May 21, 2020

Trabajo Práctico Especial 2020/1
Revisión 0

Resumen

Este documento describe el Trabajo Especial de la materia Protocolos de Comunicación para la cursada del primer cuatrimestre del año 2020.

En su ejecución los alumnos DEBEN demostrar habilidad para la programación de aplicaciones cliente/servidor con sockets, la comprensión de estándares de la industria, y la capacidad de diseñar protocolos de aplicación.

Terminología

Las palabras clave "DEBE", "NO DEBE", "OBLIGATORIO", "DEBERÁ", "NO DEBERÁ", "DEBERÍA", "NO DEBERÍA", "RECOMENDADO", "PUEDE" y "OPCIONAL" en este documento serán interpretadas como se describe en el RFC 2119 [RFC2119].

Tabla de Contenidos

1. Requerimientos Funcionales

El objetivo del trabajo es implementar un servidor proxy para el protocolo SOCKSv5[RFC1928].

El servidor DEBE

1. atender a múltiples clientes en forma concurrente y simultánea (al menos 500).

2. soportar autenticación usuario / contraseña [RFC1929].

3. soportar de mínima conexiones salientes a a servicios TCP a direcciones IPv4, IPV6, o utilizando FQDN que resuelvan cualquiera de estos tipos de direcciones.

4. ser robusto en cuanto a las opciones de conexión (si se utiliza un FQDN que resuelve a múltiples direcciones IP y una no está disponible debe intentar con otros).

5. reportar los fallos a los clientes usando toda la potencia del protocolo.

6. implementar mecanismos que permitan recolectar métricas que ayuden a monitorear la operación del sistema.

   A. cantidad de conexiones históricas

   B. cantidad de conexiones concurrentes

   C. cantidad de bytes transferidos

   D. cualquier otra métrica que considere oportuno para el entendimiento del funcionamiento dinámico del sistema

   Las métricas PUEDEN ser volátiles (si se reinicia el servidor las estadísticas pueden perderse).

7. implementar mecanismos que permitan manejar usuarios cambiar la configuración del servidor en tiempo de ejecución sin reiniciar el servidor. Las diferentes implementaciones PUEDEN decidir disponibilizar otros cambios de ejecución en tiempo de ejecución

de otras configuraciones (memoria utilizada en I/O, timeouts, etc).

8. implementar un registro de acceso que permitan a un administrador entender los accesos de cada uno de los usuarios. Pensar en el caso de que llega una queja externa y el administrador debe saber quien fue el que se conectó a cierto sitio web y cuando.

9. monitorear el tráfico y generar un registro de credenciales de acceso (usuarios y passwords) de forma similar a ettercap por lo menos para protocolos HTTP, y POP3.

10. poder realizar las resoluciones de DNS necesarias sobre HTTP [RFC8484]. Debido a que TLS/HTTPS es un tema especifico de otra materia NO DEBE implementarse soporte de TLS/HTTPS pero si de HTTP (es decir que probablemente no funcione con cualquier servidor DoH, pero si contra instalaciones locales).

2. Requerimientos No Funcionales

Adicionalmente, la implementación DEBE

1. Estar escritos en el lenguaje de programación C, específicamente con la variante C11 (ISO/IEC 9899:2011).

2. Utilizar sockets en modo no bloqueante multiplexada.

3. Tener en cuenta todos los aspectos que hagan a la buena performance, escalabilidad y disponibilidad del servidor. Se espera que se maneje de forma eficiente los flujos de información (por ejemplo no cargar en memoria mensajes muy grandes, ser

eficaz y eficiente en el intérprete de mensajes). El informe DEBE contener información sobre las pruebas de stress. Algunas preguntas interesantes a responder son:

* ¿Cual es la máxima cantidad de conexiones simultáneas que soporta?

* ¿Cómo se degrada el throughput?

4. Seguir los lineamientos de IEEE Std 1003.1-2008, 2016 Edition / Base definitions / 12. Utility Conventions [1] a menos que se especifique lo contrario: Esto se refiere a cómo manejar argumentos de línea de comandos, parámetros, etc

5. El protocolo que diseñe para monitoreo y configuración funcionará sobre SCTP. Deberá documentar detalladamente el protocolo e implementar una aplicación cliente.

6. Tanto la aplicación servidor, como la aplicación cliente de configuración/monitoreo DEBERÁN manejar los argumentos de línea de comandos de cierta forma uniforme (por ejemplo -c <puerto> podría especificar el puerto utilizado para el protocolo de configuración/monitoreo). Los detalles de qué parámetros se deben manejar será publicado en otro documento.

