72.07 Protocolos de Comunicación

2do Cuatrimestre 2017

Trabajo Práctico Especial

Grupo 4

Tobías Matorras Bratsche	55376
Eliseo Parodi	56399
Facundo Gonzalez Fernandez	55746
Ariel Debrouvier	55382

1. Índice

2. Protocolos y aplicaciones desarrolladas

2.1. Aplicaciones

Se desarrollaron 3 aplicaciones principales: un servidor proxy para el protocolo POP3, un cliente de configuración del proxy y un programa para filtrar tipos MIME.

2.1.1. Estados de pop3filter

El flujo por cada cliente es el siguiente (en cada paso puede fallar por errores de I/O o parseo):

Se resuelve el dominio del origin server y se conecta al mismo. Luego esperamos el mensaje de bienvenida del server y cuando lo recibimos le mandamos una request CAPA cuya respuesta no se envía al cliente. Esto se hace para saber si el server soporta pipelining o no. Luego se pasa al estado request donde esperamos comandos del cliente. Cuando llega un paquete, extraemos todos los comandos y los encolamos para mandarselos al origin server. Si el server soporta pipelining se los mandamos todos juntos y sino de a uno. Luego pasamos a esperar la respuesta del server. Si el mismo soporta pipelining leemos respuestas hasta vaciar la queue. En el caso de que una respuesta corresponda a un comando RETR y se cumplan ciertas condiciones disparamos una transformación externa. El retorno al flujo del proxy de nuevo depende de sus capacidades. Si hay pipelining y faltan respuestas por procesar volvemos procesar respuestas, si no volvemos a repetir el ciclo leyendo requests del cliente.

2.1.2. Transformaciones externas

Nuestro proxy se debe comunicar tanto con nuestro programa para filtrar tipos MIME como con cualquier otro programa que el administrador quiera ejecutar. Para realizar esto, se utilizan pipes, uno de lectura y otro de escritura entre el proxy y el programa al cual se le desea enviar el mail. Estos pipes se registran en el selector para poder utilizarlos cuando se pueda ejecutar una acción sobre ellos y no bloquear el proxy.

El mail enviado se parsea cuando se envía al programa y cuando se recibe de este para poder cerrar los pipes en el momento indicado. También se tomó en cuenta casos en los que las transformaciones externas puedan fallar, como por ejemplo, que el programa no sea válido, y se envía el mensaje correspondiente.

2.1.3 Pipelining

Soportamos pipelining para los pedidos del cliente como también para los pedidos al servidor. Para realizar esto se encola los pedidos del cliente,

para luego poder procesarlos cuando llegan las respuestas del servidor. El pipelining con el cliente funciona siempre, mientras que por parte del servidor, primero se envía un comando capa para saber si soporta pipelining y luego dependiendo si soporta o no, se envían los comandos de a uno o de a varios.

Si no soporta pipelining, los comandos se desencolan de a uno y luego se envían. En caso contrario se desencolan varios y se envían todos juntos. Cuando se recibe una respuesta se sabe el orden en el que llegan los comandos y a partir de este orden se procesa. Esto es especialmente útil para comandos como USER en el cual tenemos que guardar el nombre de usuario o RETR cuando las transformaciones externas están activas.

2.2. Protocolo

El protocolo de configuración del proxy funciona sobre SCTP y es el encargado de agregar o eliminar tipos MIME en tiempo de ejecución así como de obtener las métricas del proxy.

El mismo es orientado a texto, de una manera similar a POP3. Cuando se establece la conexión, se envía un mensaje de bienvenida al cliente y a continuación se intercambian comandos y respuestas hasta que el cliente cierre la conexión.

Al trabajar sobre SCTP se tiene la ventaja de que los mensajes se envían por mensajes, por lo tanto nuestro protocolo lee por paquetes para facilitar el parseo. Además se limita que el comando enviado debe entrar dentro del paquete.

2.2.1. Comandos

Cada comando consiste de una sola palabra, case-insensitive que puede recibir un argumento. El comando se separa del argumento con un espacio. Las respuestas a los mismos comienzan con un indicador de status, +OK en caso positivo o -ERR en caso contrario.

2.2.1.1. Autorización

Se requiere un usuario y contraseña para comenzar a operar con el resto de los comandos. Los comandos para loguearse son USER y PASS.

