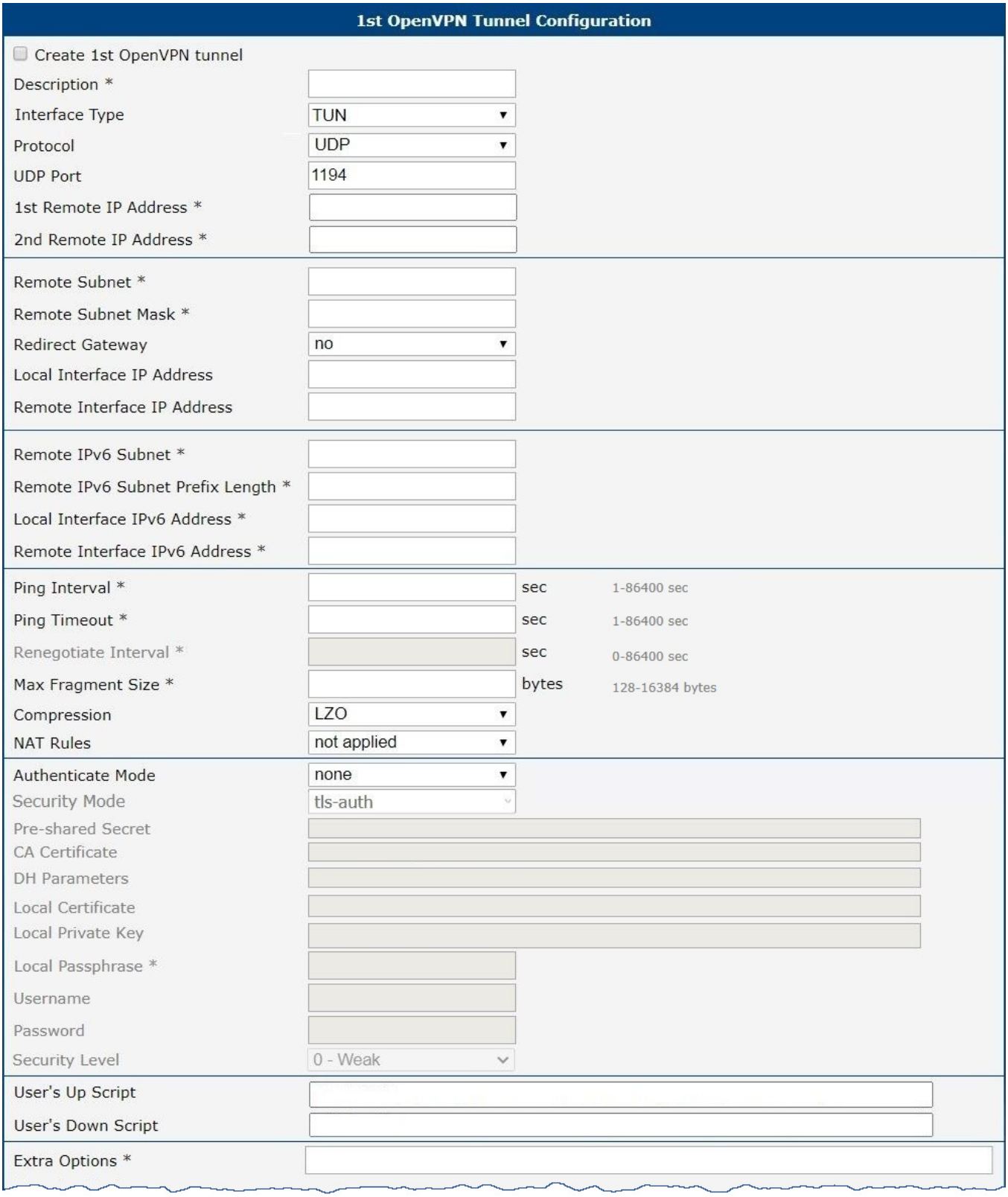

Figure 2: Configuration form for OpenVPN tunnel

3. Router on both sides of tunnel

The figure below shows a situation where the Advantech router is situated on both sides of OpenVPN tunnel. IP address of SIM cards in the router can be static or dynamic.

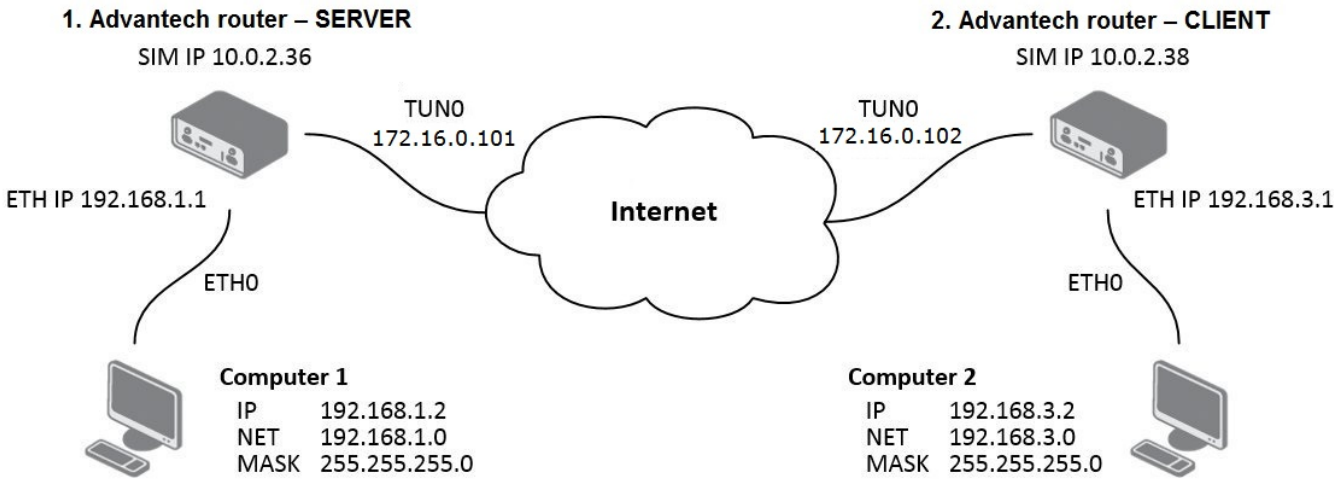

Figure 3: Router on both sides of tunnel

3.1 OpenVPN tunnel without authentication

Configuration of the first router – SERVER:

Item	Value
Remote Subnet	192.168.3.0
Remote Subnet Mask	255.255.255.0
Local Interface IP Address	172.16.0.101
Remote Interface IP Address	172.16.0.102

Table 2: Configuration of the first router (no authentication)

Configuration of the second router – CLIENT:

Item	Value
Remote IP Address	10.0.2.36
Remote Subnet	192.168.1.0
Remote Subnet Mask	255.255.255.0
Local Interface IP Address	172.16.0.102
Remote Interface IP Address	172.16.0.101

Table 3: Configuration of the second router (no authentication)

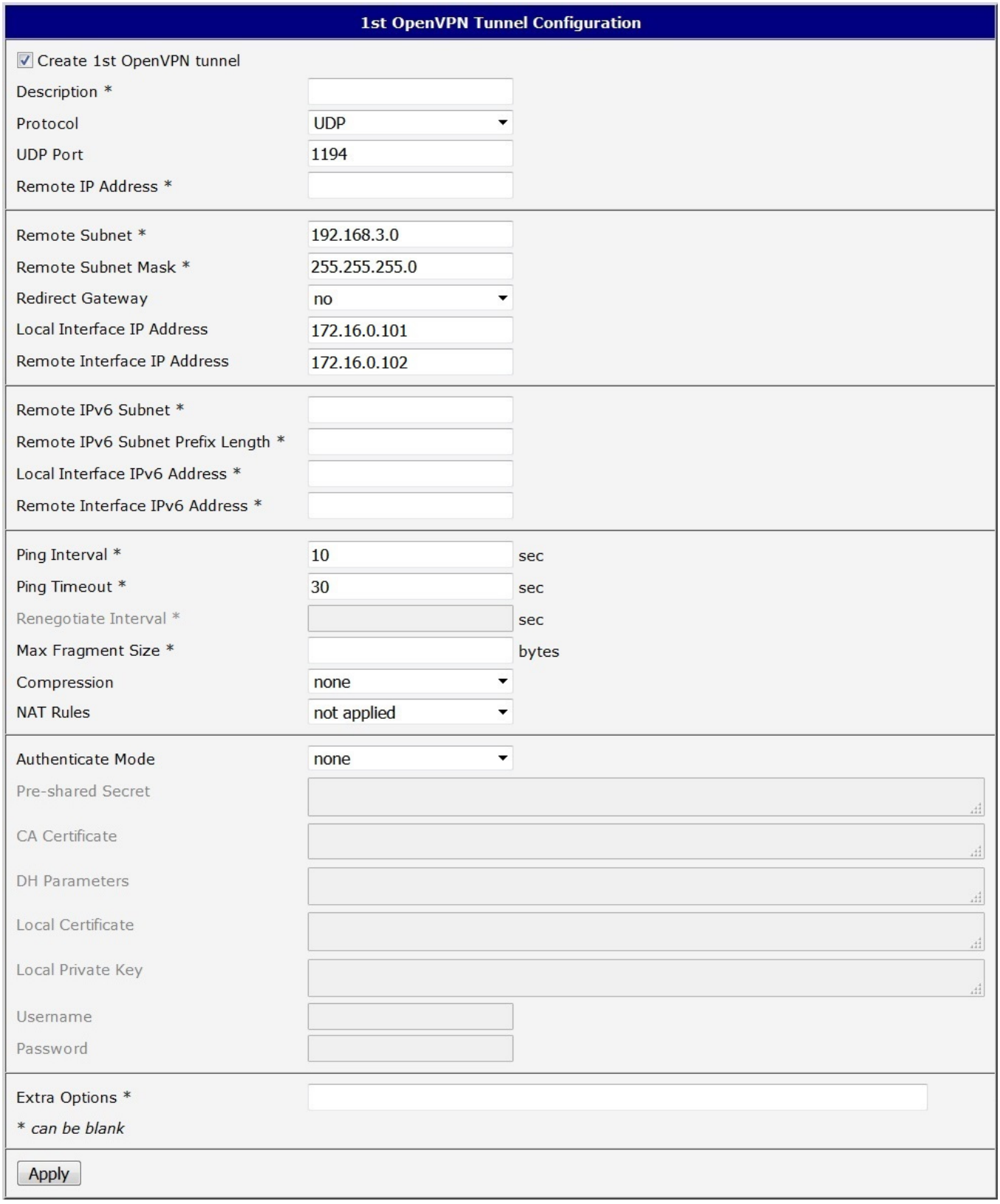

Figure 4: Configuration of the first router – SERVER (no authentication)

Note: Configuration of the second router is similar, the difference is only in items listed in table 3 *Open-VPN tunnel without authentication* on page 7. If *NAT Rules* parameter is enabled, specified rules (in the configuration form of NAT) are applied to the OpenVPN tunnel.

After establishing an OpenVPN tunnel, an interface tun0 and a route in the routing table of the router are displayed on the *Network Status* page.

Network Status

Interfaces

```
eth0      Link encap:Ethernet  HWaddr 00:55:44:33:52:98
          inet addr:192.168.2.234  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6743 errors:0 dropped:382 overruns:0 frame:0
          TX packets:532 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:541103 (528.4 KB)  TX bytes:277877 (271.3 KB)
          Interrupt:23

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:172.16.0.102  P-t-P:172.16.0.101  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Route Table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.2.27	0.0.0.0	UG	0	0	0	eth0
10.0.1.17	172.16.0.101	255.255.255.255	UGH	0	0	0	tun0
172.16.0.0	172.16.0.101	255.255.0.0	UG	0	0	0	tun0
172.16.0.1	172.16.0.101	255.255.255.255	UGH	0	0	0	tun0
172.16.0.101	0.0.0.0	255.255.255.255	UH	0	0	0	tun0
192.168.2.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
192.168.2.27	0.0.0.0	255.255.255.255	UH	0	0	0	eth0

Figure 5: Network Status

It is also possible to check successful establishment of OpenVPN tunnel in the system log (*System Log* item in menu). Listings should end with line *Initialization Sequence Completed*.

System Log

System Messages

