



OpenVPN Management

Basic OpenVPN Commands

<code>openvpn --config client.conf</code>	Start OpenVPN client with a specific configuration file.
<code>openvpn --daemon --config server.conf</code>	Start OpenVPN server in daemon mode (background process).
<code>systemctl start openvpn@client</code>	Start OpenVPN client service (using systemd).
<code>systemctl stop openvpn@server</code>	Stop OpenVPN server service (using systemd).
<code>systemctl status openvpn@client</code>	Check the status of the OpenVPN client service.
<code>journalctl -u openvpn@server</code>	View OpenVPN server logs (using journalctl).

Configuration File Directives

<code>client</code>	Specifies that this is a client configuration.
<code>server 10.8.0.0 255.255.255.0</code>	Configures OpenVPN server with a specific subnet.
<code>remote myvpn.example.com 1194</code>	Specifies the remote VPN server address and port.
<code>dev tun</code>	Uses a TUN (Layer 3) virtual network device.
<code>dev tap</code>	Uses a TAP (Layer 2) virtual network device.
<code>proto udp</code>	Uses UDP protocol for the VPN connection.
<code>proto tcp</code>	Uses TCP protocol for the VPN connection.
<code>tls-client</code>	Enables TLS client mode.

Troubleshooting

Check OpenVPN logs for error messages. Common issues include certificate errors, firewall problems, and incorrect configuration settings.
Verify that the OpenVPN service is running using <code>systemctl status openvpn@client</code> or <code>systemctl status openvpn@server</code> .
Use <code>ping</code> and <code>traceroute</code> to test connectivity to the VPN server and other network resources.

WireGuard Essentials

Basic WireGuard Commands

<code>wg-quick up wg0</code>	Activate WireGuard interface <code>wg0</code> .
<code>wg-quick down wg0</code>	Deactivate WireGuard interface <code>wg0</code> .
<code>wg show</code>	Show current WireGuard status and configuration.
<code>wg show wg0</code>	Show configuration and status for interface <code>wg0</code> .
<code>wg genkey tee privatekey</code>	Generate a private key and save it to <code>privatekey</code> .
<code>wg pubkey <privatekey tee publickey</code>	Generate a public key from a private key and save it to <code>publickey</code> .

Configuration File Parameters

<code>[Interface]</code>	Section for interface-specific settings.
<code>PrivateKey = <private_key></code>	Sets the private key for the interface.
<code>Address = 10.0.0.2/24</code>	Sets the IP address and subnet for the interface.
<code>ListenPort = 51820</code>	Sets the port WireGuard listens on.
<code>[Peer]</code>	Section for peer-specific settings.
<code>PublicKey = <public_key></code>	Sets the peer's public key.
<code>AllowedIPs = 0.0.0.0/0</code>	Sets the allowed IPs for the peer. <code>0.0.0.0/0</code> allows all IPs.
<code>Endpoint = example.com:51820</code>	Sets the peer's endpoint (IP address and port).

Troubleshooting

Ensure that the WireGuard interface is active using <code>wg show wg0</code> . Check for any errors in the output.
Verify that the firewall allows UDP traffic on the specified port (default is 51820).
Use <code>tcpdump</code> or <code>wireshark</code> to capture and analyze network traffic to identify any connectivity issues.

IPsec VPN Configuration

StrongSwan Commands

<code>ipsec start</code>	Start the IPsec service.
<code>ipsec stop</code>	Stop the IPsec service.
<code>ipsec restart</code>	Restart the IPsec service.
<code>ipsec status</code>	Check the status of IPsec connections.
<code>ipsec up</code> <code><connection_name></code> <code>></code>	Initiate a specific IPsec connection.
<code>ipsec down</code> <code><connection_name></code> <code>></code>	Terminate a specific IPsec connection.

IPsec Configuration Files

<code>ipsec.conf</code>	Main configuration file for IPsec connections.
<code>ipsec.secret</code> <code>s</code>	File containing pre-shared keys or RSA private keys.
<code>left=%any</code>	Local IP address or identifier. <code>%any</code> means any address.
<code>right=192.16</code> <code>8.1.1</code>	Remote IP address or identifier.
<code>auto=start</code>	Automatically start the connection when IPsec starts.
<code>keyexchange=</code> <code>ikev2</code>	Use IKEv2 key exchange protocol.
<code>ike=aes256-</code> <code>sha256-</code> <code>modp2048!</code>	IKE (Phase 1) encryption, hash, and DH group.
<code>esp=aes256-</code> <code>sha256!</code>	ESP (Phase 2) encryption and hash algorithm.

Troubleshooting

Check the IPsec logs for errors. These are typically located in <code>/var/log/auth.log</code> or <code>/var/log/syslog</code> .
Use <code>tcpdump</code> to capture packets and analyze the IKE and ESP exchanges.
Verify that the firewall rules allow UDP ports 500 and 4500 for IKE and NAT-T traffic, respectively.

Network Utility Commands

Basic Network Commands

<code>ping <host></code>	Test network connectivity to a host.
<code>traceroute</code> <code><host></code>	Trace the route packets take to reach a host.
<code>ifconfig</code> or <code>ip addr</code>	Display network interface configuration.
<code>netstat -rn</code> or <code>ip route</code>	Display the routing table.
<code>nslookup</code> <code><host></code>	Query DNS to find the IP address of a host.
<code>tcpdump -i</code> <code><interface></code> <code><filter></code>	Capture network traffic on a specific interface with a filter.

VPN-Specific Network Checks

<code>ifconfig</code> <code>tun0</code> or <code>ip</code> <code>addr show</code> <code>tun0</code>	Check the configuration of the TUN interface (OpenVPN).
<code>ifconfig</code> <code>tap0</code> or <code>ip</code> <code>addr show</code> <code>tap0</code>	Check the configuration of the TAP interface (OpenVPN).
<code>wg show wg0</code>	Check the status of the WireGuard interface.
<code>ping -I tun0</code> <code><ip_address></code>	Ping a host using the TUN interface.
<code>traceroute -i</code> <code>tun0</code> <code><ip_address></code>	Trace the route via the TUN interface.

Firewall Commands (iptables)

<code>iptables -L</code>	List current iptables rules.
<code>iptables -A INPUT -i tun0 -j ACCEPT</code>	Allow traffic from the TUN interface.
<code>iptables -A FORWARD -i tun0 -j ACCEPT</code>	Forward traffic through the TUN interface.
<code>iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE</code>	Enable NAT for VPN traffic.
<code>iptables -P FORWARD DROP</code>	Set default forward policy to DROP.