Carlos Li Hu

Memoria Práctica 3 de Redes 2

Introducción: una descripción de lo que se pretende realizar en la práctica

En esta práctica vamos a incorporar a nuestro servidor y cliente anteriores la capa de seguridad SSL, de forma que nuestras comunicaciones sean a partir de ahora seguras.
Para ello vamos a tener que crear unos certificados de autenticación junto a unas claves privadas/públicas, esto junto a la codificación en c de una biblioteca para usar SSL junto con estos certificados.

Diseño: una explicación de los módulos de los que se compone el programa, así como de las decisiones que se han tomado (procesos o hilos, sincronización, etc.)

- **G-2301-05-P3-ssl.c**
  Aquí encontramos la librería de funciones para el manejo de conexiones con el protocolo SSL.
  No hay mucho que comentar teniendo en cuenta que viene detallado en el enunciado los pasos que debe seguir cada una de las funciones para cumplir su funcionalidad.
  Debo añadir que esta función usa los certificados generados (desde el makefile) en la carpeta certs, donde podemos encontrar el concatenado de certificados y claves privadas de cada entidad implicada
- **cliente_echo.c**
  Este main prueba la librería previamente mencionada junto al servidor_echo. En este main nos dedicamos a establecer una conexión segura con un servidor y le enviamos cadenas de texto escritas por terminal e imprimimos lo que recibimos del servidor por pantalla. Sólo acabamos cuando tecleamos un exit por pantalla.
- **servidor_echo.c**
  Este otro main se encarga también de establecer una conexión segura con el cliente, y se dedica exclusivamente a enviar de vuelta lo que acaba de recibir de dicho cliente, todo ello hasta recibir un exit como mensaje.
- **cliente_IRC.c**
  En este fichero tenemos el cliente de la práctica 2, pero con la funcionalidad añadida para que cumpla la funcionalidad especificada en el c3po.
  Es decir, lo que hace, es ejecutarse como el xchat2 si no se le añaden argumentos, y en el caso de que se le añadan los argumentos con los flags adecuados según se especifica en el guión de la práctica, se ejecuta un main simple, que consiste en iniciar una comunicación segura con un servidor SSL, y enviarle una cadena introducida por pantalla.
- **servidor_IRC.c**
  En este fichero tenemos el servidor de la práctica 1 pero con las ligeras modificaciones que permiten que sea capaz de ejecutar el servidor con la capa segura SSL si se le especifica el puerto con “--port”. La forma en que lo hemos hecho ha sido esencialmente, introducir un flag que es una variable global de tipo boolean llamada ‘ssl_active’, que se encuentra a FALSE si se ejecuta el main sin argumentos, y está a TRUE si se le especifica el puerto para SSL. Mediante ese flag, hemos sido capaces de añadir if/else en todo el código, para ver si es necesario inicializar la capa SSL y saber si enviar los mensajes mediante las funciones por defecto send y recv o si debemos usar las funciones de envío seguro implementadas en el fichero G-2301-05-P3-ssl.c

Funcionalidad IRC: a grandes rasgos, qué funciones del protocolo se han implementado

Principalmente hemos implementado la capa segura SSL, con todo lo que ello implica, es decir, la creación de certificados de autenticación y claves privadas. Y su implementación en lenguaje c de cara a añadirlo a nuestras comunicaciones en el servidor y cliente.

Conclusiones técnicas: temas concretos de la asignatura que se han aprendido al realizar la práctica

En esta práctica hemos aprendido más en profundidad y de forma más aplicada sobre cómo se inician comunicaciones seguras en SSL. En concreto, hemos creado nuestros propios certificados de

autenticación y nuestra propia entidad CA para verificarlos, y una vez creados, los hemos usado en el proceso de codificación del handshake entre el servidor y cliente.

Conclusiones personales: a qué se ha dedicado más esfuerzo, comentarios generales

En nuestra opinión, los fundamentos que subyacen a esta práctica son bastante sencillos, el problema está en que para ponerlos a la práctica , a pesar de tener todos los pasos para codificarlo explicados en el enunciado, hemos tenido que recurrir a leer mucha documentación y/o manuales, lo que ha hecho que nos haya llevado mucho tiempo conseguir que funcione todo correctamente. Pero en general, consideramos que entender cómo funciona el protocolo a más bajo nivel ha sido interesante de aprender.