



Trabajo Práctico Especial 2: “Servidor Proxy POP3”

Integrantes:

- Donath, Holger; Legajo 58110
- Fera, Rodrigo; Legajo 58079
- Lombardia, Maximiliano; Legajo 56276
- Monti, Florencia; Legajo 55073
- Rossi, Nicolás; Legajo 53225

Grupo: 10

Año: 2021

Índice:

- Descripción detallada de los protocolos y aplicaciones desarrolladas.....03
- Problemas encontrados durante el diseño y la implementación.....04
- Limitaciones de la aplicación.....05
- Posibles extensiones.....05
- Conclusiones.....05
- Ejemplos de prueba.....05
- Guia de instalación.....06
- Instrucciones para la configuración.....06
- Ejemplos de configuración y monitoreo.....07
- Documento de diseño del proyecto.....08

Descripción detallada de los protocolos y aplicaciones desarrolladas

POP3

Es el protocolo que utiliza el servidor origen al cual se conecta el proxy en una punta. Es un protocolo de texto orientado a conexión y fue diseñado con la intención de recibir mails.

Apenas se inicia la conexión se realiza un saludo inicial (*greeting*) con una respuesta al cliente (+OK) indicando que el mismo puede empezar a enviar mensajes.

Algunos mensajes se pueden enviar sin necesidad de autenticación:

- *CAPA*: Lista las extensiones posibles que soporta el servidor
- *QUIT*: Cierra la conexión

Luego para autenticar se deben mandar estos dos comandos:

- *USER* [usuario]
- *PASS* [contraseña]

Finalmente, estos comandos son los que requieren que se tenga un usuario correctamente autenticado:

- *STAT*: Devuelve la cantidad de mensajes que no han sido borrados y su longitud
- *LIST*: Muestra aquellos mensajes que no han sido borrados, junto con su longitud
- *RETR* [número]: Muestra el mail que coincida con el número otorgado (si no existe el número de mail envía un error)
- *TOP* [número] [líneas]: Muestra la cabecera y el número de líneas requerido del mensaje especificando el número.
- *DELE* [número]: Borra el mensaje especificando el número.
- *RSET*: Recupera aquellos mensajes borrados en la conexión actual
- *UIDL* [número]: Devuelve una cadena de caracteres que funciona como identificador del mensaje el cual se especifica el número. Si no se especifica uno, se devuelve una lista con los números de mensajes y su cadena identificatoria de los mensajes no borrados.

En el caso de éxito de cualquiera de estos comandos, la respuesta del servidor es +OK y la posibilidad de un mensaje adicional indicando la información requerida o dando un mensaje adicional para mejor usabilidad.

Servidor proxy

El servidor proxy es un servidor que funciona como intermediario entre un servidor origen que funciona con el protocolo POP3 y aquellos clientes que se quieran conectar al mismo, con el objetivo de filtrar todo aquel contenido que se desee filtrar.

El filtrado puede indicarse al principio de la ejecución del proxy, o luego mediante el cliente del administrador también se puede configurar.

En primera instancia, se desarrolló una aplicación que pueda de forma transparente comunicar dos clientes conectados con TCP. Sobre esta aplicación, se realizó la conexión POP3 y se procedió a refinar el selector, y a implementar las funciones de escritura/lectura para finalmente agregar la posibilidad de realizar pipelining.

Para la conexión del cliente del administrador, se maneja un thread separado, el cual se crea en el momento que hay una conexión y se cierra en el mismo momento en que el administrador se desconecta.

Los sockets de cada cliente con el servidor origen también usan en todas nuevos threads, esto es para evitar bloqueos en el procesamiento. Este nuevo thread crea el socket, y cuando termina envía una señal al ciclo principal de que ya está disponible el socket para que el cliente le envíe mensajes.

Cliente de monitoreo

Es un cliente básico cuya funcionalidad principal es interpretar los comandos que se le envían e informar distintas configuraciones. Utiliza el protocolo Grupo10 (descrito más abajo) y admite tres opciones para ayuda, configurar puerto y dirección donde escucha el servidor.

Para ingresar a los comandos se debe colocar una contraseña, la cual actualmente es 'password'.

Protocolo de monitoreo (Protocolo Grupo10)

El protocolo de monitoreo Grupo10 sirve para usar las funciones de administrador en el proxy, tales como obtener métricas y configurar el comando de filtrado.

El mismo está pensado para funcionar sobre SCTP, un protocolo de transporte orientado a conexión, similar a TCP.

Para conectarse al cliente se debe contar con la contraseña para ingresar, la cual idealmente está solamente a manos del administrador y se debe conocer de antemano.

Las opciones permitidas son:

- *STATS*: permite ver las estadísticas que se recolectan desde el proxy (conexiones totales, conexiones concurrentes y bytes totales transferidos)
- *HELP*: muestra todos los comandos que permite el cliente del administrador
- *GETCMD*: muestra el comando que se usa para filtrar
- *SETCMD* [comando]: permite configurar el comando que se usa para filtrar
- *LOGOUT*: desconecta la sesión actual

En el caso de un procesamiento exitoso, el protocolo se devuelve como Status Code el símbolo "+" (seguido de una o varias palabras indicando la acción realizada), como por ejemplo "+OK".

En el caso de haber un error, el protocolo devuelve como Status Code el símbolo "-" (seguido de una o varias palabras indicando una mayor explicación sobre el error), como por ejemplo "-ERR".