USER:
Argumentos: nombre de usuario.
Posibles respuestas: +OK: Welcome

PASS:
Argumentos: contraseña.
Posibles respuestas:
+OK: Logged in

-ERR: Authentication failed. Try again.

2.2.1.2. Comando BAN

Argumentos: un mime type a censurar.

Descripción: se encarga de agregar un mime type and subtype a la lista de censurados.

Posibles respuestas:

+OK: type banned

-ERR: could not ban type

2.2.1.3. Comando UNBAN

Argumentos: un mime type que está censurado y se desea permitir.

Descripción: elimina de la lista de tipos censurados al recibido por argumento.

Posibles respuestas:

+OK: type unbanned

-ERR: could not unban type

2.2.1.4. Comando STATS

Argumentos: ninguno.

Descripción:

Imprime la fecha y las métricas recolectadas por el proxy hasta el momento.

Posibles respuestas:

+OK

2.2.1.5. Comando LIST

Argumentos: ninguno.

Posibles respuestas:

+OK

2.2.1.6. Comando CMD

Argumentos: un nuevo comando para ejecutar en las transformaciones externas.

Descripción: Se encarga de cambiar el comando que ejecuta los filtros.

Posibles respuestas:

+OK

2.2.1.7. Comando EXT

Argumentos: ninguno.

Descripción: activa o desactiva

Posibles respuestas:

+OK: External transformations activated.

+OK: External transformations deactivated.

3. Problemas encontrados.

A lo largo de la etapa de implementación se encontraron los siguientes problemas:

- Manejo de C: dificultades a la hora de programar en C, debido a la poca práctica que tenemos con el lenguaje.
- Poco conocimiento sobre SO.
- Parser de emails: problemas para encontrar los boundaries de un media type.

4. Limitaciones de la aplicación.

4.1. Cambio de puertos e interfaces default

En caso de que se quiera bindear tanto el proxy como el servicio de management a 127.0.0.1, no se podrá acceder al mismo con su equivalente de ipv6 ::1. Lo mismo sucede si se desea realizar el caso inverso.

4.2. Concurrencia y disponibilidad

La aplicación soporta como máximo teórico 1024 clientes.

4.3. Transformaciones externas

Si se hace un RETR de un mensaje grande, como el especificado en el guión, y el programa de transformaciones externas falla mientras lo está procesando. Se debe esperar a que el origin server termine de enviar el mail al proxy.

Si ocurre un error mientras se lee el mail del pipe de lectura de la transformación externas finalizamos el mail, en vez de copiar lo que falta.

5. Posibles extensiones.

- Mejorar autenticación del cliente de management

Para evitar enviar las passwords del cliente de management en texto plano y que las mismas puedan ser interceptadas fácilmente, se podría realizar un intercambio de random hashs entre el server y el cliente, donde este último appendea su password y aplicando por ejemplo sha256sum se envía esto por la red y el server compara con su copia local.

- Pipelining de protocolo de management

Se podría agregar la capability de pipelining sobre los comandos que se envía a través del cliente de managment.

6. Conclusiones.

El desarrollo del presente trabajo práctico implicó un desafío en el cual tuvo que enfocarse en entradas y salidas no bloqueantes, concurrencia y disponibilidad, características que el grupo no se encontraba acostumbrado a implementar.

Quizás el obstáculo más grande fue la poca práctica con el lenguaje de implementación, pero de igual manera, los códigos provistos por la cátedra fueron de vital importancia para la resolución de las consignas.

7. Ejemplos de prueba.

- pop3filter

```
> ./pop3filter localhost
```

- pop3ctl

```
> ./pop3ctl -L 127.0.0.1 -o 9091
```

- stripmime

```
> FILTER_MEDIAS=text/html ./stripmime
```

8. Guía de instalación.

Para obtener binarios ejecutables se necesita la herramienta cmake. Se debe ejecutar los siguientes comandos en el directorio raíz.

```
cmake .
make
```

Esto generará 3 binarios en la raíz del directorio:

- pop3filter: el proxy pop3.
- pop3ctl: el cliente de management.
- stripmime: el programa que censura partes mimes.

En el repositorio se incluye un archivo secret.txt que contiene el usuario y contraseña del servidor de configuración.