```
2013-05-10 18:27:52 openvpn[1338]: Attempting to establish TCP connection with 88.86.101.201:1194 [nonblock]
2013-05-10 18:27:55 openvpn[1338]: TCP connection established with 88.86.101.201:1194
2013-05-10 18:27:55 openvpn[1338]: TCPv4_CLIENT link local: [undef]
2013-05-10 18:27:55 openvpn[1338]: TCPv4_CLIENT link remote: 88.86.101.201:1194
2013-05-10 18:27:58 openvpn[1338]: WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2013-05-10 18:28:00 openvpn[1338]: [LT_server] Peer Connection Initiated with 88.86.101.201:1194
2013-05-10 18:28:14 openvpn[1338]: TUN/TAP device tap0 opened
2013-05-10 18:28:14 openvpn[1338]: /sbin/ifconfig tap0 5.11.2.2 netmask 255.255.0.0 mtu 1500 broadcast 5.11.255.255
2013-05-10 18:28:14 openvpn[1338]: Initialization Sequence Completed
```

Save Log | Save Report

Figure 6: System log

3.2 OpenVPN tunnel with pre-shared secret authentication

Configuration of the first router – SERVER:

Item	Value
Remote Subnet	192.168.3.0
Remote Subnet Mask	255.255.255.0
Local Interface IP Address	172.16.0.101
Remote Interface IP Address	172.16.0.102
Authenticate Mode	pre-shared secret
Pre-shared Secret	shared key for both of routers

Table 4: Configuration of the first router (pre-shared secret)

Configuration of the second router – CLIENT:

Item	Value
Remote IP Address	10.0.2.36
Remote Subnet	192.168.1.0
Remote Subnet Mask	255.255.255.0
Local Interface IP Address	172.16.0.102
Remote Interface IP Address	172.16.0.101
Authenticate Mode	pre-shared secret
Pre-shared Secret	shared key for both of routers

Table 5: Configuration of the second router (pre-shared secret)

Info

The procedure of creating pre-shared key is described in chapter 8 *Creation of pre-shared key in Windows* on page 34.

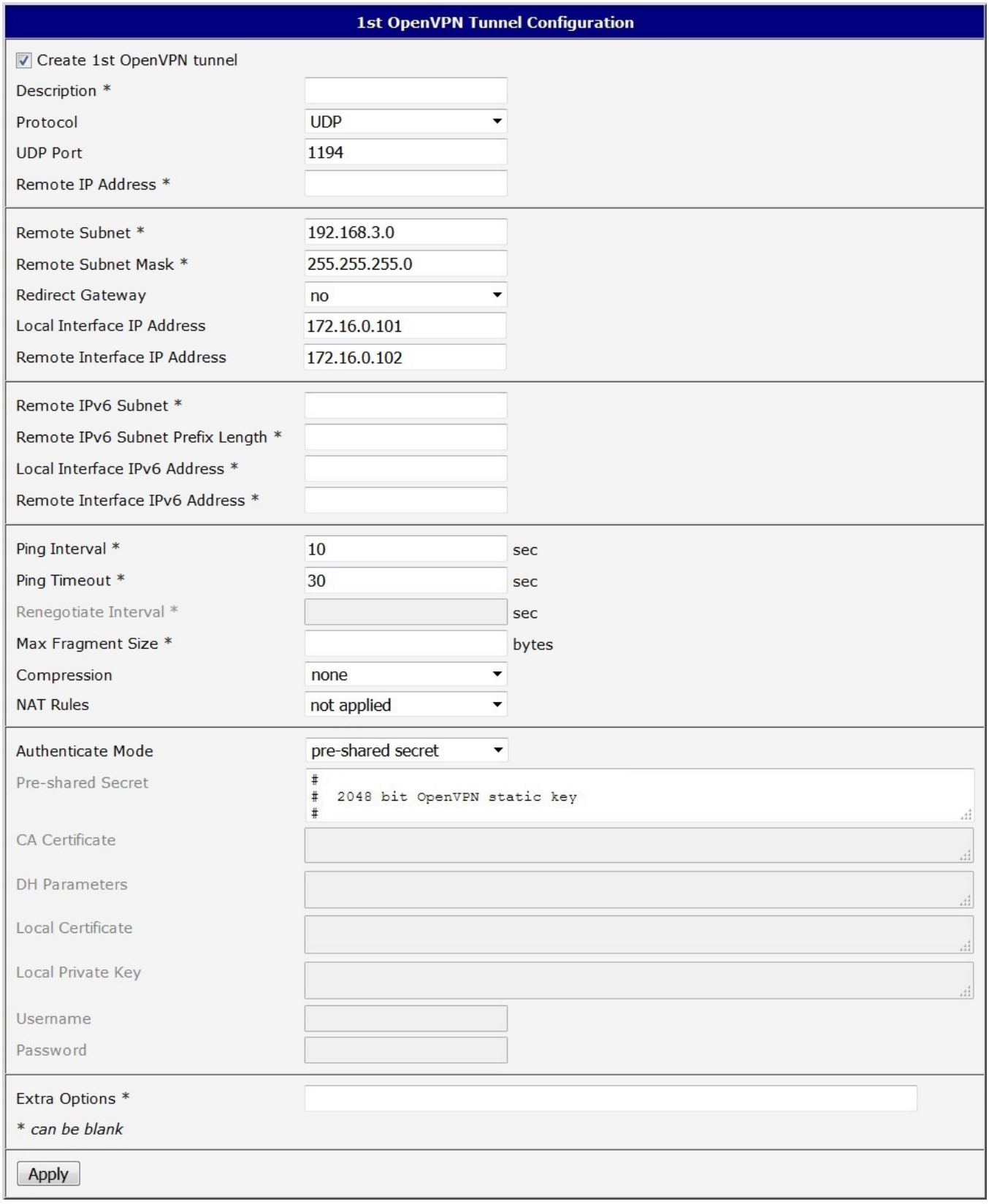

Figure 7: Configuration of the first router – SERVER (pre-shared secret)

Note: Configuration of the second router is similar, the difference is only in items listed in table 5 *OpenVPN tunnel with pre-shared secret authentication* on page 10. If *NAT Rules* parameter is enabled, specified rules (in the configuration form of NAT) are applied to the OpenVPN tunnel.

After establishing an OpenVPN tunnel, an interface tun0 and a route in the routing table of the router are displayed on the *Network Status* page.

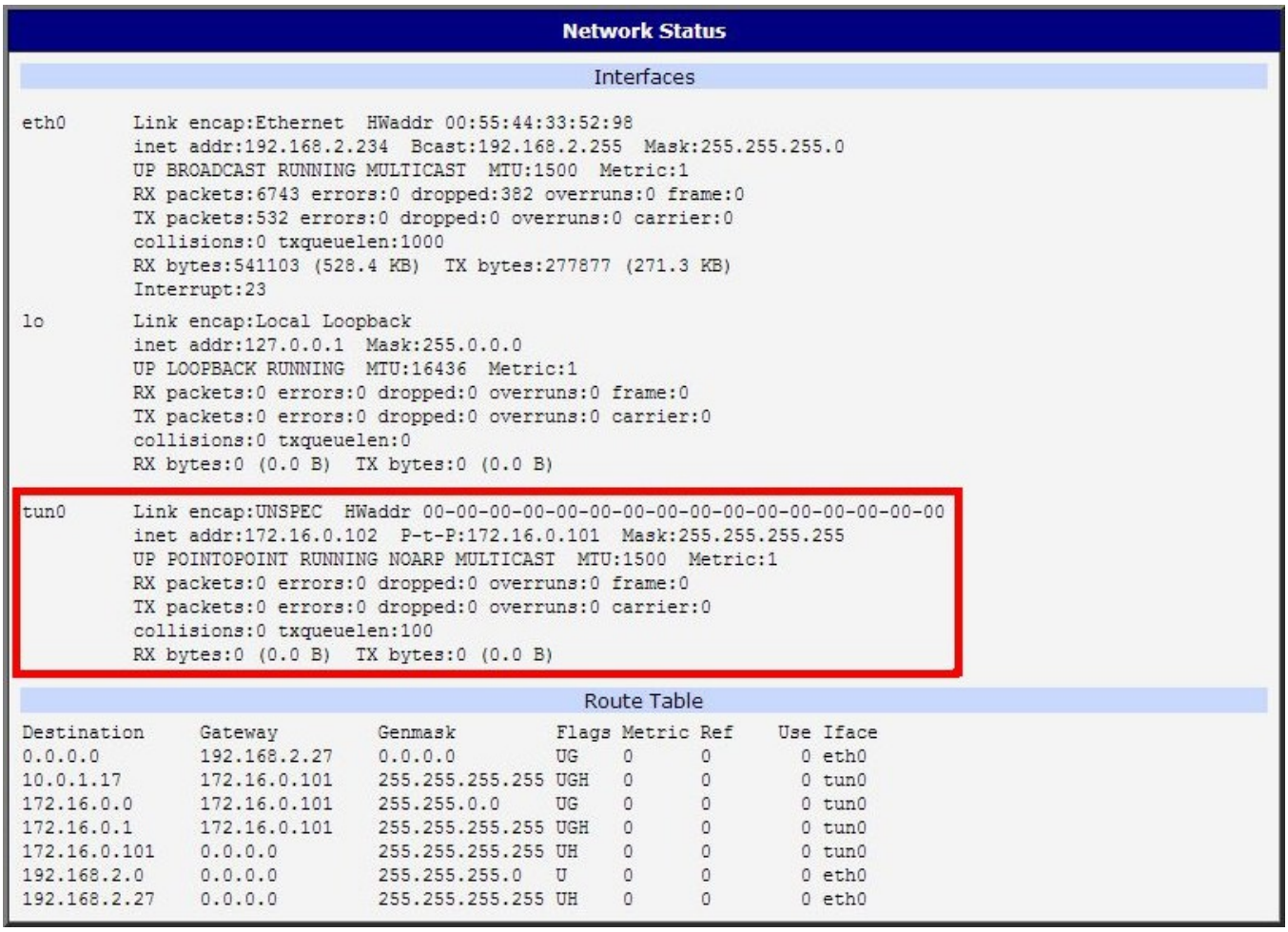

Figure 8: Network Status

It is also possible to check successful establishment of OpenVPN tunnel in the system log (*System Log* item in menu). Listings should end with line *Initialization Sequence Completed*.

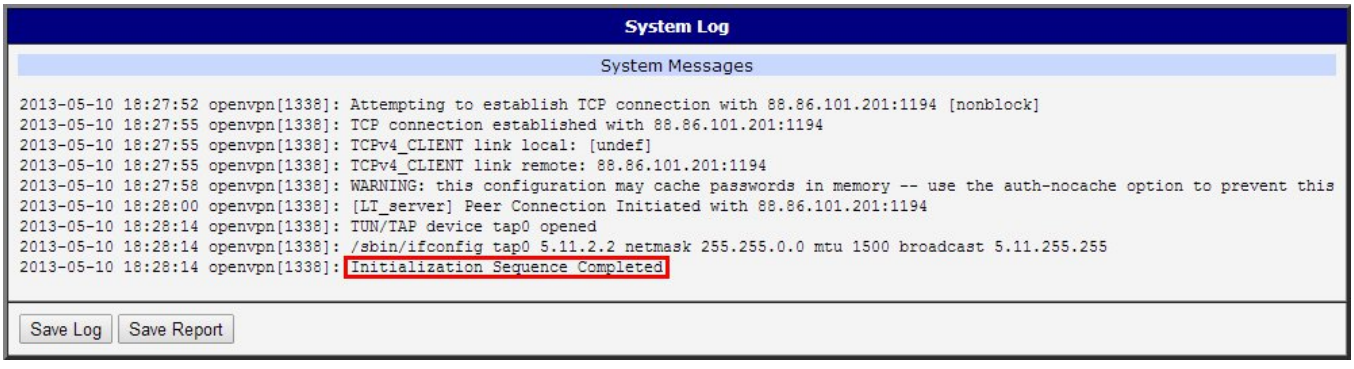

Figure 9: System log

3.3 OpenVPN tunnel with username/password authentication

The router can run **only as a client** when the username/password authentication is configured.
Configuration of the router – CLIENT only:

Item	Value
Remote IP Address	10.0.2.36
Remote Subnet	192.168.1.0
Remote Subnet Mask	255.255.255.0
Authenticate Mode	username/password
CA Certificate	generated certificate from VPN server
Username	username assigned by the VPN server
Password	password assigned by the VPN server

Table 6: Configuration of the router (username/password)

Info