7. Si bien las programas son pequeños podrá utilizar librerías o archivos (fragmento de código) desarrollados por terceros siempre que se cumplan los siguientes requisitos:

   A. La librería o fragmento NO DEBE resolver las cuestiones de fondo del Trabajo Práctico.

   B. La librería o fragmento DEBE tener una licencia aprobada por la Open Source Initiative [2].

C. El uso de la librería o fragmento DEBE ser aprobada por la Cátedra.

Para lograr la aprobación un alumno del grupo DEBE publicar una secuencia en el foro de discusión del trabajo práctico. La secuencia DEBE describir todos aquellos datos que permitan identificar a la librería (por ejemplo la versión); su licencia de esta forma justificando porqué es válido su uso; y el propósito de su inclusión. En caso de que sea un fragmento de código debe adjuntarse. Está permitido utilizar código publicado por los docentes durante la cursada actual, siempre que se atribuya correctamente.

8. A veces existirán ambigüedades en las especificaciones o múltiples formas en como se puede resolver o implementar un problema particular. Por ser una materia de ingeniería se espera que los alumnos tomen decisiones de diseño razonables en estos casos. Los alumnos pueden basar sus decisiones en lo que conoce de ante mano de la tarea y en los objetivos enumerados en este documento o demás enunciados. Los docentes pueden darle consejos sobre las ventajas y desventajas de cada decisiones, pero los alumnos son los que en última instancia las toman.

3. Evaluación

La realización del Trabajo Práctico es una actividad grupal. La calificación es de carácter grupal; pero si hay evidencias de que un alumno de un grupo no participó en la elaboración, o éste no puede defender o demostrar su participación, entonces el alumno no podrá aprobar el Trabajo Práctico.

Los grupos tendrán a su disposición las herramientas provistas por el módulo Grupos de Campus ITBA y un repositorio GIT Bitbucket para el código (y otros archivos). Se espera transparencia en el desarrollo del trabajo.

Cada grupo DEBE entregar todo el material necesario para poder reproducir el Trabajo Práctico. Como mínimo DEBE contener:

a. Un informe en formato PDF [RFC3778] o text/plain (con codificación UTF-8) que contenga las siguientes secciones (respetando el orden):

1. Índice

2. Descripción detallada de los protocolos y aplicaciones desarrolladas.

3. Problemas encontrados durante el diseño y la implementación.

4. Limitaciones de la aplicación.

5. Posibles extensiones.

6. Conclusiones.

7. Ejemplos de prueba.

8. Guía de instalación detallada y precisa. No es necesario desarrollar un programa instalador.

9. Instrucciones para la configuración.

10. Ejemplos de configuración y monitoreo.

11. Documento de diseño del proyecto (que ayuden a entender la arquitectura de la aplicación).

b. Códigos fuente y archivos de construcción

c. Un archivo README en la raíz que describa al menos:

A. la ubicación de todos los materiales previamente enumerados

B. el procedimiento necesario para generar una versión ejecutable de las aplicaciones

C. la ubicación de los diferentes artefactos generados

D. cómo se debe ejecutar las diferentes artefactos generados (y sus opciones)

La entrega se realizará por Campus ITBA en la asignación creada para ello con una fecha de entrega. Se DEBE entregar un tarball que sea el producto de clonar el repositorio GIT (por lo tanto el repositorio GIT DEBE contener todos los materiales de entrega), y su historia.

Una vez realizada la entrega los grupos DEBERÁN mostrar el correcto funcionamiento del sistema con casos de prueba provisto por los equipos y provistos ese día por la Cátedra.

Para aprobar el Trabajo Práctico se DEBE cumplir TODAS las siguientes condiciones:

- El material entregado DEBE estar completo (por ejemplo no se puede corregir si falta el informe o alguna clase)

- Se utilizan únicamente las librería permitidas para los usos definidos.

- DEBE ser correcta las cuestiones de entradas/salida no bloqueante. Por ejemplo las lecturas, escrituras y el establecimiento de nuevas conexiones DEBEN ser mediante suscripciones y no bloquearse.

- DEBE ser correcta las cuestiones relacionadas a la lectura/ escrituras parciales.

- Sumar CUATRO puntos de calificación sobre DIEZ puntos posibles.

Se aceptarán entregas tardías entre 0 horas (inclusive) y 24 horas (exclusivo) luego de la fecha límite de entrega, pero la calificación no podrá execeder de CUATRO puntos.

4. Referencias

4.1. Normative References

[RFC1928] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and L. Jones, "SOCKS Protocol Version 5", RFC 1928, March 1996.

[RFC1929] Leech, M., "Username/Password Authentication for SOCKS V5", RFC 1929, DOI 10.17487/RFC1929, March 1996, <https://www.rfc-editor.org/info/rfc1929>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3778] Taft, E., Pravetz, J., Zilles, S., and L. Masinter, "The application/pdf Media Type", RFC 3778, DOI 10.17487/RFC3778, May 2004, <http://www.rfc-editor.org/info/rfc3778>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

4.2. URIs

[1] https://pubs.opengroup.org/onlinepubs/9699919799/nframe.html

[2] https://opensource.org/licenses