Template:Networking rut850 manual vpn

□

Contents

Summary

Virtual Private Network (**VPN**) is a method of connecting multiple private networks across the Internet. VPNs can serve to achieve many different goals, but some of its main purposes are:

- access between remote private networks;
- data encryption;
- anonymity when browsing the Internet.

The information in this page is updated in accordance with the **[[Media:{{{fw_version}}}_WEBUI.bin|{{{fw_version}}}]]** firmware version.

This page is an overview of different types of VPNs supported by {{{name}}} routers.

OpenVPN

OpenVPN is an open-source software application that implements virtual private network (VPN) techniques for creating secure point-to-point or site-to-site connections in routed or bridged configurations and remote access facilities. It is often regarded as being the most universal VPN protocol because of its flexibility, support of SSL/TLS security, multiple encryption methods, many networking features and compatibility with most OS platforms.

{{{name}}} routers run OpenVPN version **2.4.5**.

OpenVPN client

An **OpenVPN client** is an entity that initiates a connection to an OpenVPN server. To create a new

client instance, go to the *Services → VPN → OpenVPN* section, select *Role: Client*, enter a custom name and click the 'Add New' button. An OpenVPN client instance with the given name will appear in the "OpenVPN Configuration" list. A maximum of six OpenVPN client instances are allowed to be added.

To begin configuration, click the 'Edit' button next to the client instance. Refer to the figure and table below for information on the OpenVPN client's configuration fields:

[[File:{{{file_openvpn_client_config}}}]]

Field	Value	Description
Enable	yes | no; default: **no**	Turns the OpenVPN instance on or off.
TUN/TAP	TUN (tunnel) | TAP (bridged); default: **TUN (tunnel)**	Virtual network device type. • **TUN** - a virtual point-to-point IP link which operates at the network layer (OSI layer 3), used when routing is required. • **TAP** - a virtual Ethernet adapter (switch), operates at the data link layer (OSI layer 2), used when bridging is required.
Protocol	UDP | TCP; default: **UDP**	Transfer protocol used for the OpenVPN connection. • **Transmission Control Protocol** (**TCP**) - most commonly used protocol in the Internet Protocol (IP) suite. It ensures the recipient will receive packets in the order they were sent by numbering, analysing response messages, checking for errors and resending them if an issue occurs. It should be used when reliability is crucial (for example, in file transfer). • **User Datagram Protocol** (**UDP**) - packets are sent to the recipient without error-checking or back-and-forth quality control, meaning that when packets are lost, they are gone forever. This makes it less reliable but faster than TCP; therefore, it should be used when transfer speed is crucial (for example, in video streaming, live calls).
Port	integer [0..65535]; default: **1194**	TCP/UDP port number used for the connection. Make sure it matches the port number specified on the server side. **NOTE**: traffic on the selected port will be automatically allowed in the router's firewall rules.
LZO	yes | no; default: **no**	Turns LZO data compression on or off.

Encryption	DES-CBC 64 | RC2-CBC 128 | DES-EDE-CBC 128 | DES-EDE3-CBC 192 | DESX-CBC 192 | BF-CBC 128 | RC2-40-CBC 40 | CAST5-CBC 128 | RC2-40CBC 40 | CAST5-CBC 128 | RC2-64-CBC 64| AES-128-CBC 128 | AES-192-CBC 192 | AES-256-CBC 256 | none; default: **BF-CBC 128**	Algorithm used for packet encryption.
Authentication	TLS | Static Key | Password | TLS/Password; default: **TLS**	Authentication mode, used to secure data sessions. • **Static key** is a secret key used for server–client authentication. • **TLS** authentication mode uses X.509 type certificates: - Certificate Authority (CA) - Client certificate - Client key All mentioned certificates can be generated using OpenVPN or Open SSL utilities on any type of host machine. One of the most popular utilities used for this purpose is called Easy-RSA. • **Password** is a simple username/password based authentication where the owner of the OpenVPN server provides the login data. • **TLS/Password** uses both TLS and username/password authentication.
TLS: TLS cipher	All | DHE+RSA | Custom; default: **All**	Packet encryption algorithm cipher.
TLS: Allowed TLS ciphers	All | DHE+RSA | Custom; default: **All**	A list of TLS ciphers accepted for this connection.
Remote host/IP address	ip; default: **none**	IP address or hostname of an OpenVPN server.
Resolve retry	integer | infinite; default: **infinite**	In case server hostname resolve fails, this field indicates the amount of time (in seconds) to retry the resolve. Specify *infinite* to retry indefinitely.
Keep alive	two integers separated by a space; default: **none**	Defines two time intervals: the first is used to periodically send ICMP requests to the OpenVPN server, the second one defines a time window, which is used to restart the OpenVPN service if no ICMP response is received during the specified time slice. When this value is specfiied on the OpenVPN server, it overrides the 'keep alive' values set on client instances. **Example**: *10 120*
Static key: Local tunnel endpoint IP	ip; default: **none**	IP address of the local OpenVPN network interface.
Static key: Remote tunnel endpoint IP	ip; default: **none**	IP address of the remote OpenVPN network (server) interface.

