
Programación con las RPC de Sun Microsystems (y II)



Seguridad en el ONC RPC

Objetivo

Crear un cliente y un servidor que utilicen el esquema de seguridad UNIX de las RPC de Sun.

Servicios a implementar:

```
typedef string cadena<>;
```

```
struct cadenas {  
    cadena primera;  
    cadena segunda;  
};
```

```
program COMPARA {  
    version COMPVERS {  
        void ACABA (void) = 1;  
        bool SONIGUALES (cadenas) = 10;  
    } = 7;  
} = 0xXXXXXXXX;
```

Servicio	Operación
SonIguales	Devuelve cierto si las cadenas son iguales.
Acaba	Desregistra y retira el servicio de memoria



Seguridad en el ONC RPC

Tareas:

- El servicio se puede implementar con `strcmp`.
- El cliente debe generar sus credenciales UNIX (`authunix_create_default`) antes de invocar el servicio.
- El servidor debe comprobar las credenciales antes de ejecutar cualquier servicio. Si las credenciales no son correctas, debe enviar un mensaje al cliente con la función `svcerr_auth()`.
- **El servicio sólo lo podrán ejecutar clientes con el UID del alumno en las máquinas Orion y Sirio.**
- El servidor debe registrar qué clientes han solicitado su servicio (Identificación, máquina e instante de creación de las credenciales). Guardar la información en el fichero `accesos.log` que se puede inicializar en la función `main` del extremo del servidor.
- Crear un cliente *malicioso* que trate de engañar al servidor. Para ello debe crear sus credenciales con una llamada a la función `authunix_create()` con parámetros *equivocados*.



Broadcast RPC

Objetivo:

Realizar invocaciones de servicios a lo largo de un segmento de red.

Tareas:

- Ejecutar una instancia del servidor creado en el ejercicio anterior en las máquinas **orion**, **sirio** y **centauro**. Comprobar con **rpcinfo** y **ps** que los servicios están registrados y en ejecución.
- Crear un cliente que invoque el servicio **SonIguales** utilizando el mecanismo de multidifusión (**client_broadcast**).
- Registrar todas las respuestas recibidas. ¿De qué máquinas se ha obtenido respuesta? ¿Por qué?.



Concurrencia

Objetivo:

Crear un servidor que admita concurrencia en sus servicios.

Tareas:

- Construir un servidor con dos servicios: `dormir` y `total`.
- El servicio `dormir` que recibe un entero como parámetro y no devuelve al cliente ningún resultado. El servicio invoca a la función `sleep(3)` pasándole el entero que recibe del cliente. Además, sumará a la variable global `Ttotal` ese valor entero.
- El servicio `total` no recibe ningún parámetro. Retorna el valor de la variable `Ttotal` al cliente.
- El servidor debe crear un proceso hijo para la ejecución de sus servicios. Todos excepto `acaba`.