Este comportamiento está basado en el de POP3, que tiene el mismo formato de respuestas

Problemas encontrados durante el diseño y la implementación

- Se tuvo que pensar una forma de permitir conexiones concurrentes y de forma no bloqueante, con lo cual para eso se utilizó la creación de nuevos threads junto con el selector y sus funciones para realizar la tarea en cuestión.
- Previo al pipelining se tenía organizada la lectura/escritura de una manera, y para implementar esta funcionalidad se procedió a modificar un poco el orden y a modularizar mejor estas cuestiones.

Limitaciones de la aplicación

- Actualmente se usa pselect para la espera de cambios en los sockets. Idealmente, para evitar que esté limitado por el límite de fd_set se podría usar ppoll, pero por cuestiones de tiempo no se logró implementar
- Actualmente se asegura soporte para 1024 conexiones, pero más allá de eso se desconoce un número mayor

Posibles extensiones

- Se podrían armar archivos de log, para posterior consulta y mejor auditabilidad
- Para el cliente del administrador existen algunas cuestiones:
 - Se podría implementar pipelining en el cliente del administrador, tal como ocurre en el cliente tradicional
 - Se podría contemplar la posibilidad de admitir múltiples conexiones y autenticación para más de un usuario
 - Se podría implementar un sistema de encriptación para la contraseña, ya que actualmente se envía en texto plano y puede ser recuperada fácilmente por un atacante
- Finalmente, se puede llegar a aumentar la funcionalidad del filtro para aceptar detección de media types y cambiarlo a otros

Conclusiones

Este trabajo práctico ha servido para aprender en profundidad sobre cuestiones de protocolos de comunicación y programación con sockets. También ha servido para reforzar los conocimientos adquiridos en las materias correlativas, como son Sistemas Operativos y previamente EDA y Arquitectura de Computadoras. La refactorización e integración de nuevas funcionalidades también forzó el pensamiento crítico para lograr integrar todas las cuestiones.

Finalmente, creemos que también sirvió como punto de partida para entender un poco cómo funcionan los sistemas modernos de filtrado de mails, principalmente para evitar correo no deseado.

Ejemplos de prueba

En el pop3filter

```
./pop3filter 127.0.0.1 -P 110
```

```
INFO: proxytcp.c:1460, fstderr    = /dev/null
```

```
INFO: proxytcp.c:1461, local_port= 1110
```

```
INFO: proxytcp.c:1462, origin_port = 110
```

```
INFO: proxytcp.c:1463, mgmt_port    = 9090
```

```
INFO: proxytcp.c:1464, mgmt_addr    = (null)
```

```
INFO: proxytcp.c:1465, pop3_addr    = (null)
```

```
INFO: proxytcp.c:1466, origin_server = 127.0.0.1
```

```
INFO: proxytcp.c:1467, cmd          = (null)
```

```
Listening on TCP port 1110
```

```
Listening on SCTP port 9090
```

```
INFO: proxytcp.c:823, Response from server:
```

```
+OK Dovecot (Ubuntu) ready.
```

```
INFO: proxytcp.c:919, greeting recieved
```

```
INFO: proxytcp.c:1290, User protosuser tried to login
```

```
INFO: proxytcp.c:823, Response from server:
```

```
+OK Logged in.
```

```
INFO: proxytcp.c:823, Response from server:
```

```
+OK 1 messages:
```

```
1 442
```

```
.
```

Guia de instalación

- En el directorio del proyecto se debe ir a la carpeta src
- Luego se debe ejecutar el comando make; esto genera los ejecutables pop3filter y pop3ctl
- Para ejecutar el pop3filter se debe correr: ./pop3filter [opciones] [dirección donde montar]

- Para ejecutar pop3ctl se debe correr: ./pop3ctl [opciones]
- Para ver todos los parámetros que soporta el pop3filter: ./pop3filter -h
- Para ver todos los comandos que soporta pop3ctl se debe correr el binario y ejecutar el comando HELP

Instrucciones para la configuración

A continuación se mencionan las opciones estilo POSIX que soporta este servidor:

-e <error-file>: Configura el archivo de error al cual se redirige stderr. El valor por defecto es /dev/null.

-h: Imprime la ayuda.

-I <pop3-address>: Configura la dirección donde va a escuchar el proxy. Por defecto escucha en todas las interfaces.

-L <management-address>: Configura la dirección en la cual el administrador va a escuchar. Por defecto usa loopback.

-o <management-port>: Configura el puerto para el administrador. Por defecto es 9090.

-p <local-port>: Configura el puerto TCP para conexiones provenientes de los clientes para conexiones POP3. Por defecto el puerto es 1110.

-P <origin-port>: Configura el puerto TCP de origen para el servidor origen POP3. Por defecto el puerto es 110.

-t <cmd>: Configura el comando externo para filtrar.

-v: Imprime información asociada a la versión del proxy.

Ejemplos de configuración y monitoreo

En el pop3ctl:

```
./pop3ctl
```

```
+Greetings
```

```
Please enter Password to enter.
```

```
Password:
```

```
password
```

```
Welcome Administrator!
```

```
These are all available commmands:
```

```
STATS          Prints useful statistics about the proxy server
```

```
GETCMD         Prints the current filter command
```

SETCMD <cmd> Sets new filter command

LOGOUT Disconnects admin client

HELP Prints this message

STATS

Total connections: 0

Concurrent connections: 0

Total bytes transferred: 0

GETCMD

No command is set

SETCMD cat

New filter command set to cat

GETCMD

cat

(se conecta un cliente en el medio)

STATS

Total connections: 1

Concurrent connections: 1

Total bytes transferred: 29

STATS

Documento de diseño del proyecto