Remote network IP address	ip; default: **none**	LAN IP address of the remote network (server).
Remote network IP netmask	netmask; default: **none**	LAN IP subnet mask of the remote network (server).
Password: User name	string; default: **none**	Username used for authentication to the OpenVPN server.
Password: Password	string; default: **none**	Password used for authentication to the OpenVPN server.
Extra options	string; default: **none**	Extra OpenVPN options to be used by the OpenVPN instance.
TLS/Password: HMAC authentication algorithm	none | SHA1 | SHA256 | SHA384 | SHA512; default: **SHA1**	HMAC authentication algorithm type.
TLS/Password: Additional HMAC authentication	yes | no; default: **no**	An additional layer of HMAC authentication on top of the TLS control channel to protect against DoS attacks.
TLS/Password: HMAC authentication key	.key file; default: **none**	Uploads an HMAC authentication key file.
TLS/Password: HMAC key direction	0 | 1 | none; default: **1**	The value of the key direction parameter should be complementary on either side (client and server) of the connection. If one side uses *0*, the other side should use *1*, or both sides should omit the parameter altogether.
TLS/Password: Certificate authority	.ca file; default: **none**	Certificate authority is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate.
TLS: Client certificate	.crt file; default: **none**	Client certificate is a type of digital certificate that is used by client systems to make authenticated requests to a remote server. Client certificates play a key role in many mutual authentication designs, providing strong assurances of a requester's identity.
TLS: Client key	.key file; default: **none**	Authenticates the client to the server and establishes precisely who they are.
TLS: Private key decryption password (optional)	string; default: **none**	A password used to decrypt the server's private key. Use only if server's .key file is encrypted with a password.
Static key: Static pre-shared key	.key file; default: **none**	Uploads a secret key file used for server–client authentication.

Additional notes:

- Some configuration fields become available only when certain other parameters are selected. The names of the parameters are followed by a prefix that specifies the authentication type under which they become visible. Different color codes are used for different prefixes:
  - Red for Authentication: TLS
  - Purple for Authentication: Static key
  - Blue for Authentication: Password
- After changing any of the parameters, don't forget to click the **Save** button located at the bottom-right side of the page.

OpenVPN server

An **OpenVPN server** is an entity that waits for incoming connections from OpenVPN clients. To create a new server instance, go to the *Services → VPN → OpenVPN* section, select *Role: Server*, enter a custom name and click the 'Add New' button. An OpenVPN server instance with the given name will appear in the "OpenVPN Configuration" list. Only one OpenVPN server instance is allowed to be added.

A server needs to have a public IP address in order to be available from the public network (the Internet).

To begin configuration, click the 'Edit' button next to the server instance. Refer to the figure and table below for information on the OpenVPN server's configuration fields:

[[File:{{{file_openvpn_server_config}}}]]

Field	Value	Description
Enable	yes | no; default: **no**	Turns the OpenVPN instance on or off.
TUN/TAP	TUN (tunnel) | TAP (bridged); default: **TUN (tunnel)**	Virtual network device type. • **TUN** - a virtual point-to-point IP link which operates at the network layer (OSI layer 3), used when routing is required. • **TAP** - a virtual Ethernet adapter (switch), operates at the data link layer (OSI layer 2), used when bridging is required.
Protocol	UDP | TCP; default: **UDP**	Transfer protocol used for the connection. • **Transmission Control Protocol** (**TCP**) - most commonly used protocol in the Internet Protocol (IP) suite. It ensures the recipient will receive packets in the order they were sent by numbering, analysing response messages, checking for errors and resending them if an issue occurs. It should be used when reliability is crucial (for example, file transfer). • **User Datagram Protocol** (**UDP**) - packets are sent to the recipient without error-checking or back-and-forth quality control, meaning that when packets are lost, they are gone forever. This makes it less reliable but faster than TCP; therefore, it should be used when transfer speed is crucial (for example, video streaming, live calls).
Port	integer [0..65535]; default: **1194**	TCP/UDP port number used for the connection. Make sure it matches the port number specified on the server side. **NOTE**: traffic on the selected port will be automatically allowed in the router's firewall rules.
LZO	yes | no; default: **no**	Turns LZO data compression on or off.