The procedure of creating certificate is described in chapter 9 *Creation of certificates in Windows* on page 35.

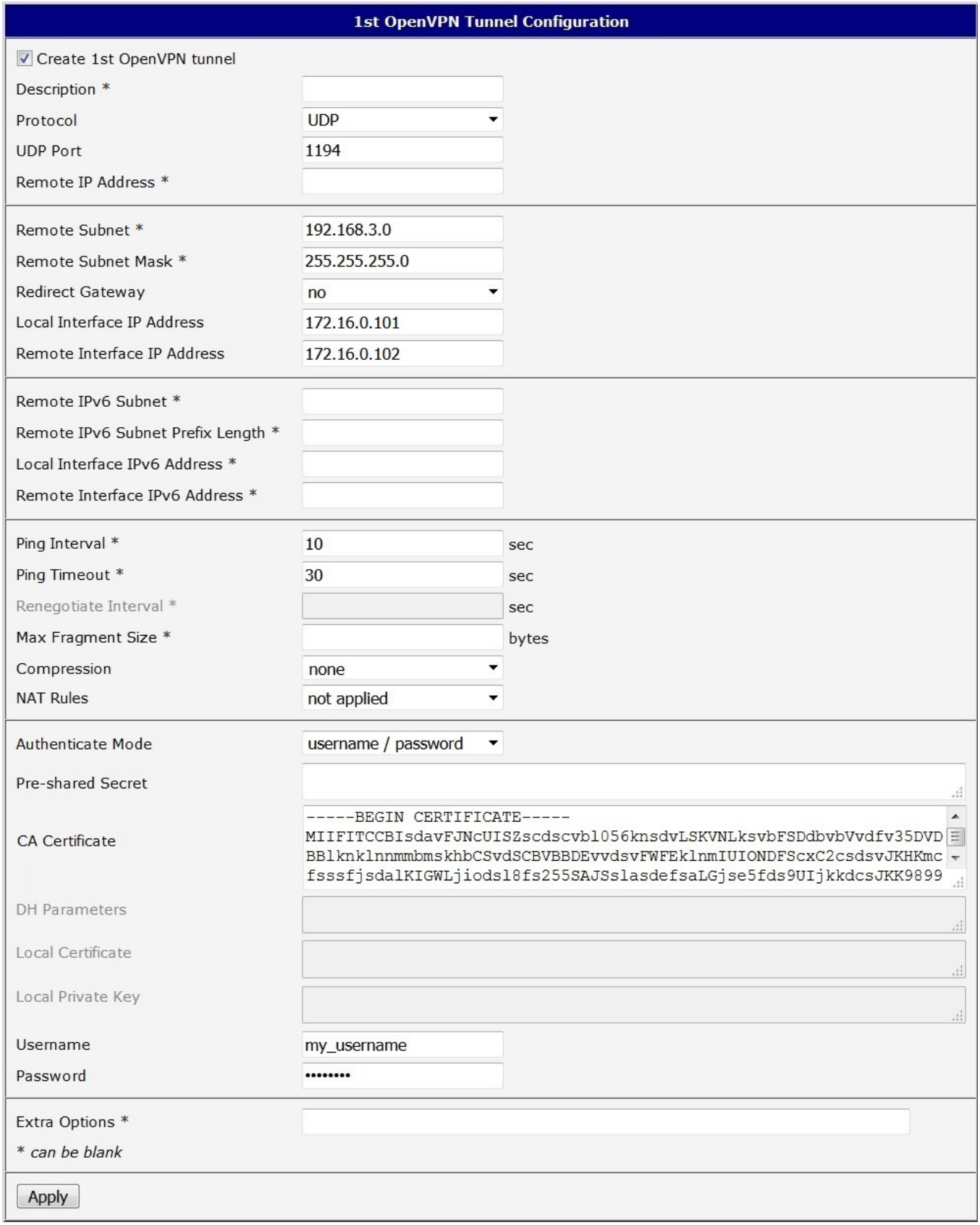

Figure 10: Configuration of the first router – SERVER (username/password)

Note: Configuration of the second router is similar, the difference is only in items listed in table 6 *OpenVPN tunnel with username/password authentication* on page 13. If *NAT Rules* parameter is enabled, specified rules (in the configuration form of NAT) are applied to the OpenVPN tunnel.

After establishing an OpenVPN tunnel, an interface tun0 and a route in the routing table of the router are displayed on the *Network Status* page.

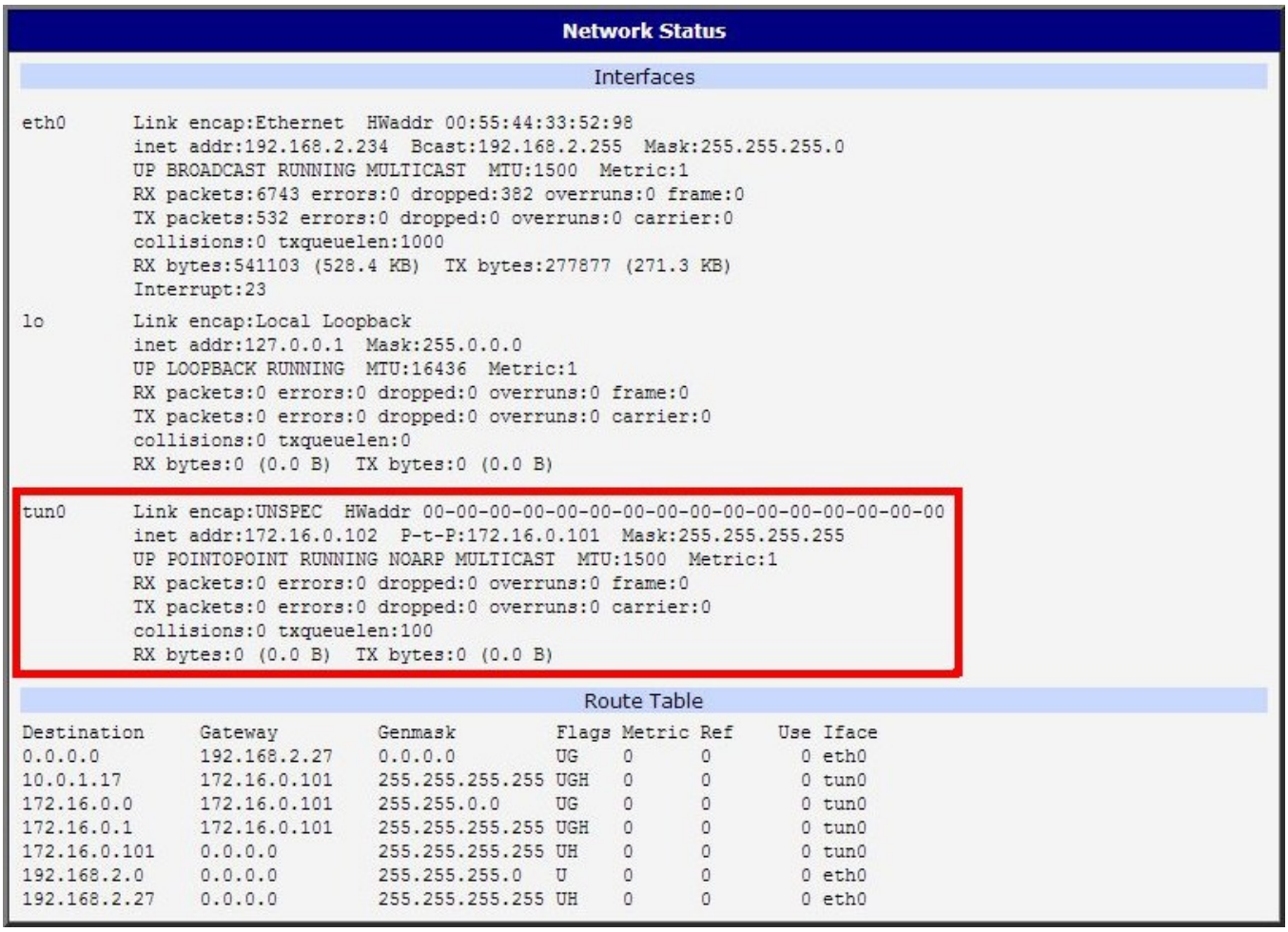

Figure 11: Network Status

It is also possible to check successful establishment of OpenVPN tunnel in the system log (*System Log* item in menu). Listings should end with line *Inicialization Sequence Completed*.

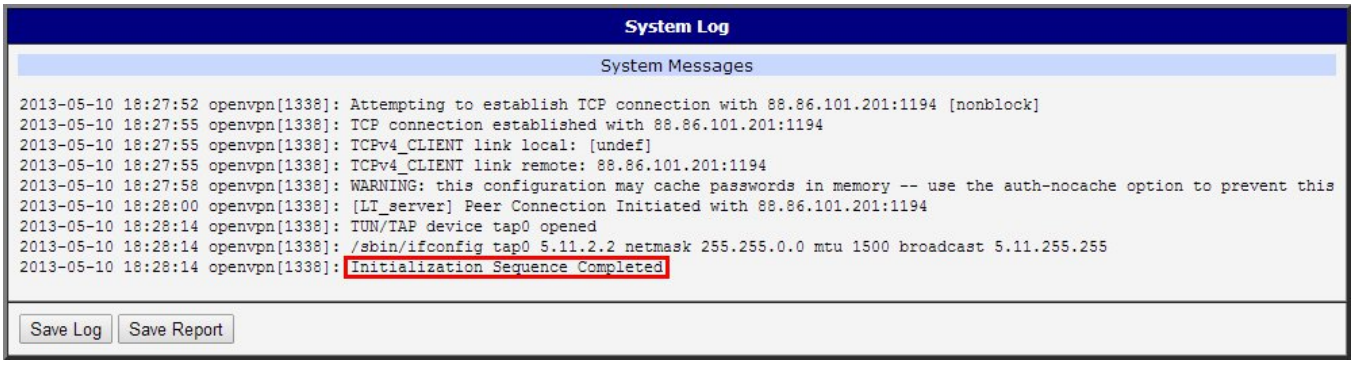

Figure 12: System log

3.4 OpenVPN tunnel with X.509 certificate authentication

Configuration of the first router – SERVER:

Item	Value
Remote Subnet	192.168.3.0
Remote Subnet Mask	255.255.255.0
Local Interface IP Address	172.16.0.101
Remote Interface IP Address	172.16.0.102
Authenticate Mode	X.509 certificate (server)
CA Certificate	generated certificate from VPN server
DH Parameters	Diffie-Hellman protocol for key exchange
Local Certificate	local certificate assigned by the VPN server
Local Private Key	local private key assigned by the VPN server

Table 7: Configuration of the first router (X.509 certificate)

Configuration of the second router – CLIENT:

Item	Value
Remote IP Address	10.0.2.36
Remote Subnet	192.168.1.0
Remote Subnet Mask	255.255.255.0
Local Interface IP Address	172.16.0.102
Remote Interface IP Address	172.16.0.101
Authenticate Mode	X.509 certificate (client)
CA Certificate	generated certificate from VPN server
Local Certificate	local certificate assigned by the VPN server
Local Private Key	local private key assigned by the VPN server

Table 8: Configuration of the second router (X.509 certificate)

Info

The procedure of creating certificate is described in chapter 9 *Creation of certificates in Windows* on page 35.

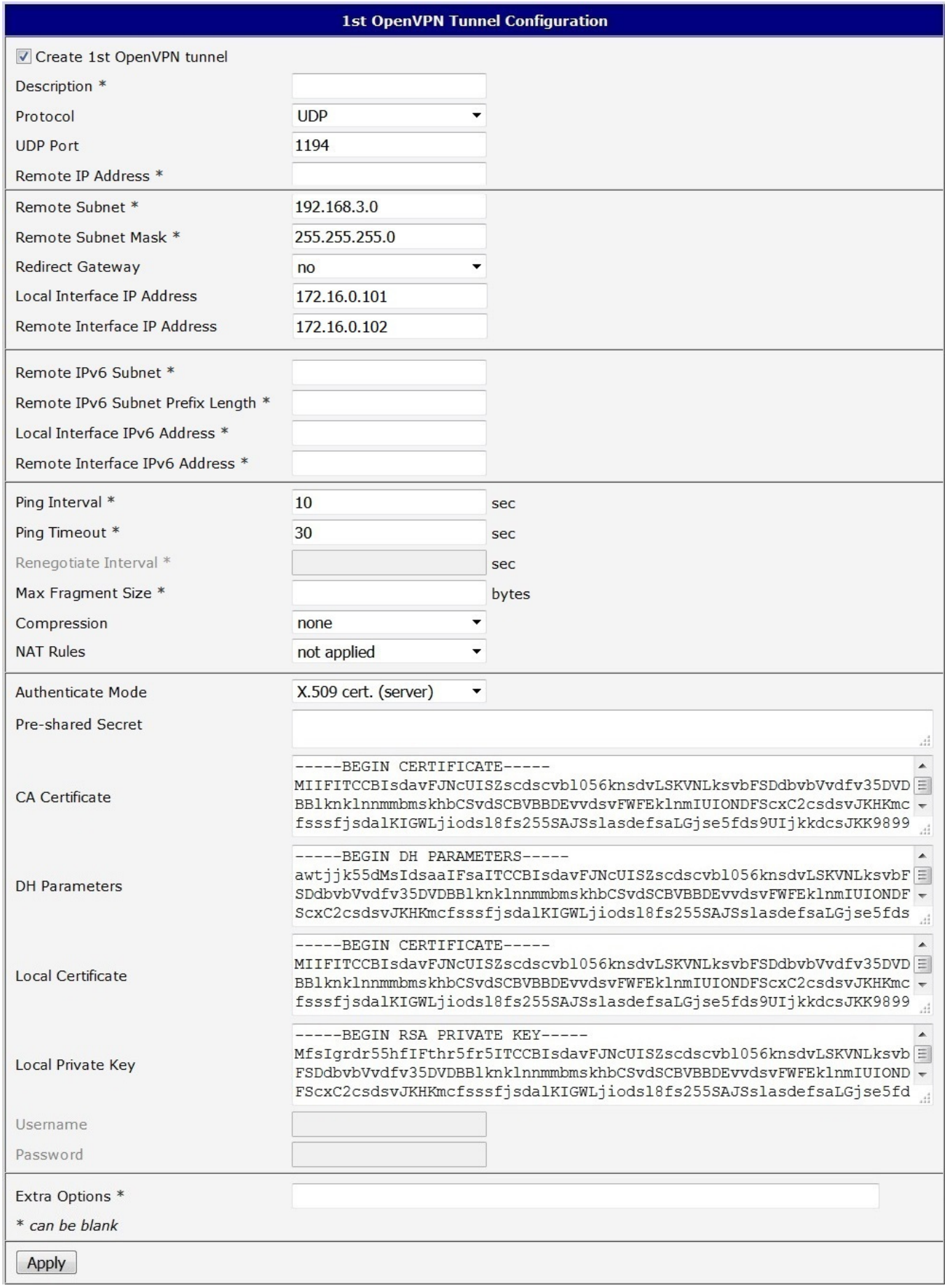

Figure 13: Configuration of the first router – SERVER (X.509 certificate)

Note: Configuration of the second router is similar, the difference is only in items listed in table 8 *OpenVPN tunnel with X.509 certificate authentication* on page 16. If *NAT Rules* parameter is enabled, specified rules (in the configuration form of NAT) are applied to the OpenVPN tunnel.

After establishing an OpenVPN tunnel, an interface tun0 and a route in the routing table of the router are displayed on the *Network Status* page.

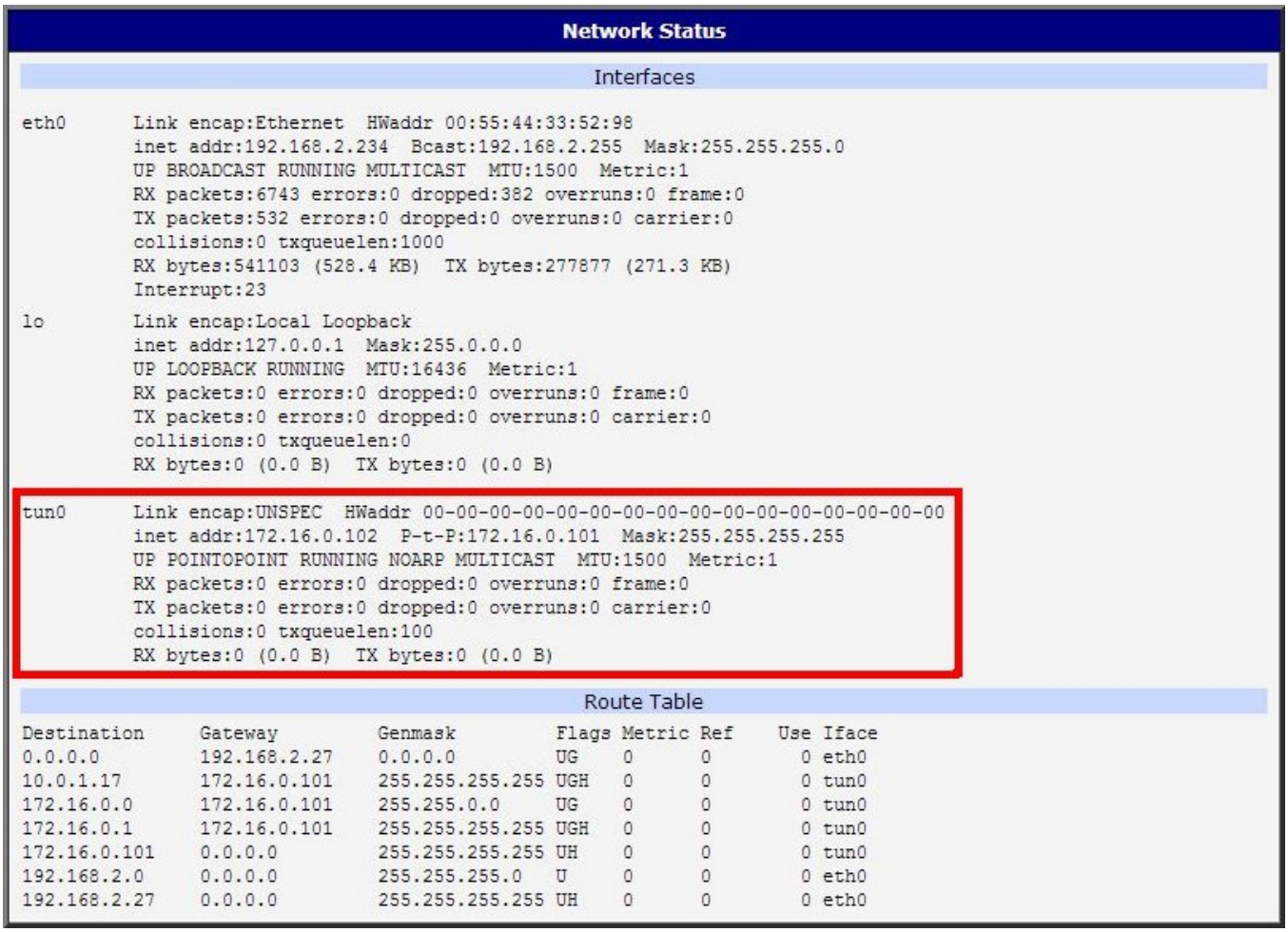

Figure 14: Network Status

It is also possible to check successful establishment of OpenVPN tunnel in the system log (*System Log* item in menu). Listings should end with line *Inicialization Sequence Completed*.

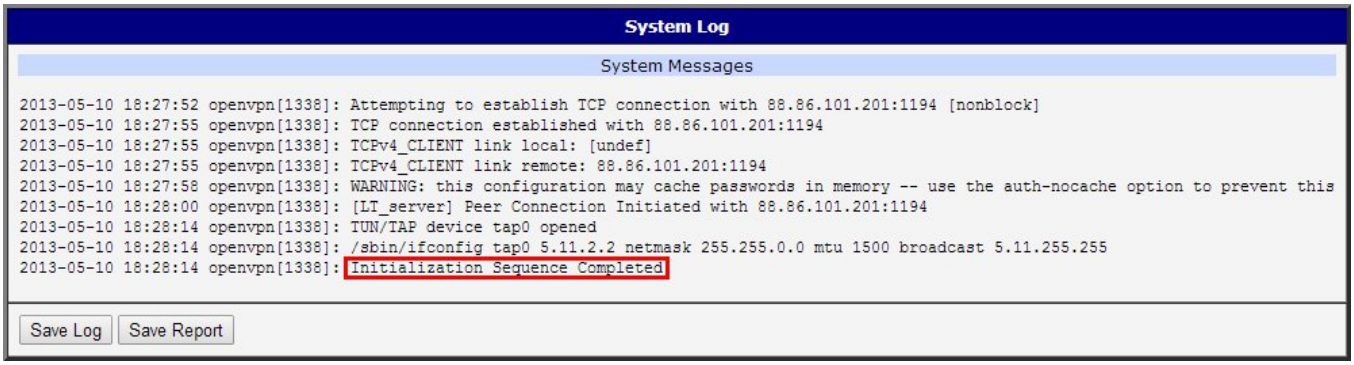

Figure 15: System log

4. Tunnel against WIN/Linux CLIENT

The figure below shows situation, where Advantech router is on one side of OpenVPN tunnel and device with an operating system Windows/Linux in CLIENT mode is on the other side.

