```
SSLeay_add_ssl_algorithms( );
client_method = SSLv2_client_method( );
SSL_load_error_strings( );
ctx = SSL_CTX_new(client_method);
printf("(1) SSL context initialized\n\n");
/*=============================================*/
/* (2) convert server hostname into IP address */
/*=============================================*/

hostname = argv[1];
host_entry = gethostbyname(hostname);
bcopy(host_entry->h_addr, &(ip.s_addr), host_entry->h_length);

printf("(2) '%s' has IP address '%s'\n\n", hostname, inet_ntoa(ip));
/*=================================================*/
/* (3) open a TCP connection to port 443 on server */
/*=================================================*/

sd = socket (AF_INET, SOCK_STREAM, 0);

memset(&server_socket_address, '\0', sizeof(server_socket_address));
server_socket_address.sin_family = AF_INET;
server_socket_address.sin_port = htons(443);
memcpy(&(server_socket_address.sin_addr.s_addr),
       host_entry->h_addr, host_entry->h_length);

err = connect(sd, (struct sockaddr*) &server_socket_address,
              sizeof(server_socket_address));
if (err < 0) { perror("can't connect to server port"); exit(1); }

printf("(3) TCP connection open to host '%s', port %d\n\n",
       hostname, server_socket_address.sin_port);

/*=========================================================*/
/* (4) initiate the SSL handshake over the TCP connection */
/*=========================================================*/

ssl = SSL_new(ctx);      /* create SSL stack endpoint */
SSL_set_fd(ssl, sd);     /* attach SSL stack to socket */
err = SSL_connect(ssl); /* initiate SSL handshake */

printf("(4) SSL endpoint created & handshake completed\n\n");

/*=============================================*/
/* (5) print out the negotiated cipher chosen */
/*=============================================*/

printf("(5) SSL connected with cipher: %s\n\n", SSL_get_cipher(ssl));

/*=========================================*/
/* (6) print out the server's certificate */
/*=========================================*/

server_cert = SSL_get_peer_certificate(ssl);
printf("(6) server's certificate was received:\n\n");
str = X509_NAME_oneline(X509_get_subject_name(server_cert), 0, 0);
printf("      subject: %s\n", str);
```