Encryption	DES-CBC 64 | RC2-CBC 128 | DES-EDE-CBC 128 | DES-EDE3-CBC 192 | DESX-CBC 192 | BF-CBC 128 | RC2-40-CBC 40 | CAST5-CBC 128 | RC2-40CBC 40 | CAST5-CBC 128 | RC2-64-CBC 64| AES-128-CBC 128 | AES-192-CBC 192 | AES-256-CBC 256 | none; default: **BF-CBC 128**	Algorithm used for packet encryption.
Authentication	TLS | Static Key | TLS/Password; default: **TLS**	Authentication mode, used to secure data sessions. • **Static key** is a secret key used for server–client authentication. • **TLS** authentication mode uses X.509 type certificates: - Certificate Authority (CA) - Client certificate - Client key All mentioned certificates can be generated using OpenVPN or Open SSL utilities on any type of host machine. One of the most popular utilities used for this purpose is called Easy-RSA. • **TLS/Password** uses both TLS and username/password authentication.
Static key: Local tunnel endpoint IP	ip; default: **none**	IP address of the local OpenVPN network interface.
Static key: Remote tunnel endpoint IP	ip; default: **none**	IP address of the remote OpenVPN network (client) interface.
Static key: Remote network IP address	ip; default: **none**	LAN IP address of the remote network (client).
Static key: Remote network IP netmask	netmask; default: **none**	LAN IP subnet mask of the remote network (client).
TLS/TLS/Password: TLS cipher	All | DHE+RSA | Custom; default: **All**	Packet encryption algorithm cipher.
TLS/Password: Allowed TLS ciphers	All | DHE+RSA | Custom; default: **All**	A list of TLS ciphers accepted for this connection.
TLS/TLS/Password: Client to client	yes | no; default: **no**	Allows OpenVPN clients to communicate with each other on the VPN network.
TLS/TLS/Password: Keep alive	two integers separated by a space; default: **none**	Defines two time intervals: the first is used to periodically send ICMP requests to the OpenVPN server, the second one defines a time window, which is used to restart the OpenVPN service if no ICMP response is received during the specified time slice. When this value is specifiied on the OpenVPN server, it overrides the 'keep alive' values set on client instances. **Example**: *10 120*

TLS/TLS/Password: Virtual network IP address	ip; default: **none**	IP address of the OpenVPN network.
TLS/TLS/Password: Virtual network netmask	netmask; default: **none**	Subnet mask of the OpenVPN network.
TLS/TLS/Password: Push option	OpenVPN options; default: **none**	Push options are a way to "push" routes and other additional OpenVPN options to connecting clients.
TLS/TLS/Password: Allow duplicate certificates	yes | no; default: **no**	When enabled allows multiple clients to connect using the same certificates.
TLS/Password: User name	string; default: **none**	Username used for authentication to this OpenVPN server.
TLS/Password: Password	string; default: **none**	Password used for authentication to this OpenVPN server.
Static key: Static pre-shared key	.key file; default: **none**	Uploads a secret key file used for server–client authentication.
TLS/TLS/Password: Certificate authority	.ca file; default: **none**	Certificate authority is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate.
TLS/TLS/Password: Server certificate	.crt file; default: **none**	A type of digital certificate that is used to identify the OpenVPN server.
TLS/TLS/Password: Server key	.key file; default: **none**	Authenticates clients to the server.
TLS/TLS/Password: Diffie Hellman parameters	.pem file; default: **none**	DH parameters define how OpenSSL performs the Diffie-Hellman (DH) key-exchange.
TLS/TLS/Password: CRL file (optional)	.pem file | .crl file; default: **none**	A certificate revocation list (CRL) file is a list of certificates that have been revoked by the certificate authority (CA). It indicates which certificates are no longer acccepted by the CA and therefore cannot be authenticated to the server.

Additional notes:

- Some configuration fields become available only when certain other parameters are selected. The names of the parameters are followed by a prefix that specifies the authentication type under which they become visible. Different color codes are used for different prefixes:
  - Red for Authentication: TLS
  - Purple for Authentication: Static key
  - Blue for Authentication: TLS/Password
- After changing any of the parameters, don't forget to click the **Save** button located at the bottom-right side of the page.

