NAME

openssl−s_time, s_time − SSL/TLS performance timing program

SYNOPSIS

openssl s_time [**−help**] [**−connect host:port**] [**−www page**] [**−cert filename**] [**−key filename**] [**−CApath directory**] [**−CAfile filename**] [**−no−CAfile**] [**−no−CApath**] [**−reuse**] [**−new**] [**−verify depth**] [**−nameopt option**] [**−time seconds**] [**−ssl3**] [**−bugs**] [**−cipher cipherlist**] [**−ciphersuites val**]

DESCRIPTION

The **s_time** command implements a generic SSL/TLS client which connects to a remote host using SSL/TLS. It can request a page from the server and includes the time to transfer the payload data in its timing measurements. It measures the number of connections within a given timeframe, the amount of data transferred (if any), and calculates the average time spent for one connection.

OPTIONS

−help
Print out a usage message.

−connect host:port
This specifies the host and optional port to connect to.

−www page
This specifies the page to GET from the server. A value of '/' gets the index.htm[l] page. If this parameter is not specified, then **s_time** will only perform the handshake to establish SSL connections but not transfer any payload data.

−cert certname
The certificate to use, if one is requested by the server. The default is not to use a certificate. The file is in PEM format.

−key keyfile
The private key to use. If not specified then the certificate file will be used. The file is in PEM format.

−verify depth
The verify depth to use. This specifies the maximum length of the server certificate chain and turns on server certificate verification. Currently the verify operation continues after errors so all the problems with a certificate chain can be seen. As a side effect the connection will never fail due to a server certificate verify failure.

−nameopt option
Option which determines how the subject or issuer names are displayed. The **option** argument can be a single option or multiple options separated by commas. Alternatively the **−nameopt** switch may be used more than once to set multiple options. See the **x509** (1) manual page for details.

−CApath directory
The directory to use for server certificate verification. This directory must be in ''hash format'', see **verify** for more information. These are also used when building the client certificate chain.

−CAfile file
A file containing trusted certificates to use during server authentication and to use when attempting to build the client certificate chain.

−no−CAfile
Do not load the trusted CA certificates from the default file location

−no−CApath
Do not load the trusted CA certificates from the default directory location

−new
Performs the timing test using a new session ID for each connection. If neither **−new** nor **−reuse** are specified, they are both on by default and executed in sequence.

−reuse
Performs the timing test using the same session ID; this can be used as a test that session caching is working. If neither **−new** nor **−reuse** are specified, they are both on by default and executed in sequence.

−ssl3
This option disables the use of SSL version 3. By default the initial handshake uses a method which should be compatible with all servers and permit them to use SSL v3 or TLS as appropriate.

The timing program is not as rich in options to turn protocols on and off as the **s_client** (1) program and may not connect to all servers. Unfortunately there are a lot of ancient and broken servers in use which cannot handle this technique and will fail to connect. Some servers only work if TLS is turned off with the **−ssl3** option.

Note that this option may not be available, depending on how OpenSSL was built.

−bugs
There are several known bugs in SSL and TLS implementations. Adding this option enables various workarounds.

−cipher cipherlist
This allows the TLSv1.2 and below cipher list sent by the client to be modified. This list will be combined with any TLSv1.3 ciphersuites that have been configured. Although the server determines which cipher suite is used it should take the first supported cipher in the list sent by the client. See **ciphers** (1) for more information.

−ciphersuites val
This allows the TLSv1.3 ciphersuites sent by the client to be modified. This list will be combined with any TLSv1.2 and below ciphersuites that have been configured. Although the server determines which cipher suite is used it should take the first supported cipher in the list sent by the client. See **ciphers** (1) for more information. The format for this list is a simple colon (“:”) separated list of TLSv1.3 ciphersuite names.

−time length
Specifies how long (in seconds) **s_time** should establish connections and optionally transfer payload data from a server. Server and client performance and the link speed determine how many connections **s_time** can establish.

NOTES

s_time can be used to measure the performance of an SSL connection. To connect to an SSL HTTP server and get the default page the command

```
openssl s_time −connect servername:443 −www / −CApath yourdir −CAfile yourfile.p
```

would typically be used (https uses port 443). ’commoncipher’ is a cipher to which both client and server can agree, see the **ciphers** (1) command for details.

If the handshake fails then there are several possible causes, if it is nothing obvious like no client certificate then the **−bugs** and **−ssl3** options can be tried in case it is a buggy server. In particular you should play with these options **before** submitting a bug report to an OpenSSL mailing list.

A frequent problem when attempting to get client certificates working is that a web client complains it has no certificates or gives an empty list to choose from. This is normally because the server is not sending the clients certificate authority in its “acceptable CA list” when it requests a certificate. By using **s_client** (1) the CA list can be viewed and checked. However, some servers only request client authentication after a specific URL is requested. To obtain the list in this case it is necessary to use the **−prexit** option of **s_client** (1) and send an HTTP request for an appropriate page.

If a certificate is specified on the command line using the **−cert** option it will not be used unless the server specifically requests a client certificate. Therefore, merely including a client certificate on the command line is no guarantee that the certificate works.

BUGS

Because this program does not have all the options of the **s_client** (1) program to turn protocols on and off, you may not be able to measure the performance of all protocols with all servers.

The **−verify** option should really exit if the server verification fails.

SEE ALSO

s_client (1), **s_server** (1), **ciphers** (1)

COPYRIGHT