



Internetworking 2020

Remote control Protocolo

Integrantes:

Patricio Toledo

Diego Delgado

Matias Stocki

Federico Lopez

Introducción

Se desarrolló un protocolo el cual permite la ejecución de comandos bash remotamente, esto se pensó con la meta de poder suspender/apagar equipos de manera local o a través de internet.

Fue probado en las distribuciones Arch Linux y Raspbian, y fue programado en C, haciendo uso del protocolo UDP. Las bibliotecas utilizadas tanto del lado del servidor como cliente son:

`stdio stdlib unistd string sys/types sys/socket arpa/inet netinet/in`

Y solo del lado del cliente (las anterior también):

`netdb unistd`

Objetivos

Realizar una aplicación red (Cliente - Servidor) que nos permita poder ejecutar comandos bash tanto como los host de la red local (LAN), como de los host de un red externa (Internet). La aplicación se podrán ejecutar los comandos desde cualquier host sin ningún tipo de protocolo de conexión ssh, etc.

Alcance

El alcance establecido para esta aplicación es la correcta ejecución de los siguiente comando con sus respectivo parámetro si lo requiere.

- Ejecutar comandos shutdown con parámetro 0
- Ejecutar comando reboot

Arquitectura

La aplicación utilizará el protocolo de transporte UDP iterativo, esto nos permite consultas rápidas de una sola solicitud y una sola respuesta, además como UDP está orientando a la no conexión y su estructura de datagrama incorpora suficiente información de direccionamiento en su cabecera, nos permite atender varios clientes conservando la solicitud de cada uno.

Memoria del proyecto

La motivación de este proyecto surgió al conocerla existencia del estándar 'Wake on Lan' el cual permite encender un host que esté conectado por cable ethernet en la red local, al contar con este estándar planteamos la posibilidad de poder apagar dicho host o reiniciarlo dentro de la red local, y en un futuro fuera de ella.

Esto es realmente útil cuando se combina con un smartphone y se puede encender, apagar y suspender desde este mismo con un botón.

Para realizar dicho proyecto utilizamos las herramientas anteriormente descritas, y para el desarrollo del mismo, utilizamos una terminal zsh y el editor de texto neovim.

Para desarrollar el mismo, primero hicimos una conexión muy básica en la cual se enviaba un mensaje estático contenido dentro del código desde el cliente al servidor, luego buscamos documentación sobre C en internet de como poder ejecutar comandos bash, y dimos con la función popen, la cual permite abrir un nuevo proceso y devuelve la salida de este mismo. Una vez

hecho esto gracias al material y las clases de la materia devolver esta salida al cliente resultó sencillo, y dimos por finalizado el proyecto.

Concluimos que haciendo nacer con Systemd nuestro compilado del servidor y con otro host que utilice el compilado del cliente, puede resultar útil para equipos domésticos, que no estén expuestos a la red externa (ya que este protocolo carece de seguridad). Y en cuanto a C, este es un lenguaje robusto, que al ser de bajo nivel, ofrece una performance mucho mayor a otros lenguajes, siendo muy útil cuando se quiere comprender cómo funcionan algunos protocolos que se utilizan por debajo de las aplicaciones que se desarrollan en lenguajes de alto nivel.

Código

Cliente / Servidor

Repositorio GitHub: <https://github.com/redBaron23/remote-command-protocol>