TLS Clients

TLS Clients is a way to differentiate clients by their Common Names (CN), which are found in the client certificate file. It can be used to assign specific VPN addresses to corresponding clients and bind them to their LAN addresses, making the server aware of which client has which LAN IP

address.

The TLS Clients section can be found in the OpenVPN Server configuration window, provided that the OpenVPN server uses TLS or TLS/Password authentication methods. To create a new TLS client, type in the new client's name in the text field found bellow the TLS Clients tab and click the 'Add' button. Refer to the figure and table below for information on the TLS Clients' configuration fields:

[[File:{{{file_openvpn_tls_clients_config}}}]]

Field	Value	Description
VPN instance name	string; default: **none**	Indicates which OpenVPN instance the TLS Client will be associated with. When left empty, this field is filled automatically.
Endpoint name	string; default: **none**	A custom name for the client.
Common name (CN)	string; default: **none**	Client's Common Name (CN) found in the client certificate file.
Virtual local endpoint	ip; default: **none**	Client's local address in the virtual network.
Virtual remote endpoint	ip; default: **none**	Client's remote address in the virtual network.
Private network	ip; default: **none**	Client's private network (LAN) IP address.
Private netmask	ip; default: **none**	Client's private network (LAN) IP netmask.

IPsec

To create a new IPsec instance, go to the *Services → VPN → IPsec* section, enter a custom name and click "Add". An IPsec instance with the given name will appear in the "IPsec Configuration" list.

To begin configuration, click the 'Edit' button located next to the instance.

IPsec configuration

The **IPsec configuration** section is used to configure the main parameters of an IPsec connection. Refer to the figure and table below for information on the configuration fields located in the general settings section.

[[File:{{{file_ipsec_config}}}]]

Field	Value	Description
Enable	yes | no; default: **no**	Turns the IPsec instance on or off.
Mode	Main | Aggressive; default: **Main**	Internet Security and Key Management Protocol (ISAKMP) phase 1 exchange mode. • **Main** - performs three two-way exchanges between the initiator and the receiver (a total of 9 messages). • **Aggressive** - performs fewer exchanges than main mode (a total of 6 messages) by storing most data into the first exchange. In aggressive mode, the information is exchanged before there is a secure channel, making it less secure but faster than main mode.

My identifier type	FQDN | User FQDN | Address; default: **FQDN**	Defines the type of identity used in user (IPsec instance) authentication. • **FQDN** - identity defined by fully qualified domain name. It is the complete domain name for a host (for example, *something.somedomain.com*). Only supported with IKEv2. • **User FQDN** - identity defined by fully qualified username string (for example, *username@something.somedomain.com*). Only supported with IKEv2. • **Address** - identity by IP address.
My identifier	ip | string; default: **none**	Defines how the user (IPsec instance) will be identified during authentication.
Dead Peer Detection	yes | no; default: **no**	A function used during Internet Key Exchange (IKE) to detect a "dead" peer. It used to reduce traffic by minimizing the number of messages when the opposite peer in unavailable and as failover mechanism.
Pre shared key	string; default: **none**	A shared password used for authentication between IPsec peers before a secure channel is established.
Remote VPN endpoint	host | ip; default: **none**	IP address or hostname of the remote IPsec instance.
IP address/Subnet mask	ip/netmask; default: **none**	Remote network IP address and subnet mask used to determine which part of the network can be accessed in the VPN network. Netmask range [0..32]. This value must differ from the device's LAN IP.
Enable keepalive	yes | no; default: **no**	When enabled, the instance sends ICMP packets to the specified host at the specified frequency. If no response is received, the router will attempt to restart the connection.
Host	host | ip; default: **none**	Hostname or IP address to which keepalive ICMP packets will be sent to.
Ping period (sec)	integer [0..9999999]; default: **none**	The frequency at which keepalive ICMP packets will be sent to the specified host or IP address.

Additional notes:

- Some configuration fields become available only when certain other parameters are selected. The names of the parameters are followed by a prefix that specifies the authentication type under which they become visible. Different color codes are used for different prefixes:
  - Red for Type: Tunnel
  - Purple for Type: Transport
  - Blue for Dead Peer Detection: Enabled
- After changing any of the parameters, don't forget to click the **Save** button located at the bottom-right side of the page.