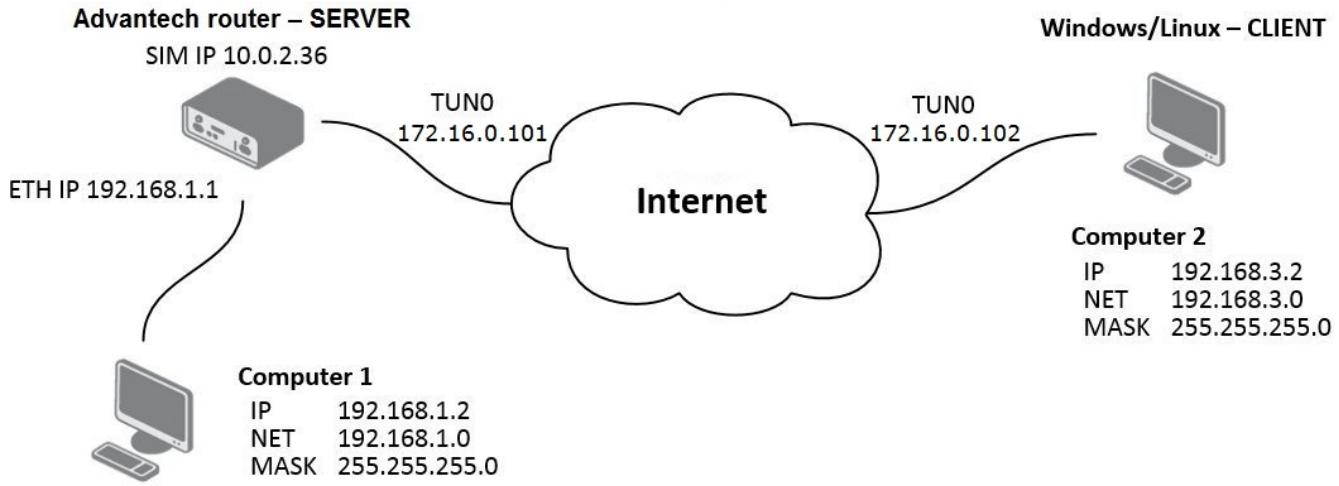

Figure 16: OpenVPN tunnel against Windows/Linux CLIENT

4.1 OpenVPN tunnel configuration on the router

Item	Value
Remote Subnet	192.168.3.0
Remote Subnet Mask	255.255.255.0
Local Interface IP Address	172.16.0.101
Remote Interface IP Address	172.16.0.102
Authenticate Mode	X.509 certificate (server)
CA Certificate	generated certificate from router (SERVER)
DH Parameters	Diffie-Hellman protocol for key exchange
Local Certificate	local certificate assigned by router (SERVER)
Local Private Key	local private key assigned by router (SERVER)

Table 9: Router configuration

1st OpenVPN Tunnel Configuration

☑ Create 1st OpenVPN tunnel	
Description *	
Protocol	UDP
UDP Port	1194
Remote IP Address *	
Remote Subnet *	192.168.3.0
Remote Subnet Mask *	255.255.255.0
Redirect Gateway	no
Local Interface IP Address	172.16.0.101
Remote Interface IP Address	172.16.0.102
Remote IPv6 Subnet *	
Remote IPv6 Subnet Prefix Length *	
Local Interface IPv6 Address *	
Remote Interface IPv6 Address *	
Ping Interval *	10 sec
Ping Timeout *	30 sec
Renegotiate Interval *	sec
Max Fragment Size *	bytes
Compression	none
NAT Rules	not applied
Authenticate Mode	X.509 cert. (server)
Pre-shared Secret	
CA Certificate	-----BEGIN CERTIFICATE----- MIIFITCCBIsdavFJNcUISZscdscvbl056knsdvLSKVNLksvbFSDdbvbVvdfv35DVD BBlknklnnmmbmskhbCSvdSCBVBBDEvvdsvFWFEklnmIUIONDFScxC2csdsvJKHKmc fsssfjsdalKIGWLjiodsl8fs255SAJSslasdefsaLGjse5fds9UIjkkdcsJKK9899
DH Parameters	-----BEGIN DH PARAMETERS----- awtjjk55dMsIdsaaIFsaITCCBIsdavFJNcUISZscdscvbl056knsdvLSKVNLksvbF SDdbvbVvdfv35DVDBBlknklnnmmbmskhbCSvdSCBVBBDEvvdsvFWFEklnmIUIONDF ScxC2csdsvJKHKmcfsssfjsdalKIGWLjiodsl8fs255SAJSslasdefsaLGjse5fds
Local Certificate	-----BEGIN CERTIFICATE----- MIIFITCCBIsdavFJNcUISZscdscvbl056knsdvLSKVNLksvbFSDdbvbVvdfv35DVD BBlknklnnmmbmskhbCSvdSCBVBBDEvvdsvFWFEklnmIUIONDFScxC2csdsvJKHKmc fsssfjsdalKIGWLjiodsl8fs255SAJSslasdefsaLGjse5fds9UIjkkdcsJKK9899
Local Private Key	-----BEGIN RSA PRIVATE KEY----- MfsIgrdr55hfIFthr5fr5ITCCBIsdavFJNcUISZscdscvbl056knsdvLSKVNLksvb FSDdbvbVvdfv35DVDBBlknklnnmmbmskhbCSvdSCBVBBDEvvdsvFWFEklnmIUIOND FScxC2csdsvJKHKmcfsssfjsdalKIGWLjiodsl8fs255SAJSslasdefsaLGjse5fd
Username	
Password	
Extra Options *	

* *can be blank*

Apply

Figure 17: Router configuration

Note: If *NAT Rules* parameter is enabled, specified rules (in the configuration form of NAT) are applied to the OpenVPN tunnel.

After establishing an OpenVPN tunnel, an interface tun0 and a route in the routing table of the router are displayed on the *Network Status* page.

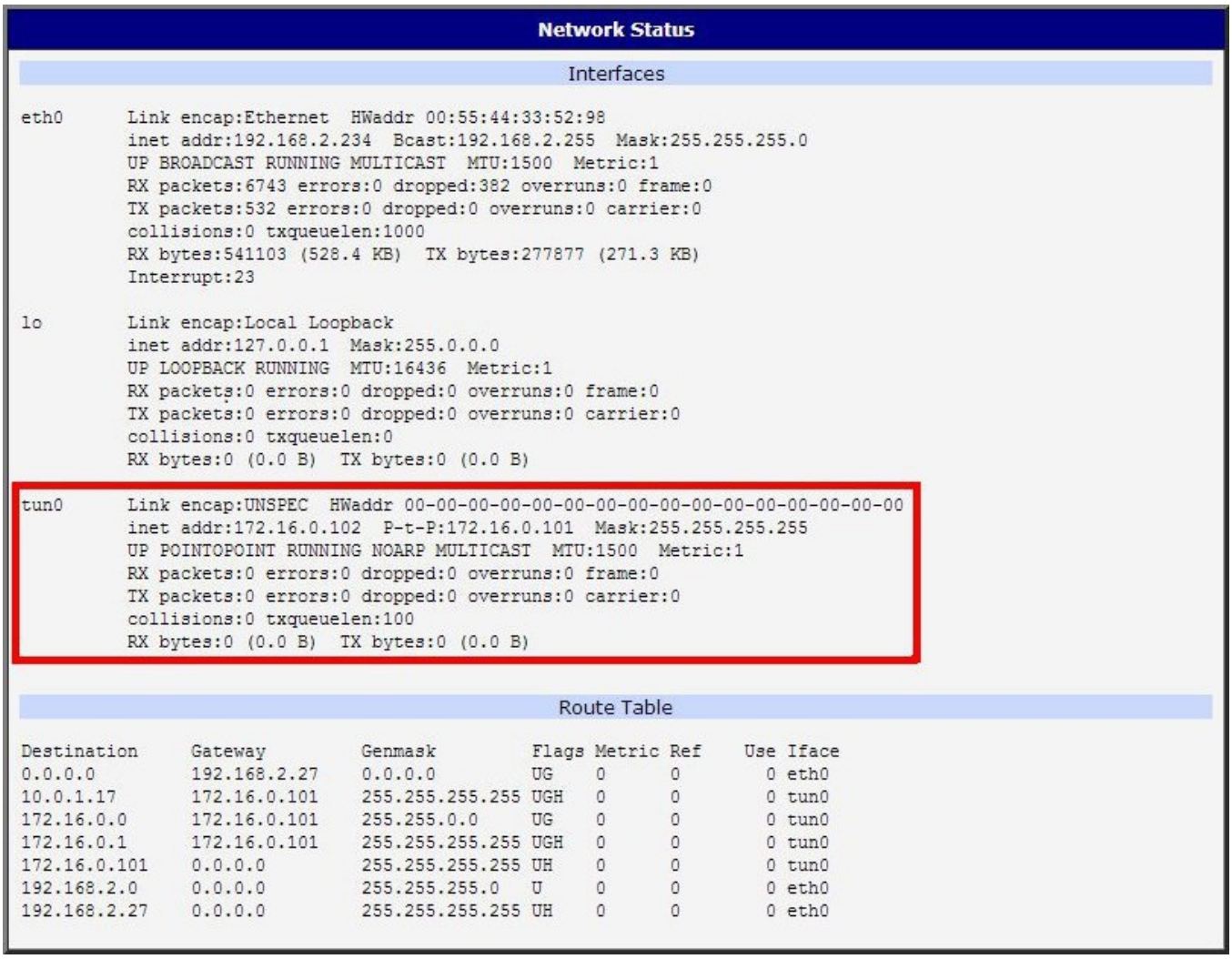

Figure 18: Network Status

It is also possible to check successful establishment of OpenVPN tunnel in the system log (*System Log* item in menu). Listings should end with line *Inicialization Sequence Completed*.

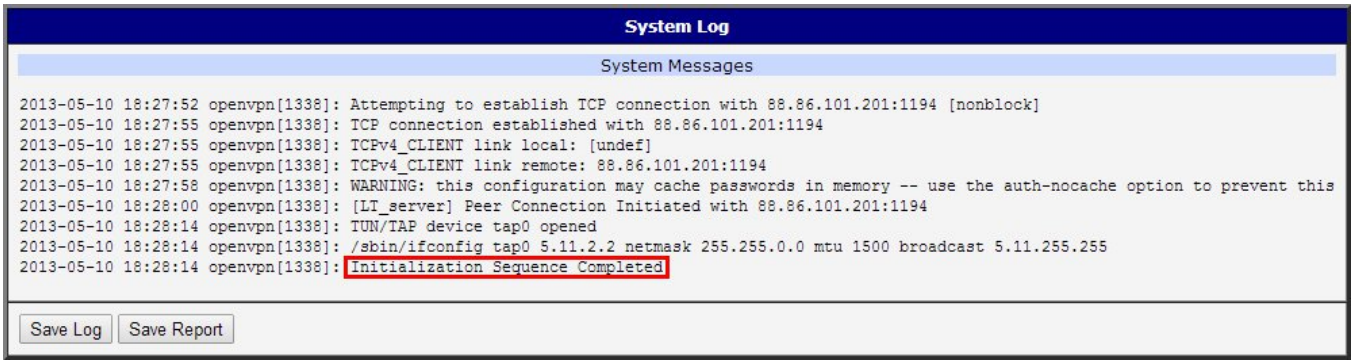

Figure 19: System log

4.2 OpenVPN tunnel configuration on WIN/Linux client

It is necessary to perform the following configuration on the computer, which is referred to as *Computer 1* in the diagram from the beginning of this chapter.

conf Code

