NAME

charon-cmd - Simple IKE client (IPsec VPN client)

SYNOPSIS

charon-cmd --host *hostname* **--identity** *identity* **[options]**

DESCRIPTION

charon-cmd is a program for setting up IPsec VPN connections using the Internet Key Exchange protocol (IKE) in version 1 and 2. It supports a number of different road-warrior scenarios.

Like the IKE daemon **charon**, **charon-cmd** has to be run as **root** (or more specifically as a user with **CAP_NET_ADMIN** capability).

Of the following options at least *--host* and *--identity* are required. Depending on the selected authentication *profile* credentials also have to be provided with their respective options.

Many of the **charon**-specific configuration options in *strongswan.conf* also apply to **charon-cmd**. For instance, to configure customized logging to **stdout** the following snippet can be used:

```
charon-cmd {
        filelog {
                stdout {
                        default = 1
                        ike = 2
                        cfg = 2
                }
        }
}
```

OPTIONS

--help

Prints usage information and a short summary of the available options.

--version

Prints the strongSwan version.

--debug *level*

Sets the default log level (defaults to 1). *level* is a number between -1 and 4. Refer to *strongswan.conf* for options that allow a more fine-grained configuration of the logging output.

--host *hostname*
DNS name or IP address to connect to.

--identity *identity*
Identity the client uses for the IKE exchange.

--eap-identity *identity*
Identity the client uses for EAP authentication.

--xauth-username *username*
Username the client uses for XAuth authentication.

--remote-identity *identity*
Server identity to expect, defaults to *hostname*.

--cert *path*
Trusted certificate, either for authentication or trust chain validation. To provide more than one certificate multiple **--cert** options can be used.

--rsa *path*
RSA private key to use for authentication (if a password is required, it will be requested on demand). For other key types use **--***priv*.

--priv *path*
Private key to use for authentication (if a password is required, it will be requested on demand).

--p12 *path*
PKCS#12 file with private key and certificates to use for authentication and trust chain validation (if a password is required it will be requested on demand).

--agent[=*socket*]
Use SSH agent for authentication. If *socket* is not specified it is read from the **SSH_AUTH_SOCK** environment variable.

--local-ts *subnet*
Additional traffic selector to propose for our side, the requested virtual IP address will always be proposed.

--remote-ts *subnet*
Traffic selector to propose for remote side, defaults to 0.0.0.0/0.

--ike-proposal *proposal*

IKE proposal to offer instead of default. For IKEv1, a single proposal consists of one encryption algorithm, an integrity/PRF algorithm and a DH group. IKEv2 can propose multiple algorithms of the same kind. To specify multiple proposals, repeat the option.

--esp-proposal *proposal*

ESP proposal to offer instead of default. For IKEv1, a single proposal consists of one encryption algorithm, an integrity algorithm and an optional DH group for Perfect Forward Secrecy rekeying. IKEv2 can propose multiple algorithms of the same kind. To specify multiple proposals, repeat the option.

--ah-proposal *proposal*

AH proposal to offer instead of ESP. For IKEv1, a single proposal consists of an integrity algorithm and an optional DH group for Perfect Forward Secrecy rekeying. IKEv2 can propose multiple algorithms of the same kind. To specify multiple proposals, repeat the option.

--profile *name*

Authentication profile to use, the list of supported profiles can be found in the **Authentication Profiles** sections below. Defaults to **ikev2-pub** if a private key was supplied, and to **ikev2-eap** otherwise.

IKEv2 Authentication Profiles

ikev2-pub

IKEv2 with public key client and server authentication

ikev2-eap

IKEv2 with EAP client authentication and public key server authentication

ikev2-pub-eap

IKEv2 with public key and EAP client authentication (RFC 4739) and public key server authentication

IKEv1 Authentication Profiles

The following authentication profiles use either Main Mode or Aggressive Mode, the latter is denoted with a **-am** suffix.

ikev1-pub, **ikev1-pub-am**

IKEv1 with public key client and server authentication

ikev1-xauth, **ikev1-xauth-am**

IKEv1 with public key client and server authentication, followed by client XAuth authentication

ikev1-xauth-psk, **ikev1-xauth-psk-am**

IKEv1 with pre-shared key (PSK) client and server authentication, followed by client XAuth authentication (INSECURE!)

ikev1-hybrid, **ikev1-hybrid-am**

IKEv1 with public key server authentication only, followed by client XAuth authentication

SEE ALSO

strongswan.conf(5), **ipsec**(8)