9. Instrucciones para la configuración.

Con el protocolo de configuración se puede modificar en runtime los media types censurados, mediante el uso de los comandos BAN “media” y UNBAN “media”. Solo se puede modificar de a un media type por vez.

Por otro lado se puede modificar el comando que se ejecutará cuando se desee filtrar mensajes, utilizando CMD.

En caso de que se desee desactivar o activar las transformaciones externas se puede utilizar EXT.

10. Ejemplos de configuración y monitoreo.

10.1. Censura del tipo image/jpeg

```
> BAN image/jpeg
> +OK: type banned
```

10.2. Quitar la censura sobre el tipo text/plain

```
> UNBAN text/plain
> +OK: type unbanned
```

10.3. Obtener las métricas

```
> STATS
> +OK
> Date: 2017-11-14T06:56:56Z
> Concurrent connections: 1
> Historical Access: 2
> Transferred Bytes: 6573
> Retrieved Messages: 2
```

10.4. Utilizar stripmime como filtro

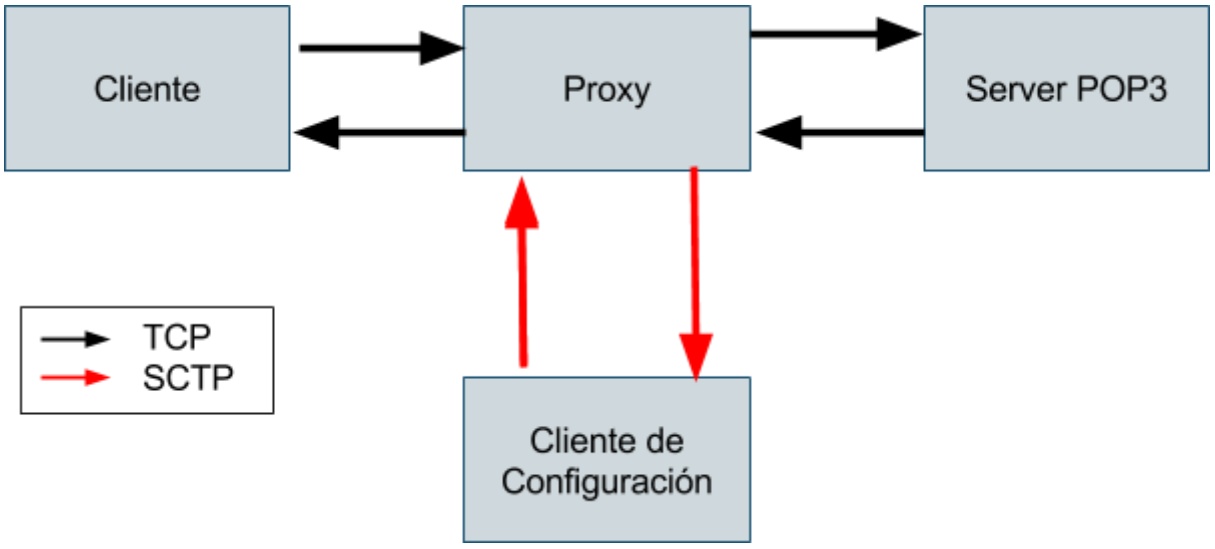
```
> CMD ‘./stripmime’
> +OK
```

10.5. Activar o desactivar las transformaciones externas

```
> EXT
> +OK: External transformations activated.
```

11. Documento de diseño del proyecto.

Arquitectura básica de la aplicación:



A continuación se explica que realizan algunos de los archivos principales:

- POP3filter/src/main.c: punto de entrada del proxy pop3, se encarga de establecer los sockets pasivos TCP y SCTP e inicializar el selector.
- POP3filter/src/selector.c: selector provisto por la cátedra que se encarga de multiplexar la entrada salida.
- POP3filter/src/pop3.c: se encarga del flujo principal de la aplicación.
- POP3filter/src/management.c: se encarga del flujo del protocolo de configuración.
- POP3filter/src/parameters.c: se encarga del parseo y de mantener los comandos.
- POP3ctl/src/pop3ctl.c:cliente que se encarga de establecer la conexión SCTP.
- stripMIME/src/main.c : punto de entrada del programa que ejecuta censura de mimes.