```
remote 10.0.2.36
tls-client

dev tun

pull

ifconfig 172.16.0.102 172.16.0.101
route 192.168.1.0 255.255.255.0 172.16.0.102

mute 10

ca cacert.pem
cert client-cert.pem
key  client-key2.pem

verb 3
```

5. Tunnel against WIN/Linux SERVER

The figure below shows situation, where Advantech router is on one side of OpenVPN tunnel and device with an operating system Windows/Linux in SERVER mode is on the other side.

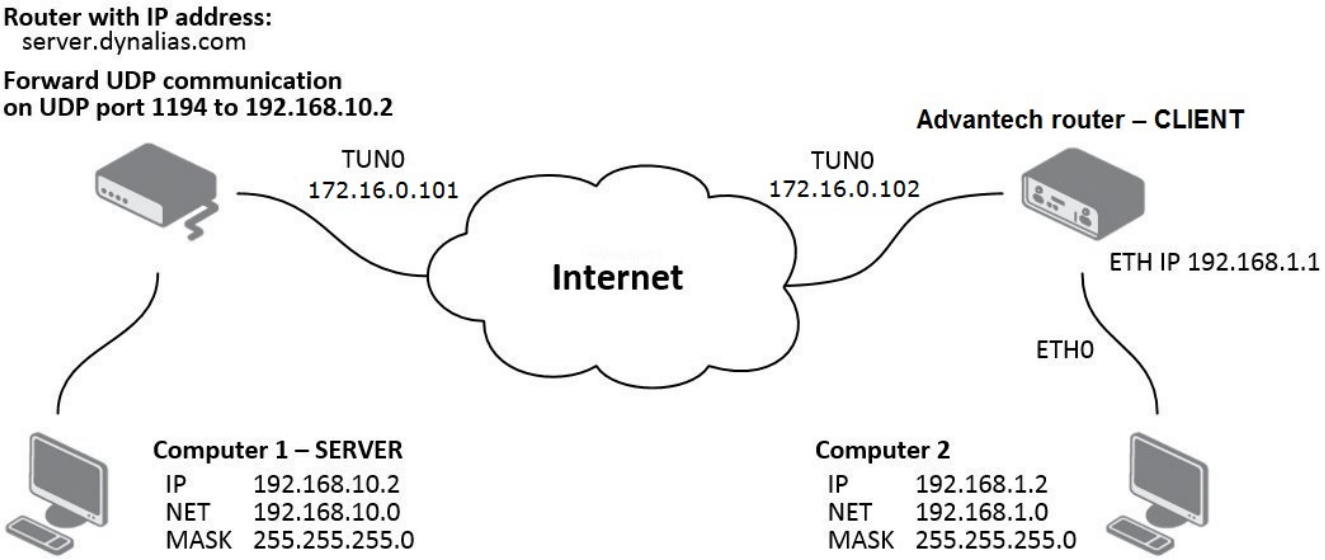

Figure 20: OpenVPN tunnel against Windows/Linux SERVER

5.1 OpenVPN tunnel configuration on the router

Item	Value
Remote IP Address	server.dynalias.com
Remote Subnet	192.168.10.0
Remote Subnet Mask	255.255.255.0
Local Interface IP Address	172.16.0.102
Remote Interface IP Address	172.16.0.101
Authenticate Mode	X.509 certificate (client)
CA Certificate	generated certificate from router
DH Parameters	Diffie-Hellman protocol for key exchange
Local Certificate	local certificate assigned by router
Local Private Key	local private key assigned by router

Table 10: Router configuration

Figure 21: Router configuration

Note: If *NAT Rules* parameter is enabled, specified rules (in the configuration form of NAT) are applied to the OpenVPN tunnel.

After establishing an OpenVPN tunnel, an interface tun0 and a route in the routing table of the router are displayed on the *Network Status* page.

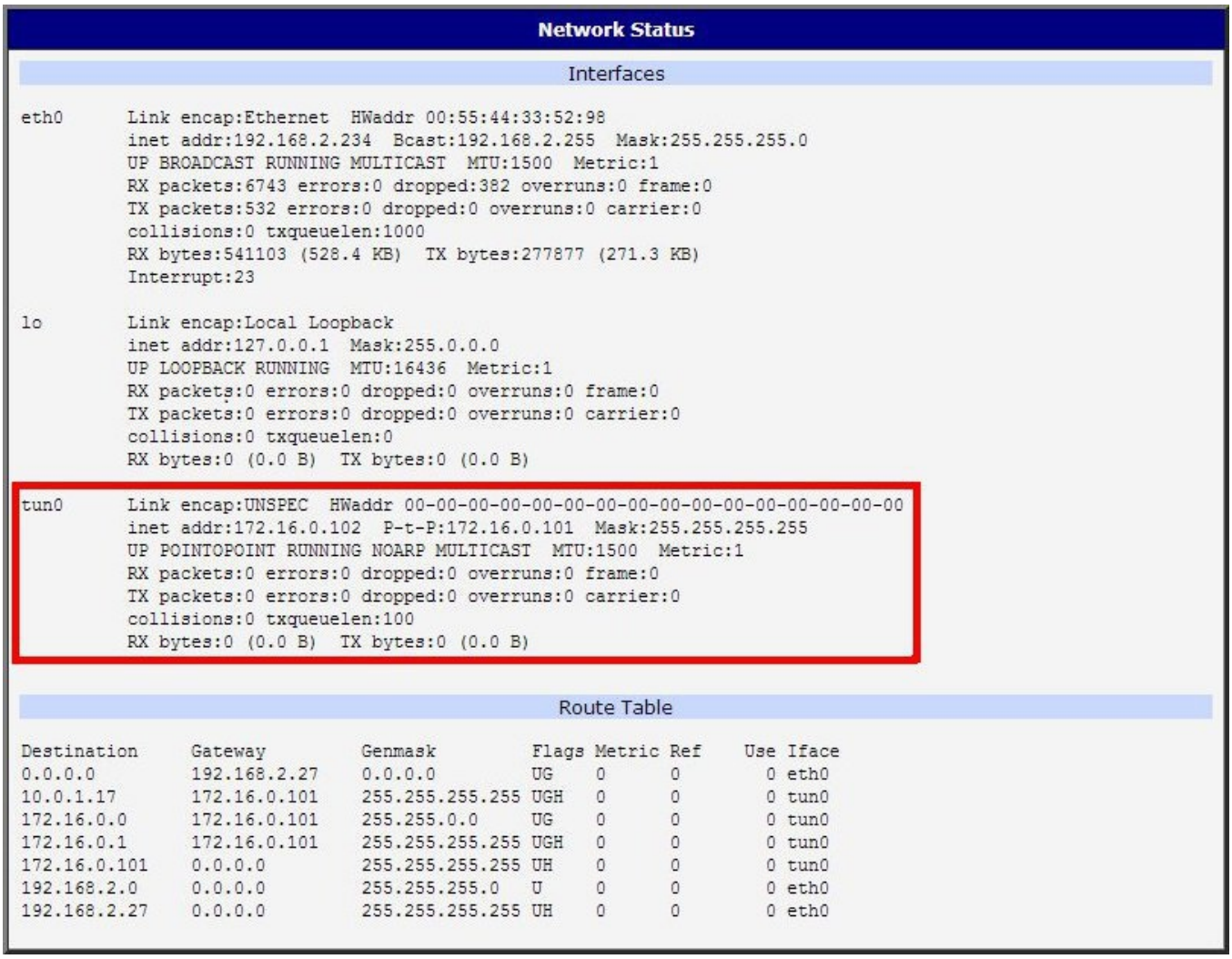

Figure 22: Network Status

It is also possible to check successful establishment of OpenVPN tunnel in the system log (*System Log* item in menu). Listings should end with line *Inicialization Sequence Completed*.

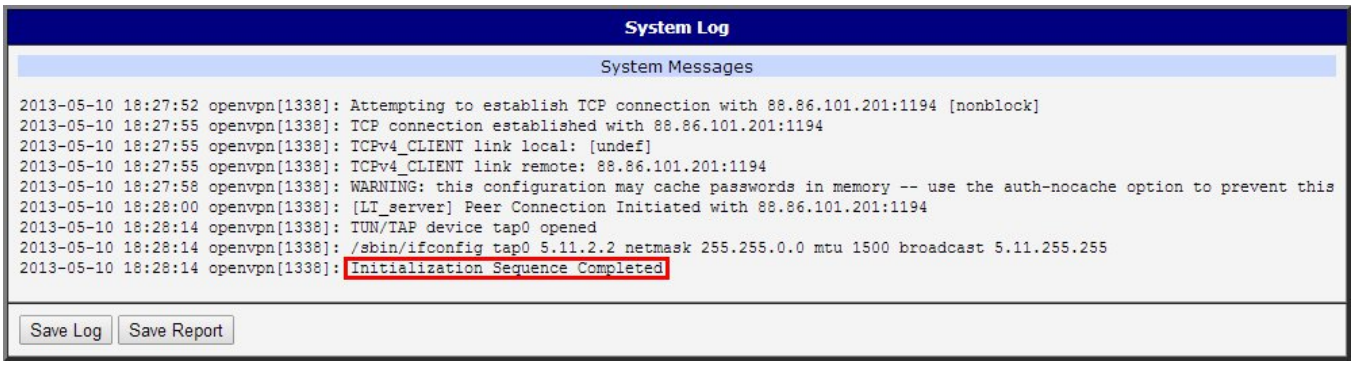

Figure 23: System log

5.2 Tunnel configuration on Computer 1 – Server

It is necessary to perform the following configuration on the computer, which is referred to as *Computer 1 – Server* in the diagram from the beginning of this chapter.

conf Code

