Memoria de P5

Filtros de conversión XDR (parte III) y autenticación del cliente

Shunya Zhan

07 de abril del 2025

ÍNDICE

Introducción

El presente proyecto tiene como objetivo implementar un sistema cliente-servidor basado en RPC (Remote Procedure Call) para procesar datos enviados por un cliente y gestionados por un servidor. RPC es una tecnología que permite la comunicación entre programas en una red, facilitando la ejecución de procedimientos en un servidor remoto como si fueran llamadas locales.

En este sistema, se utiliza la autenticación UNIX (AUTH_UNIX) para validar las credenciales del cliente antes de procesar cualquier dato. AUTH_UNIX es un método de autenticación que emplea las credenciales del sistema operativo UNIX, proporcionando una capa adicional de seguridad al verificar la identidad del usuario que realiza la solicitud.

El proyecto no solo se centra en la autenticación, sino también en la gestión y procesamiento de diferentes tipos de datos, incluyendo enteros, flotantes y cadenas de texto. Esto demuestra la versatilidad y eficiencia de RPC en aplicaciones distribuidas, donde la capacidad de manejar diversos tipos de datos es crucial.

En resumen, este proyecto busca ilustrar el uso práctico de RPC en la creación de aplicaciones distribuidas seguras y eficientes, destacando la importancia de la autenticación y el manejo de datos en entornos de red.

Desarrollo

Arquitectura del Sistema

El archivo credenciales.x define el programa RPC, los tipos de datos y las funciones disponibles.

Se utiliza rpcgen para generar los archivos necesarios (credenciales.h, credenciales_clnt.c, credenciales_svc.c, credenciales_xdr.c).

El sistema consta de dos componentes principales:

Cliente

- Envía datos al servidor utilizando llamadas RPC.
- Configura la autenticación UNIX para garantizar la seguridad.
- Procesa las respuestas del servidor y las muestra al usuario.

Servidor

- Recibe las solicitudes del cliente.
- Valida las credenciales del cliente utilizando AUTH_UNIX.
- Procesa los datos enviados según su tipo:
    - Enteros: Multiplica el valor por 2.
    - Flotantes: Multiplica el valor por 10 y lo trunca a entero.
    - Cadenas: Devuelve un valor de error (-1).
- Responde al cliente con los resultados procesados.

Memoria

1. Autenticación UNIX: Inicialmente, el cliente no enviaba credenciales UNIX al servidor, lo que resultaba en errores de autenticación. Configurar authunix_create_default en el cliente.
2. Errores de compilación: Falta de la función main del cliente. Implementar la función main para manejar la ejecución del cliente.
3. Procesamiento de datos: El servidor no maneja correctamente las cadenas. Implementar un caso por defecto que devuelva un valor de error (-1).

Compilación y ejecución

Compilación: Abre una terminal, levantar el docker cliente y servidor y después navega al directorio donde se encuentran los archivos fuente. Luego, ejecuta el siguiente comando para compilar:

- gcc -o servidor -I /usr/include/tirpc credenciales_server.c credenciales_svc.c credenciales_xdr.c -l tirpc
- gcc -o cliente -I /usr/include/tirpc credenciales_client.c credenciales_clnt.c credenciales_xdr.c -l tirpc

Ejecución: En la terminal, ejecuta el servidor con el siguiente comando:

- ./cliente <IP_del_servidor>
- ./servidor

Prueba y resultado

Servidor:

```
rpcuserserver@15e5bddb0ad8:~/p5$ gcc -o servidor -I /usr/include/tirpc credenciales_server.c credenciales_svc.
c credenciales_xdr.c -l tirpc
```

```
rpcuserserver@15e5bddb0ad8:~/p5$ ./servidor
Credenciales tipo UNIX: 1
Cliente autenticado:
  Host: 937a38ebb92c
  UID: 1001
  GID principal: 1001
  GIDs secundarios: 1001
Recibido entero: 42
Credenciales tipo UNIX: 1
Cliente autenticado:
  Host: 937a38ebb92c
  UID: 1001
  GID principal: 1001
  GIDs secundarios: 1001
Recibido flotante: 3.14
Credenciales tipo UNIX: 1
Cliente autenticado:
  Host: 937a38ebb92c
  UID: 1001
  GID principal: 1001
  GIDs secundarios: 1001
Recibido mensaje: Hola, servidor
```

Cliente:

```
rpcuserclient@937a38ebb92c:~/p5$ gcc -o cliente -I /usr/include/tirpc credenciales_client.c credenciales_clnt.c credenci
ales_xdr.c -l tirpc
rpcuserclient@937a38ebb92c:~/p5$ ./cliente 172.17.0.2
Enviando entero: 42
Resultado recibido del servidor (entero): 84
Enviando flotante: 3.14
Resultado recibido del servidor (flotante): 31
Enviando mensaje: Hola, servidor
Resultado recibido del servidor (mensaje): -1
```

Conclusión

Este proyecto ha demostrado cómo implementar un sistema cliente-servidor utilizando RPC con autenticación UNIX, logrando establecer una comunicación segura y eficiente entre ambos componentes. A lo largo del desarrollo, se ha evidenciado la capacidad del sistema para procesar diferentes tipos de datos, como enteros, flotantes y cadenas de texto, adaptándose a las necesidades específicas de cada tipo de información.

La autenticación UNIX (AUTH_UNIX) ha jugado un papel crucial en la validación de las credenciales del cliente, asegurando que solo usuarios autorizados puedan interactuar con el servidor. Este mecanismo de seguridad es fundamental en aplicaciones distribuidas, donde la protección de los datos y la integridad de las comunicaciones son esenciales.

Además, el sistema ha demostrado ser robusto en el manejo de errores de autenticación, proporcionando respuestas adecuadas cuando las credenciales no son válidas. Esta capacidad de gestionar errores de manera efectiva es vital para mantener la estabilidad y seguridad del sistema en entornos de red.

En resumen, este proyecto no solo ha cumplido con los objetivos iniciales de implementar un sistema cliente-servidor con RPC y autenticación UNIX, sino que también ha sentado las bases para el desarrollo de aplicaciones distribuidas más complejas. La arquitectura y los principios implementados aquí pueden ser extendidos y adaptados para satisfacer las demandas de sistemas que requieran autenticación y procesamiento remoto de datos, ofreciendo una solución segura y escalable para diversas aplicaciones en el ámbito de las redes y la computación distribuida.

Anexo

- Código fuente
- Campus Virtual
- Copilot