NAME

gnutls-cli-debug - GnuTLS debug client

SYNOPSIS

gnutls-cli-debug [**-flags**] [**-flag** [*value*]] [**--option-name**[[=|]*value*]] [hostname]

Operands and options may be intermixed. They will be reordered.

DESCRIPTION

TLS debug client. It sets up multiple TLS connections to a server and queries its capabilities. It was created to assist in debugging GnuTLS, but it might be useful to extract a TLS server's capabilities. It connects to a TLS server, performs tests and print the server's capabilities. If called with the '-V' parameter more checks will be performed. Can be used to check for servers with special needs or bugs.

OPTIONS

-d *num*, **--debug**=*num*

Enable debugging. This option takes an integer number as its argument. The value of *num* is constrained to being:

in the range 0 through 9999

Specifies the debug level.

-V, **--verbose**

More verbose output.

-p *num*, **--port**=*num*

The port to connect to. This option takes an integer number as its argument. The value of *num* is constrained to being:

in the range 0 through 65536

--app-proto

This is an alias for the *--starttls-proto* option.

--starttls-proto=*str*

The application protocol to be used to obtain the server's certificate (https, ftp, smtp, imap, ldap, xmpp, lmtp, pop3, nntp, sieve, postgres).

Specify the application layer protocol for STARTTLS. If the protocol is supported, gnutls-cli will proceed to the TLS negotiation.

-v *arg*, **--version**=*arg*
Output version of program and exit. The default mode is ‘v’, a simple version. The ‘c’ mode will print copyright information and ‘n’ will print the full copyright notice.

-h, **--help**
Display usage information and exit.

-!, **--more-help**
Pass the extended usage information through a pager.

EXAMPLES

```
$ gnutls-cli-debug localhost
GnuTLS debug client 3.5.0
Checking localhost:443
                    for SSL 3.0 (RFC6101) support... yes
                whether we need to disable TLS 1.2... no
                whether we need to disable TLS 1.1... no
                whether we need to disable TLS 1.0... no
                whether %NO_EXTENSIONS is required... no
                    whether %COMPAT is required... no
                    for TLS 1.0 (RFC2246) support... yes
                    for TLS 1.1 (RFC4346) support... yes
                    for TLS 1.2 (RFC5246) support... yes
                        fallback from TLS 1.6 to... TLS1.2
                for RFC7507 inappropriate fallback... yes
                        for HTTPS server name... Local
                    for certificate chain order... sorted
            for safe renegotiation (RFC5746) support... yes
             for Safe renegotiation support (SCSV)... no
             for encrypt-then-MAC (RFC7366) support... no
            for ext master secret (RFC7627) support... no
                for heartbeat (RFC6520) support... no
             for version rollback bug in RSA PMS... dunno
            for version rollback bug in Client Hello... no
        whether the server ignores the RSA PMS version... yes
```

```
whether small records (512 bytes) are tolerated on handshake... yes
whether cipher suites not in SSL 3.0 spec are accepted... yes
whether a bogus TLS record version in the client hello is accepted... yes
whether the server understands TLS closure alerts... partially
whether the server supports session resumption... yes
for anonymous authentication support... no
for ephemeral Diffie-Hellman support... no
for ephemeral EC Diffie-Hellman support... yes
ephemeral EC Diffie-Hellman group info... SECP256R1
for AES-128-GCM cipher (RFC5288) support... yes
for AES-128-CCM cipher (RFC6655) support... no
for AES-128-CCM-8 cipher (RFC6655) support... no
for AES-128-CBC cipher (RFC3268) support... yes
for CAMELLIA-128-GCM cipher (RFC6367) support... no
for CAMELLIA-128-CBC cipher (RFC5932) support... no
for 3DES-CBC cipher (RFC2246) support... yes
for ARCFOUR 128 cipher (RFC2246) support... yes
for MD5 MAC support... yes
for SHA1 MAC support... yes
for SHA256 MAC support... yes
for ZLIB compression support... no
for max record size (RFC6066) support... no
for OCSP status response (RFC6066) support... no
for OpenPGP authentication (RFC6091) support... no
```

You could also use the client to debug services with starttls capability.

```
$ gnutls-cli-debug --starttls-proto smtp --port 25 localhost
```

EXIT STATUS

One of the following exit values will be returned:

0 (EXIT_SUCCESS)
: Successful program execution.

1 (EXIT_FAILURE)
: The operation failed or the command syntax was not valid.

SEE ALSO

gnutls-cli(1), gnutls-serv(1)

AUTHORS

COPYRIGHT

BUGS

Please send bug reports to: bugs@gnutls.org