```
local 192.168.10.2
tls-server

dev tun

pull

ifconfig 172.16.0.101 172.16.0.102
route 192.168.1.0 255.255.255.0 172.16.0.102

mute 10

ca cacert.pem
cert client-cert.pem
key client-key2.pem

verb 3
```

6. Multiclient-Server – Router (CLIENT)

The figure below shows situation, where OpenVPN multiserver is on one side of OpenVPN tunnel and several Advantech routers (three in this case) in CLIENT mode are on the other side.

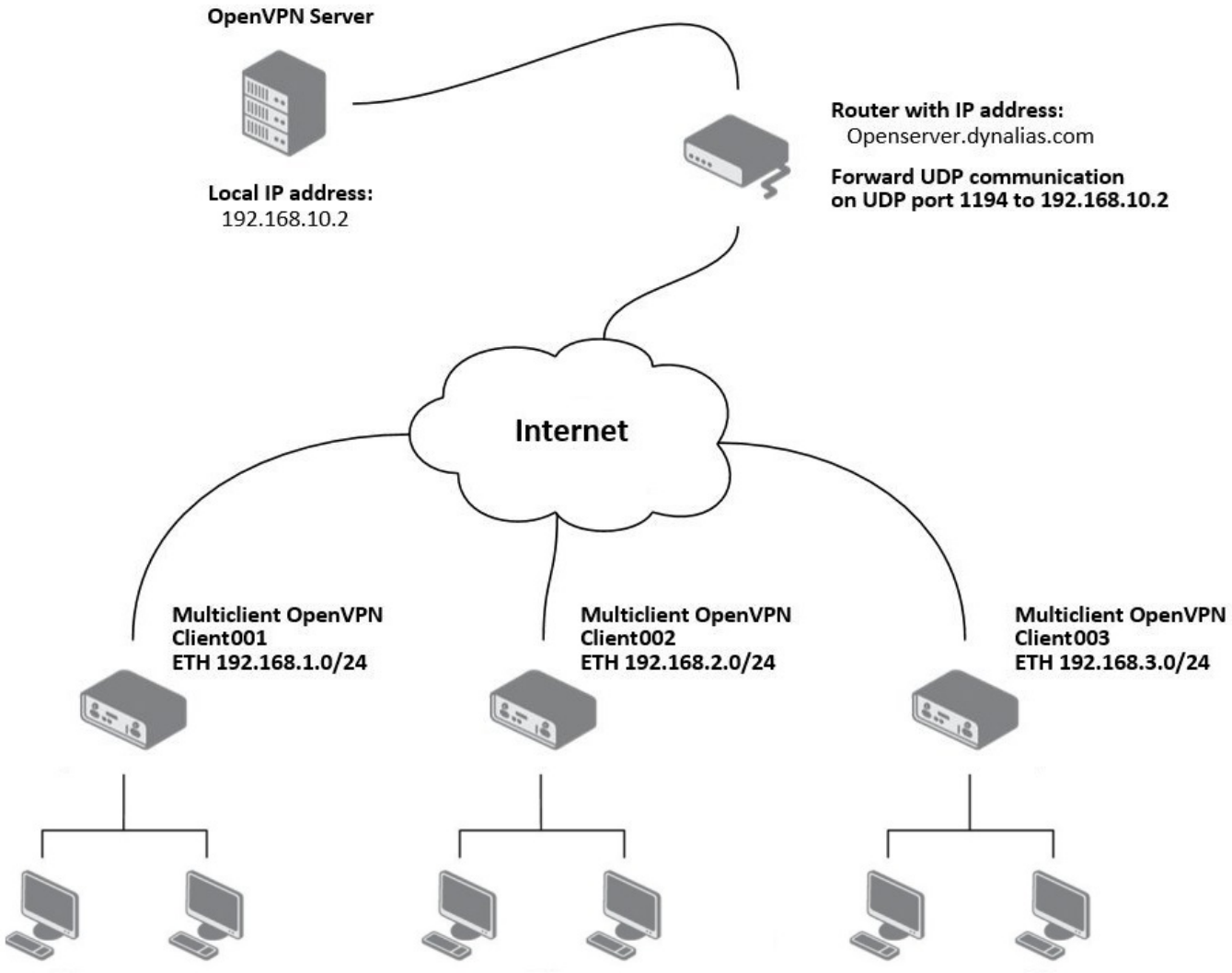

Figure 24: OpenVPN Multiserver – Advantech router (CLIENT)

6.1 OpenVPN tunnel configuration on Advantech routers

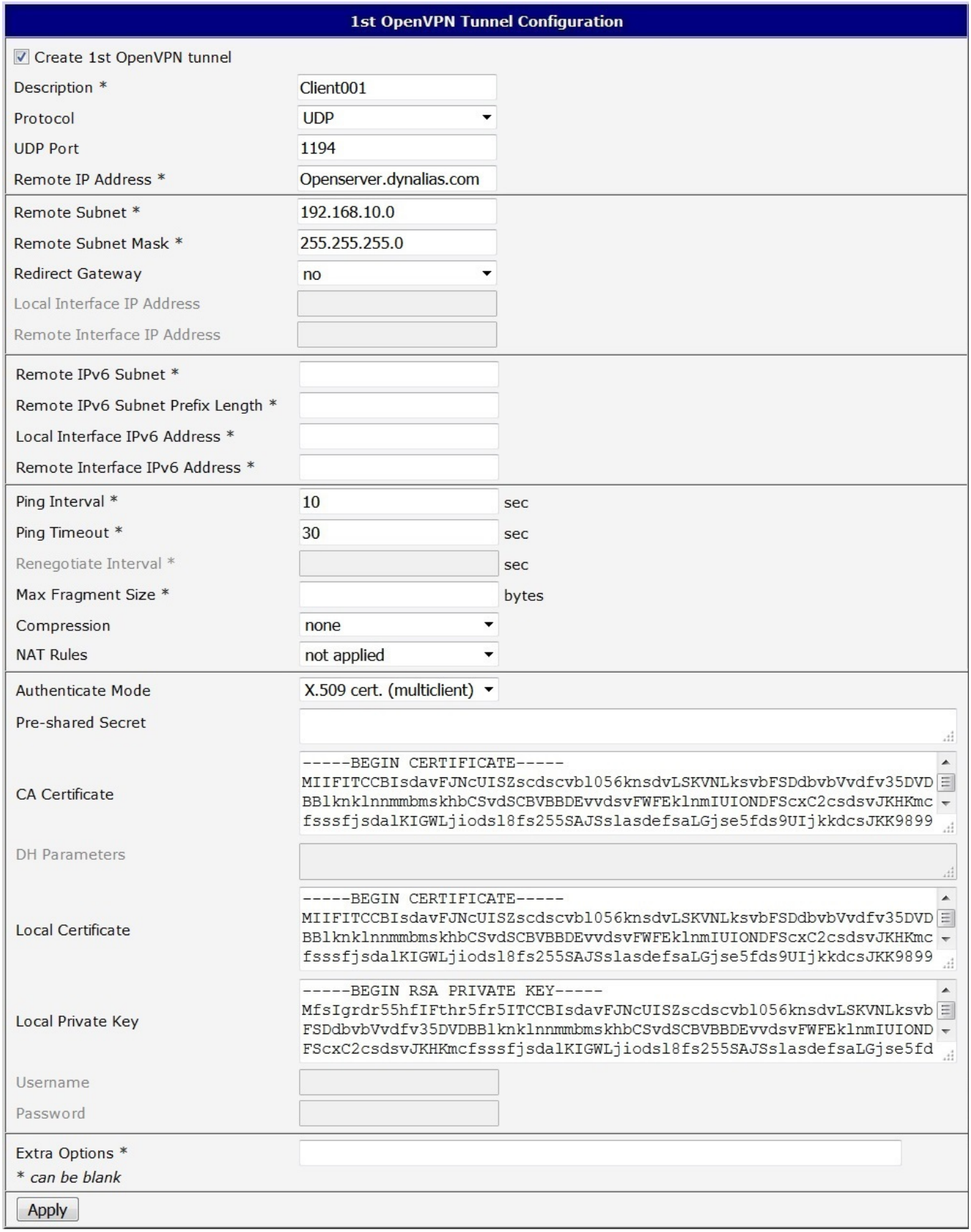

Figure 25: Configuration of Advantech router

Note: Configuration of other routers is similar, the difference is only in item *Description*.

6.2 OpenVPN server configuration

The configuration file (*.ovpn) stored on the server will consist of:

conf Code

```
server 10.8.0.0 255.255.255.0
port 1194
proto udp
dev tun
comp-lzo
keepalive 10 60
dh dh1024.pem
ca ca.crt
key server.key
cert server.crt
ifconfig-pool-persist ipp.txt
status openvpn-status.log
client-config-dir ccd
persist-key
persist-tun
verb 3
route 192.168.1.0 255.255.255.0
route 192.168.2.0 255.255.255.0
route 192.168.3.0 255.255.255.0
```

In the configuration above is specified configuration directory named as *ccd*. This directory is stored on the server in root directory of *OpenVPN* application. File names of client's configuration files stored at this directory must match the names of the certificates generated for every single client. In our case, there will be three configuration files with following content:

file *ccd\Client001* `iroute 192.168.1.0 255.255.255.0`
file *ccd\Client002* `iroute 192.168.2.0 255.255.255.0`
file *ccd\Client003* `iroute 192.168.3.0 255.255.255.0`

7. Multiclient-Server – Router (CL. to CL.)

The figure below shows situation, where OpenVPN server is on one side of OpenVPN tunnel and several Advantech routers (three in this case) in CLIENT mode are on the other side.

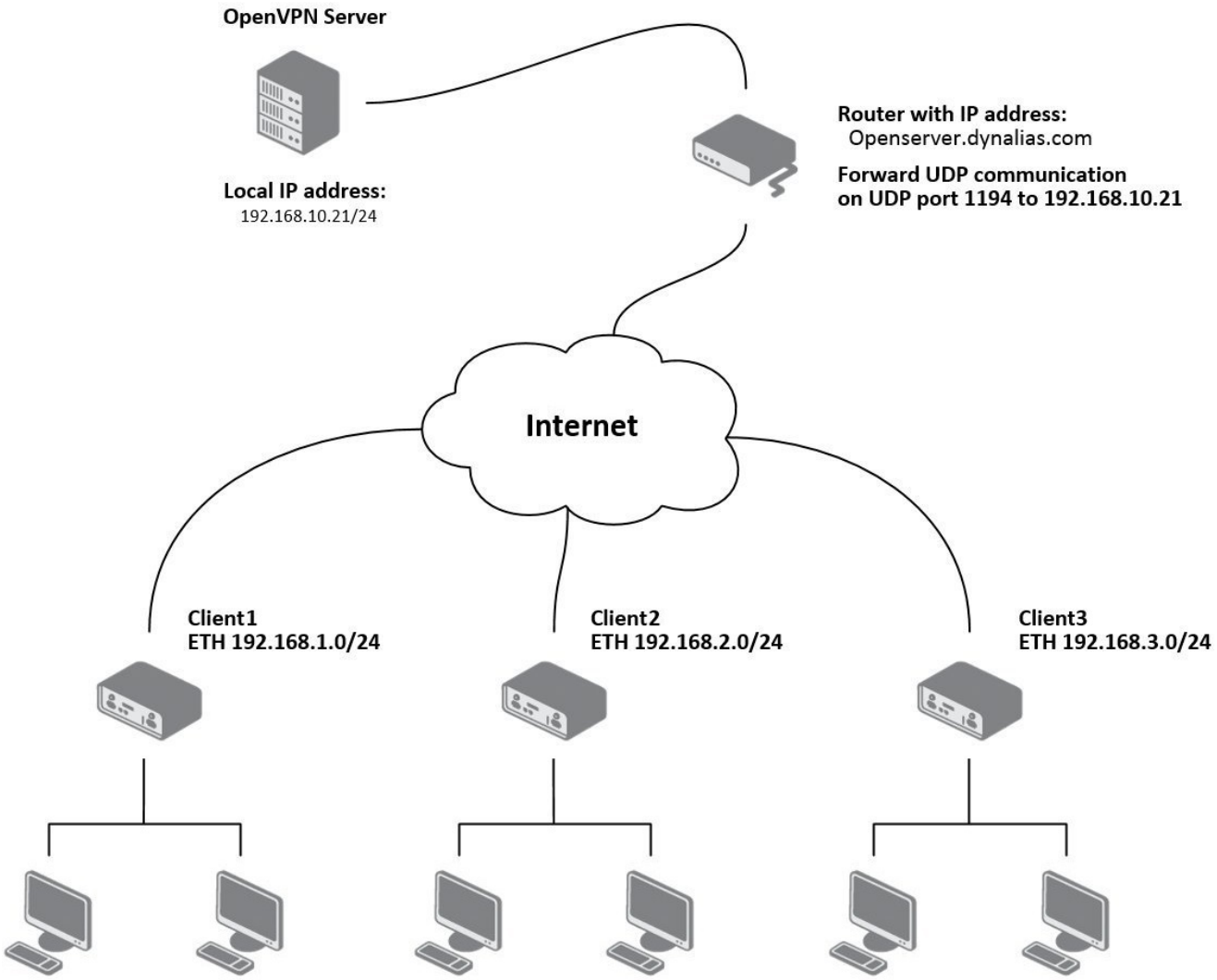

Figure 26: OpenVPN client to client

7.1 OpenVPN server configuration

Configuration file (*.ovpn) stored on the server will contain of:

conf Code

```
server 10.8.0.0 255.255.255.0
port 1194
proto udp
dev tun
comp-lzo
keepalive 10 60
dh dh1024.pem
ca ca.crt
key server.key
cert server.crt
ifconfig-pool-persist ipp.txt
status openvpn-status.log
client-config-dir ccd
client-to-client
persist-key
persist-tun
verb 3
route 192.168.1.0 255.255.255.0
route 192.168.2.0 255.255.255.0
route 192.168.3.0 255.255.255.0
```

In the configuration above is specified cofiguration directory named as *ccd*. This directory is stored on the server in root directory of *OpenVPN* application. File names of client's configuration files stored at this directory must match the names of certifications generated for every single client. In our case, there will be three configuration files with following content (routes between the clients can be defined according to need):

file *ccd\Client1*

conf Code

```
iroute 192.168.1.0 255.255.255.0
push ”route 192.168.2.0 255.255.255.0”
push ”route 192.168.3.0 255.255.255.0”
push ”route 192.168.10.0 255.255.255.0”
```

file *ccd\Client2*

conf Code

```
iroute 192.168.2.0 255.255.255.0
push ”route 192.168.1.0 255.255.255.0”
push ”route 192.168.3.0 255.255.255.0”
push ”route 192.168.10.0 255.255.255.0”
```

file *ccd\Client3*

conf Code