Phase settings

IKE (Internet Key Exchange) is a protocol used to set up security associations (SAs) for the IPsec connection. This process is required before the IPsec tunnel can be established. It is done in two phases:

Phase	Mode

Phase 1
• Establishes a secure channel between peers
• Authenticates peers
• Negotiates SA policy
• Shares secret keys
• Establishes secure tunnel for phase 2

Phase 2
• Sets up matching IPsec SAs
• Periodically renegotiates IPsec SAs

Main mode (figure 1)
• 6 packets exchanged
• Identity protected during exchange

Aggressive mode (figure 2)
• 3 packets exchanged
• Identity information exchanged before a secure channel is established

Quick mode
• 3 packets exchanged
• IPsec SA parameters (ESP/AH, SHA/MD5) established
• SA lifetime set

Figure 1
[[File:{{{file_ipsec_main_mode}}}]]

Figure 2
[[File:{{{file_ipsec_aggressive_mode}}}]]

[[File:{{{file_ipsec_phase}}}]]

Field	Value	Description
Encryption algorithm	DES | 3DES | AES128 | AES192 | AES256; default: **3DES**	Algorithm used for data encryption.
Authentication/Hash algorithm	MD5 | SHA1 | SHA256 | SHA384 | SHA512; default: **SHA1**	Algorithm used for exchanging authentication and hash information.
DH group/PFS group	MODP768 | MODP1024 | MODP1536 | MODP2048 | MODP3072 | MODP4096; default: **MODP1536**	Diffie-Hellman (DH) group used in the key exchange process. Higher group numbers provide more security, but take longer and use more resources to compute the key.
Lifetime	integer; default: **8 hours**	Defines a time period after which the phase will re-initiate its exchange of information.

L2TP

In computer networking, **Layer 2 Tunneling Protocol** (**L2TP**) is a tunneling protocol used to support virtual private networks (VPNs). It is more secure than PPTP but, because it encapsulates the transferred data twice, but it is slower and uses more CPU power.

L2TP client

An **L2TP client** is an entity that initiates a connection to an L2TP server. To create a new client instance, go to the *Services → VPN → L2TP* section, select *Role: Client*, enter a custom name and click the 'Add New' button. An L2TP client instance with the given name will appear in the "L2TP Configuration" list.

To begin configuration, click the 'Edit button located next to the client instance. Refer to the figure and table below for information on the L2TP client's configuration fields:

[[File:{{{file_l2tp_client_config}}}]]

Field	Value	Description

Enable	yes | no; default: **no**	Turns the L2TP instance on or off.
Server	ip | host; default: **none**	IP address or hostname of an L2TP server.
Username	string; default: **none**	Username used for authentication to the L2TP server.
Password	string; default: **none**	Password used for authentication to the L2TP server.
Default route	yes | no; default: **no**	When turned on, this connection will become the router's default route. This means that all traffic directed to the Internet will go through the L2TP server and the server's IP address will be seen as this device's source IP to other hosts on the Internet. **NOTE**: this can only be used when [[{{{name}}}_WAN#Operation_Modes Failover|Backup WAN]] is turned off.

L2TP server

An **L2TP server** is an entity that waits for incoming connections from L2TP clients. To create a new server instance, go to the *Services → VPN → L2TP* section, select *Role: Server*, enter a custom name and click the 'Add New' button. An L2TP server instance with the given name will appear in the "L2TP Configuration" list. Only one L2TP server instance is allowed to be added.

A server needs to have a public IP address in order to be available from the public network (the Internet).

To begin configuration, click the 'Edit' button located next to the server instance. Refer to the figure and table below for information on the L2TP server's configuration fields:

[[File:{{{file_l2tp_server_config}}}]]

Field	Value	Description
Enable	yes | no; default: **no**	Turns the L2TP instance on or off.
Local IP	ip; default: **192.168.0.1**	IP address of this L2TP network interface.
Remote IP range begin	ip; default: **192.168.0.20**	L2TP IP address leases will begin from the address specified in this field.
Remote IP range end	ip; default: **192.168.0.30**	L2TP IP address leases will end with the address specified in this field.
User name	string; default: **user**	Username used for authentication to this L2TP server.
Password	string; default: **pass**	Password used for authentication to this L2TP server.
L2TP Client's IP	ip; default: **none**	Assigns an IP address to the client that uses the adjacent authentication info. This field is optional and if left empty the client will simply receive an IP address from the IP pool defined above.

[[Category:{{{name}}} Services section]]