VPN Configuration Cheat Sheet

A comprehensive cheat sheet covering VPN configuration across various platforms and protocols, including OpenVPN, WireGuard, and IPsec. This guide provides essential commands, configuration examples, and troubleshooting tips for setting up and managing VPN connections.

OpenVPN Configuration

Server Configuration

OpenVPN server configuration typically resides in `/etc/openvpn/server.conf` or a similar path. Here's a basic configuration example:

```
port 1194
proto udp
dev tun
ca /etc/openvpn/easy-rsa/pki/ca.crt
cert /etc/openvpn/easy-rsa/pki/issued/server.crt
key /etc/openvpn/easy-rsa/pki/private/server.key
dh /etc/openvpn/dh.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status.log
log-append  openvpn.log
verb 3
```

Key Parameters:

- `port`: The port OpenVPN listens on.
- `proto`: Protocol used (UDP or TCP).
- `dev`: Tunnel device (tun or tap).
- `ca`, `cert`, `key`: Paths to CA certificate, server certificate, and private key.
- `dh`: Diffie-Hellman parameters.
- `server`: VPN subnet and netmask.
- `push`: Options pushed to clients (e.g., DNS servers, gateway redirection).

Client Configuration

Client configuration files (e.g., `client.conf`) define how clients connect to the OpenVPN server:

```
client
dev tun
proto udp
remote your_server_ip 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client.crt
key client.key
comp-lzo
verb 3
```

Key Parameters:

- `client`: Specifies this is a client configuration.
- `remote`: Server IP address and port.
- `ca`, `cert`, `key`: Paths to client certificates and keys.

Starting and Stopping OpenVPN

Start OpenVPN server:	
	```bash
	systemctl start openvpn@server
	```

Stop OpenVPN server:	
	```bash
	systemctl stop openvpn@server
	```

Check OpenVPN status:	
	```bash
	systemctl status openvpn@server
	```

WireGuard Configuration

Server Configuration

WireGuard configurations are typically located in `/etc/wireguard/wg0.conf` (for the `wg0` interface). Example server configuration:

```
[Interface]
PrivateKey = <server_private_key>
Address = 10.6.0.1/24
ListenPort = 51820
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

[Peer]
PublicKey = <client_public_key>
AllowedIPs = 10.6.0.2/32
```

Key Parameters:

- `PrivateKey`: Server's private key.
- `Address`: Server's IP address in the VPN subnet.
- `ListenPort`: Port WireGuard listens on.
- `PostUp`, `PostDown`: Commands to execute when the interface is brought up or down (e.g., iptables rules).
- `PublicKey`: Client's public key.
- `AllowedIPs`: IP addresses allowed for the client.

Client Configuration

Example client configuration:

```
[Interface]
PrivateKey = <client_private_key>
Address = 10.6.0.2/32
DNS = 8.8.8.8, 8.8.4.4

[Peer]
PublicKey = <server_public_key>
Endpoint = your_server_ip:51820
AllowedIPs = 0.0.0.0/0
PersistentKeepalive = 25
```

Key Parameters:

- `PrivateKey`: Client's private key.
- `Address`: Client's IP address in the VPN subnet.
- `DNS`: DNS servers to use.
- `PublicKey`: Server's public key.
- `Endpoint`: Server IP address and port.
- `AllowedIPs`: IP ranges allowed via the VPN.
- `PersistentKeepalive`: Interval to send keepalive packets.

Starting and Stopping WireGuard

Bring up WireGuard interface:	
	```bash
	wg-quick up wg0
	```

Bring down WireGuard interface:	
	```bash
	wg-quick down wg0
	```

Check WireGuard status:	
	```bash
	wg show wg0
	```

IPsec Configuration

IKEv2 Configuration (Strongswan)

Strongswan is a popular IPsec implementation. Configuration files are usually located in `/etc/ipsec.conf` and `/etc/ipsec.secrets`.

`/etc/ipsec.conf` example:

```
config setup
    charondebug="ike 1, knl 1, cfg 0"

conn ikev2-vpn
    auto=add
    keyexchange=ikev2
    ike=chacha20poly1305-sha512-curve25519!
    esp=chacha20poly1305-sha512!
    dpdaction=clear
    rekey=no
    left=%any
    leftid=@your_server_id
    leftcert=server.pem
    leftsubnet=0.0.0.0/0
    right=%any
    rightid=%any
    rightauth=eap-mschapv2
    rightsourceip=10.1.0.0/24
    eap_identity=%identity
```

`/etc/ipsec.secrets` example:

```
: RSA serverKey.pem
username : EAP "password"
```

Key Parameters:

- `left`: Local endpoint configuration (server).
- `right`: Remote endpoint configuration (client).
- `leftid`: Server identifier.
- `rightid`: Client identifier.
- `leftcert`: Server certificate.
- `rightauth`: Client authentication method (e.g., EAP-MSCHAPv2).
- `rightsourceip`: IP address pool for clients.

Starting and Stopping IPsec (Strongswan)

Start IPsec service:	
	```bash
	systemctl start ipsec
	```

Stop IPsec service:	
	```bash
	systemctl stop ipsec
	```

Restart IPsec service:	
	```bash
	systemctl restart ipsec
	```

Check IPsec status:	
	```bash
	ipsec status
	```

Troubleshooting VPN Connections

Common Issues and Solutions

Connectivity Problems:

- **Firewall Issues:** Ensure VPN traffic is allowed through firewalls (e.g., ports 1194 for OpenVPN, 51820 for WireGuard, 500/4500 for IPsec).
- **Routing Problems:** Verify that the routing is configured correctly, especially if using VPN for specific subnets.
- **Incorrect IP Addresses:** Double-check server and client IP configurations.

Authentication Failures:

- **Certificate Issues:** Verify that certificates are valid and correctly configured on both the server and client.
- **Incorrect Credentials:** Ensure that usernames and passwords are correct.

DNS Resolution Issues:

- **DNS Configuration:** Ensure that VPN clients are using the correct DNS servers (e.g., pushed via OpenVPN or configured in WireGuard).
- **DNS Leaks:** Test for DNS leaks to ensure that DNS queries are going through the VPN.

Debugging Tools

Ping:	
	```bash
	ping
	```

Traceroute:	
	```bash
	traceroute
	```

Tcpdump:	
	```bash
	tcpdump -i port
	```

Netstat/Ss:	
	```bash
	netstat -tulnp
	ss -tulnp
	```