```
iroute 192.168.3.0 255.255.255.0
push ”route 192.168.1.0 255.255.255.0”
push ”route 192.168.2.0 255.255.255.0”
push ”route 192.168.10.0 255.255.255.0”
```

7.2 OpenVPN tunnel configuration on Advantech routers

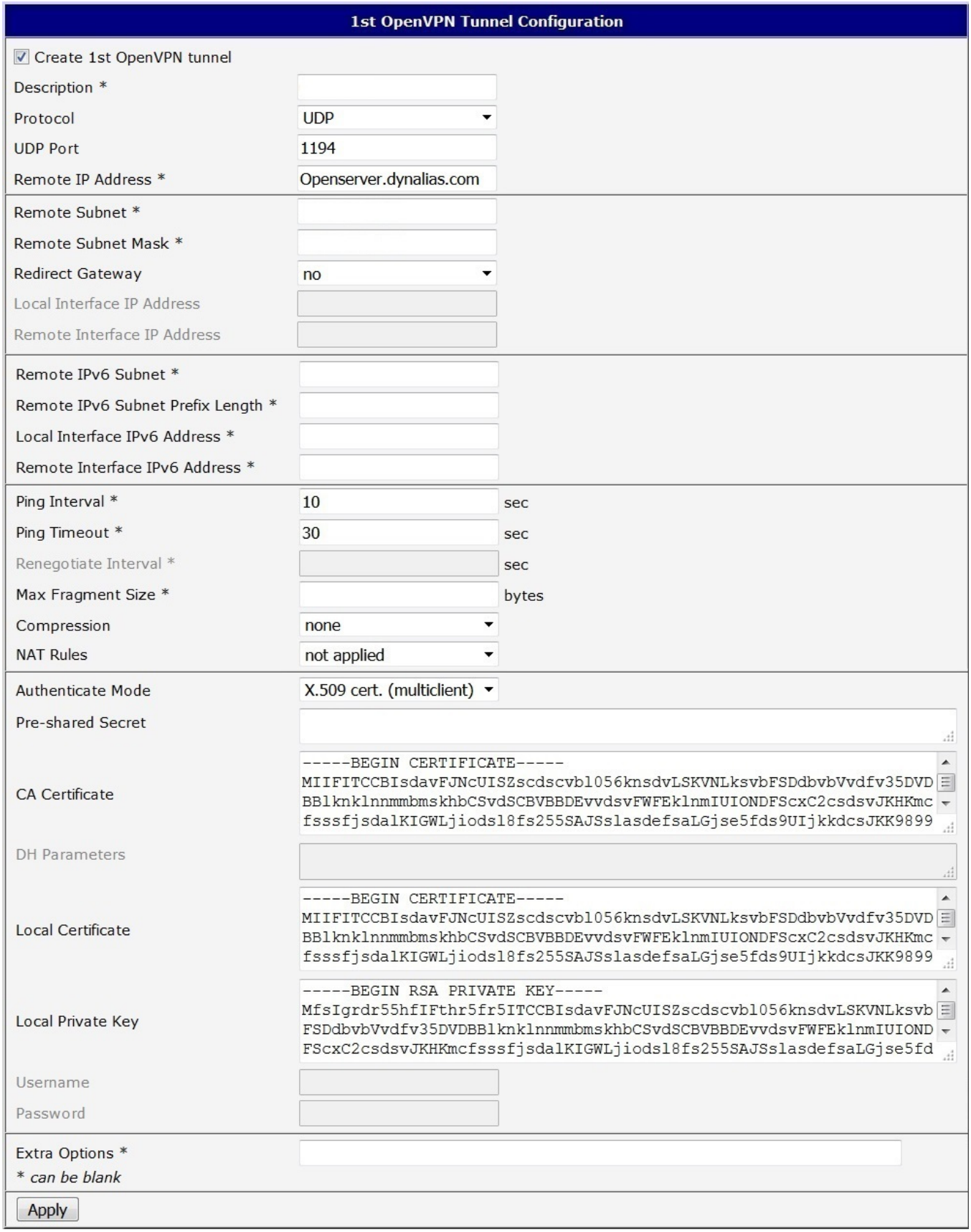

Figure 27: Advantech router configuration

After establishing an OpenVPN tunnel, an interface tun0 and a route in the routing table of the router are displayed on the *Network Status* page.

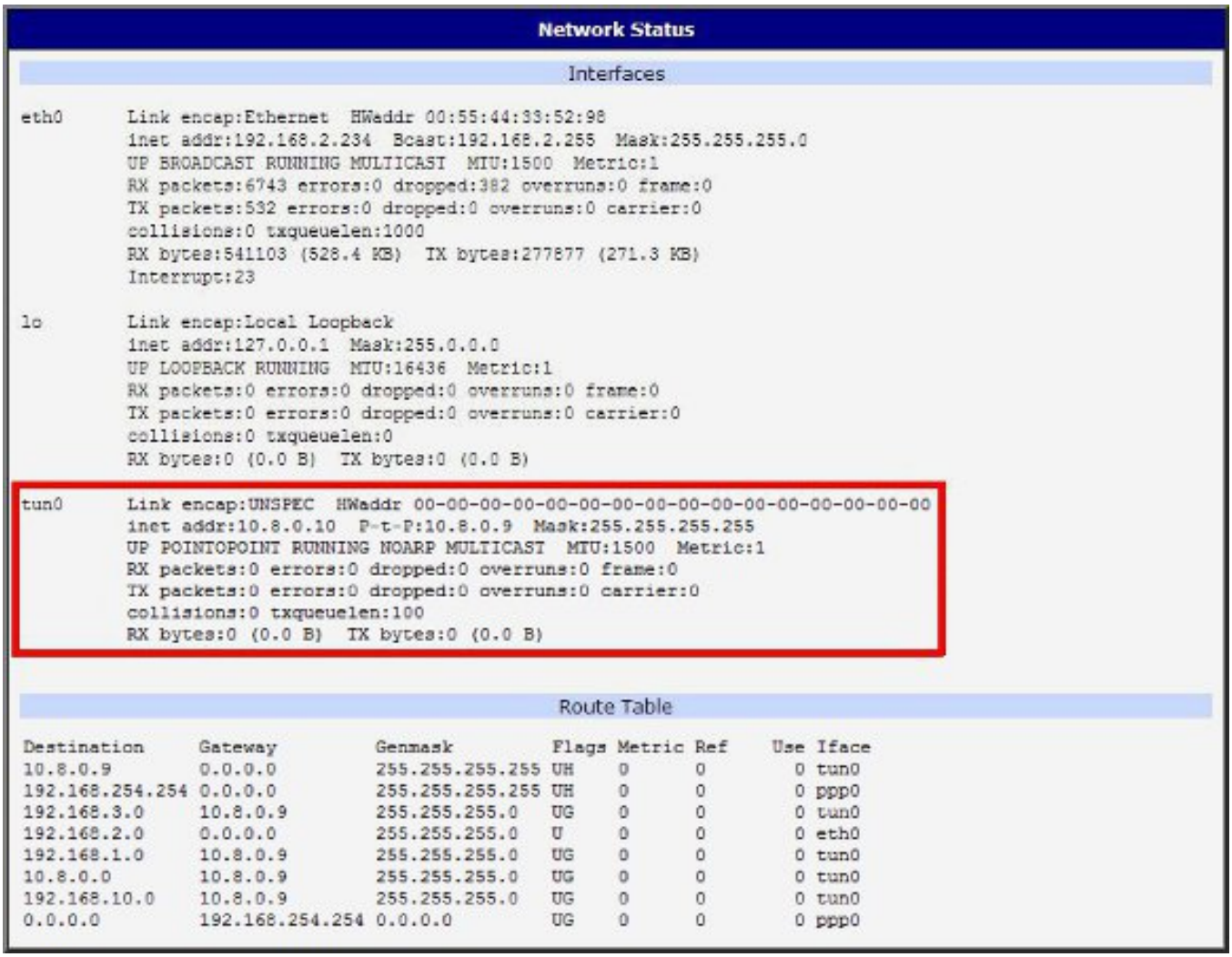

Figure 28: Network Status

It is also possible to check successful establishment of OpenVPN tunnel in the system log (*System Log* item in menu). Listings should end with line *Inicialization Sequence Completed*.

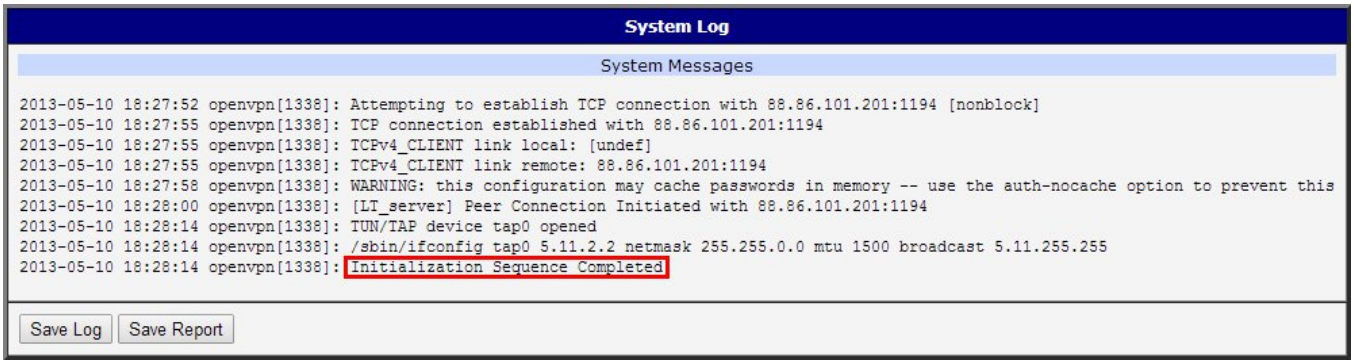

Figure 29: System log

8. Creation of pre-shared key in Windows

Warning

Creating a pre-shared key requires the *OpenVPN* program to be installed. Description of installation can be found in appendix A: *Installation of OpenVPN on Windows* on page A1.

The figure below describes a way to easily generate a pre-shared key. The key is stored into file called *static.key* and its content should be inserted into the *Pre-shared Secret* box in the form for configuration of *OpenVPN* tunnel in the router.

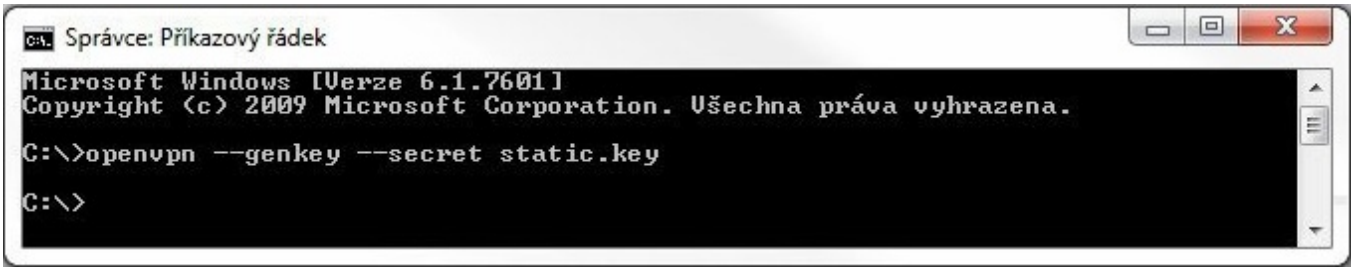

Figure 30: Generating a pre-shared key

Example of pre-shared key:

text Code

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
52dbd2b3380dabd210e8665cf0304de8
ac53ce6bf3ac2605bd3653fd66a113a4
373d57375763de58a38992f580efb97b
817e1b6d61ffbbf559ed9d2c927cef13
39baa06de34c7b4b05df6d4971aa97d0
ec72e4465af647a89e82b335db3dcbb8
a7dd9d190960215ac137e8e2456d2deb
4446b74b3360fe5bf0ac565d4a253a78
9823fd9891db70e190926dbf557c5ad9
cbdb7c0a649a1948b3e5dccce838fc4c
fd6e12b69b7d6bea95c87ee670e85fb1
8ac594f8a9a56921bb2e423dbcd3cbad
650d1543e486ffb956e7a9780925adfe
369e32c5913674bb655b414bde5eb6a0
184c6f2a51f648285f0ab91ea2fe8a20
a9bc715fe96301af90f41f17432e79e3
-----END OpenVPN Static key V1-----
```

9. Creation of certificates in Windows

Warning

Creating certificates requires the *OpenVPN* program and *Easy-RSA* utility to be installed. Description of installation can be found in appendix A: *Installation of OpenVPN on Windows* on page A1 and in appendix B: *Installation of Easy-RSA on Windows* on page B1.

9.1 Introduction

Digital certificates are digitally signed public encryption keys. They are issued by a certification authority (CA). Certificates are kept in X.509 format, which contains information such as the owner of the public key, the certificate issuer or the creator of the digital signature. Certificates are used to identify the counterparty when creating a secure connection (HTTPS, VPN, etc.). On the basis of principle of a trust transfer, it is possible to trust unknown certificates signed by trusted certification authorities. It is typically used a hierarchical model.

9.2 Generating of certificates

Easy-RSA needs to first initialize a directory for the *Public Key Infrastructure* (PKI). Multiple PKIs can be managed with a single installation of *Easy-RSA*, but the default directory is called simply "pki" unless otherwise specified.

First, you need to open an *Easy-RSA* console. It is done by executing of *EasyRSA-Start.bat* file located in *Easy-RSA* root folder. To create or clear out (re-initialize) a new PKI, use the command `./easyrsa init-pki` which will create a new, blank PKI structure ready to be used. Once created, this PKI can be used to make a new CA or generate keypairs.

The next step will be to create a certificate authority (CA) using the command `./easyrsa build-ca`. Now, it is possible to generate certificates and keys for elements in the network (server, client01, client02, ...). In case of a server, use `./easyrsa build-server-full server` command.

For clients use `./easyrsa build-client-full clientXY` command, where clientXY term means a particular client (client01, client02, ...). It follows that the certificates and keys must be generated for each element in the network separately.

Finally, there is a need to generate a Diffie-Hellman parameters (DH key). Use `./easyrsa gen-dh` command to generate the key file. Please note that this process may take a long time.

9.3 Overview of the generated files

The following table describes the meaning of the generated files and their location (to be uploaded to server or to the client).

File location	Description	To be uploaded to
issued\server.crt	Signed certificate of VPN server	server
private\server.key	Personal RSA key of VPN server	server
reqs\server.req	Request for signing	server (not required)
issued\client01.crt	Signed certificate of VPN client	client
private\client01.key	Personal RSA key of VPN client	client
reqs\client01.req	Request for signing	server (not required)
private\ca.key	Key to the CA	secret and secure repository
ca.crt	CA certificate	clients and server
dh.pem	Diffie-Hellman key	server only

Table 11: Overview of the generated files

10. Related Resources

You can obtain all product-related documents, software updates, and supplementary materials on the Advantech *Engineering Portal* at *icr.advantech.com*.

For easy access to specific resources, please refer to the following sections of the portal:

- **Router Support Materials:** To access your router's supporting documents (such as the *Hardware Manual* and *Configuration Manual*), the latest firmware, or other technical resources, navigate to *Support* → *Router Models*. Locate your specific model and select the appropriate tab under the *Documents to download* section. Available tabs include *Brochures*, *Manuals*, *Certificates*, *Firmware*, *Images/3D Models*, *PCN/SA*, and *Others*.
- **Router Apps:** To extend your router's functionality, installation packages and comprehensive manuals for various extension modules are available by navigating to *Download* → *Router Apps*.
- **Application Notes:** For detailed guides, configuration examples, and step-by-step instructions for implementing specific networking features and use cases, navigate to *Download* → *Application Notes*.
- **Development Documents:** If you are interested in custom scripting, programming your own applications, or compiling custom modules, navigate to *Development* page.

Appendix A: Installation of OpenVPN on Windows

The *OpenVPN* installation file can be downloaded from following address: *https://openvpn.net/index.php/download/community-downloads.html*.

Open the downloaded installation file, the following window will be displayed.

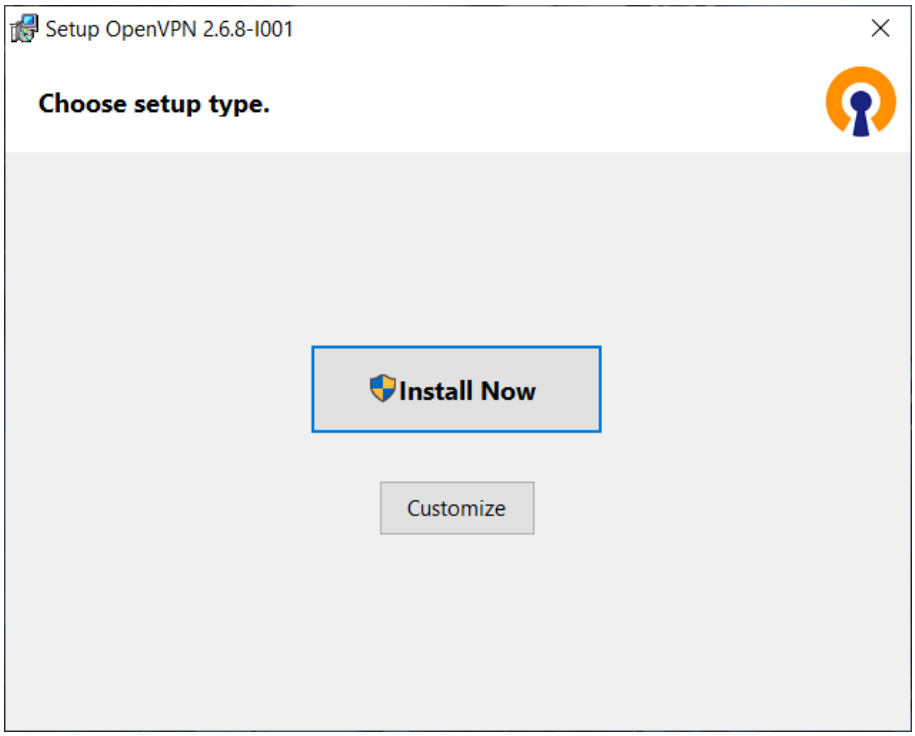

Figure 31: Installation of OpenVPN – basic information

You can either press the *Install Now* button, or choose the Custom Installation by pressing *Customize* button. (see the figure 32) and then press *Install Now* button. After that, press the *Close* button (see the figure 34).

A window will appear allowing you to select the components that will be included in the installation of OpenVPN program. You can also specify a directory in which *OpenVPN* program will be installed. To start the installation press *Install Now* button and wait for completion of the process. Finally, press the *Close* button. (see the figure 34)

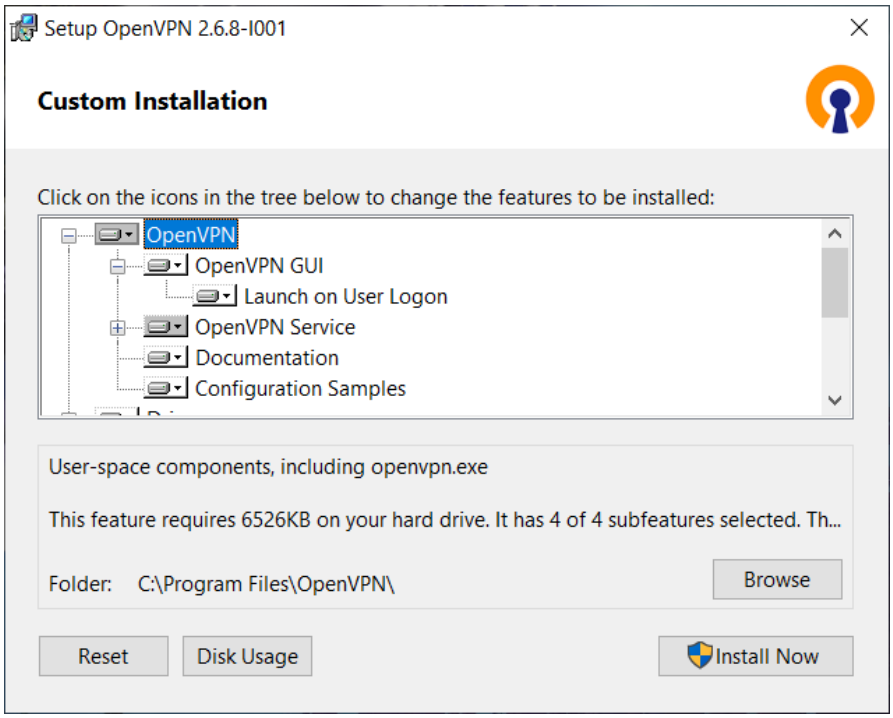

Figure 32: Installation of OpenVPN – components

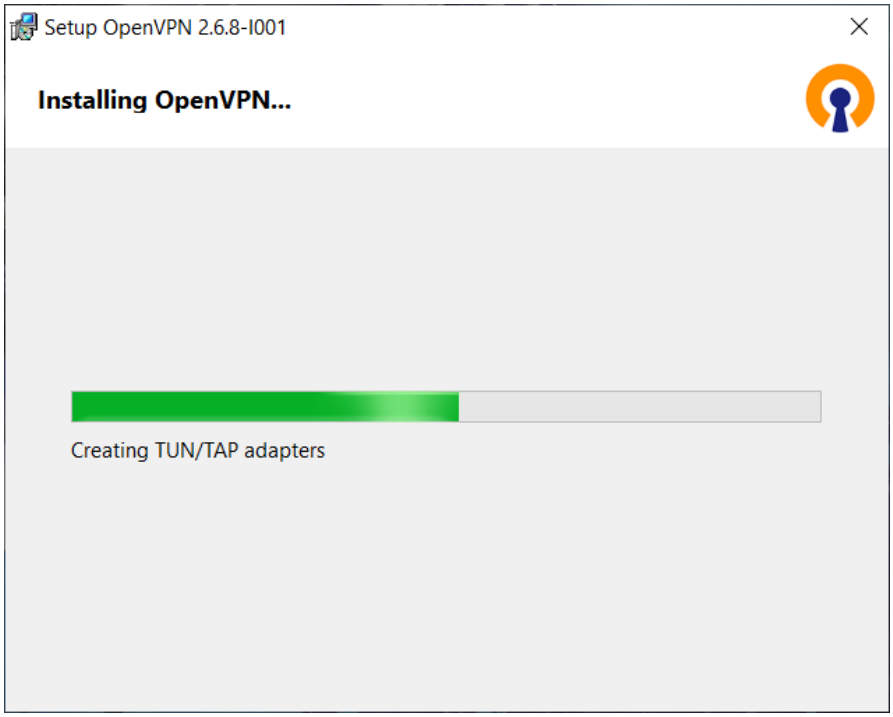

Figure 33: Installation of OpenVPN – progress

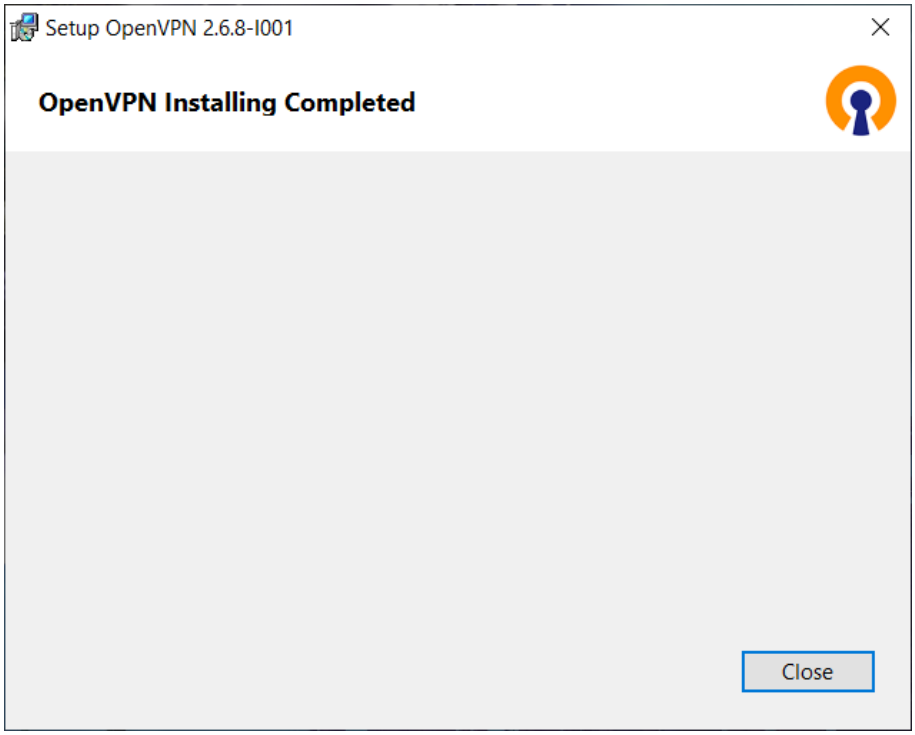

Figure 34: Installation of OpenVPN – complete

Appendix B: Installation of Easy-RSA on Windows

Easy-RSA is a utility for managing X.509 PKI, or Public Key Infrastructure (PKI). The official *Windows* release also comes bundled with the programs necessary to use Easy-RSA. The shell code attempts to limit the number of external programs it depends on. Crypto-related tasks use *openssl* as the functional backend.

The *Easy-RSA* utility was installed along with the *OpenVPN* installation of version 2.2.x and earlier. Since *OpenVPN* version 2.3.x the *Easy-RSA* utility has to be installed separately. It can be downloaded from *https://github.com/OpenVPN/easy-rsa* address.

Easy-RSA's main program is a script, supported by a couple of config files. As such, there is no formal "installation" required. Preparing to use Easy-RSA is as simple as downloading the compressed package and extract it to a location of your choosing. There is no compiling or OS-dependent setup required.

You should install and run *Easy-RSA* as a non-root (non-Administrator) account as root access is not required. The installation package also includes the *doc* folder containing the documentation for the *Easy-RSA* utility.

Public Key Infrastructure (PKI) describes the collection of files and associations between the CA, keypairs, requests, and certificates. An Easy-RSA PKI contains the following directory structure:

- `private\` - Dir with private keys generated on this host.
- `reqs\` - Dir with locally generated certificate requests (for a CA imported requests are stored here).

In a clean PKI no files will exist until, just the bare directories. Commands called later will create the necessary files depending on the operation. When building a CA, a number of new files are created by a combination of Easy-RSA and (indirectly) openssl. The important CA files are:

- `ca.crt` – This is the CA certificate.
- `index.txt` – This is the "master database" of all issued certs.
- `serial` – Stores the next serial number (serial numbers increment).
- `private\ca.key` – This is the CA private key (security-critical).
- `certs_by_serial\` – Dir with all CA–signed certs by serial number.
- `issued\` – Dir with issued certs by commonName.

Easy-RSA 3 no longer needs any configuration file prior to operation, unlike earlier versions. However, the `vars.example` file contains many commented options that can be used to control non-default behavior as required. Reading this file will provide an idea of the basic configuration available. Note that a `vars` file must be named just vars (without an extension) to actively use it. It is not necessary to use this config file unless you wish to change operational defaults. These defaults should be fine for many uses without the need to copy and edit the `vars` file.

Invoking *Easy-RSA* is done through your preferred shell. For Windows OS, use the `EasyRSA-Start.bat` program to provide a POSIX-shell environment suitable for using Easy-RSA. The basic format for running commands is `./easyrsa command [cmd-opts]` where `command` is the name of a command to run, and `cmd-opts` are any options to supply to the command. Some commands have mandatory or optional cmd-opts. Note the leading `.\` component of the command. This is required in Unix-like environments and may be a new concept to some Windows users.

General usage and command help can be shown with `./easyrsa help [command]`. When run without any command, general usage and a list of available commands are shown; when a command is supplied, detailed help output for that command is shown.