

Apéndice A

Manual del programador

A.1. Introducción

En este capítulo vamos a dar un repaso general a toda la estructura que compone la aplicación de forma que el usuario se familiarice con ella y sobre todo que sea capaz de orientarse a la hora de realizar modificaciones o ampliaciones en la herramienta.

A.2. Estructura de Maya

La herramienta se encuentra dividida en dos partes bien diferenciadas. Por un lado tenemos la parte del punto de control, donde se ubica la herramienta principal, con la interfaz al usuario y la base de datos y por otra parte encontramos la parte que se ubica en los dispositivos; scripts, procesos (cliente y servidor), etc.

A.2.1. Estructura de la herramienta principal

Como acabamos de mencionar la herramienta principal se instalará en el computador que haga las funciones de punto de control. Por un lado tendremos la interfaz web, implementada en PHP, con todos los ficheros necesarios contenidos en la carpeta *Maya_web*.

El fichero principal que carga la aplicación se llama *index.html* y a continuación vamos a detallar el resto de archivos se encuentran en la carpeta comenzando por los ficheros que manejan las distintas operaciones de la aplicación:

- *activateIP.php*: donde está implementada la funcionalidad de habilitar y deshabilitar un dispositivo.
- *changeparameters.php*: en este fichero se implementa toda la funcionalidad de cambio de parámetros excepto la habilitar/deshabilitar dispositivo.

- *deleteIP.php*: maneja la operación de borrado de un dispositivo de la lista.
- *insertIP.php*: se encarga de la inserción de un elemento de en la lista.
- *insertlist.php*: para la inserción de varios dispositivos cargados desde un fichero.
- *index.php*: el archivo principal de la aplicación.

También encontraremos varios ficheros con extensión *inc* que seran usados por los archivos que acabamos de mencionar con el objetivo de reutilizar código y hacerlo más sencillo de cara al usuario. Estos ficheros son:

- *array_operations.inc*: se incluyen diversas funciones para manejar arrays.
- *dbopen.inc* y *dbclose.inc*: operaciones de abrir y cerrar la base de datos.
- *db_operations.inc*: diversas funciones de consulta a la base de datos.
- *misc.inc*: funciones de diversa utilidad.
- *params.inc*: en este fichero encontraremos los parámetros que el usuario debe configurar para adaptar la aplicación a su computador.

Por último, hay una serie de ficheros que no forman parte de la interfaz web pero que son básicos en el correcto funcionamiento de esta utilidad, como son:

- *clientCP*: el proceso que se encargará de enviar los mensajes UDP para el cambio de parámetros en los dispositivos, es invocado desde el código PHP.
- *rsa.sh*: es el *script* que va a generar el par de claves pública y privada y el HIT del punto de control.
- *num_seq*: este fichero almacenará el último número de secuencia usado en los paquetes que se envía desde el punto de control.

En cuanto a la base de datos (de nombre *Maya*), podemos encontrar dos tablas que nos permiten almacenar toda la información correspondiente para cubrir las necesidades de la herramienta. La estructura y los campos de ambas tablas ya se describió anteriormente, concretamente en los puntos ?? y ??.

A.2.2. Estructura de los ficheros de los dispositivos remotos

Los archivos que deben ser copiados en los puntos de acceso que deseemos incluir en el sistema se encuentran dentro de la carpeta *MAYA*, donde, a su vez, podemos encontrar cuatro subcarpetas:

- *cparam*: en esta carpeta podemos encontrar los *scripts* implementados en comandos *shell* de Linux, que se encargan de realizar los cambios en la configuración. Hay un archivo por cada uno de los parámetros a cambiar.
- *init_scripts*: aquí se encuentran los *scripts* que ejecutará el punto de acceso cuando sea arrancando, también se encuentra el fichero *install.sh* que se encarga de configurar el dispositivo para que ejecute los scripts incluidos en esta carpeta cuando éste arranque.
- *server*: en este directorio se hallan los ficheros relacionados con el aspecto de seguridad (claves pública y privada, HIT, etc...) y los procesos servidor y cliente para el sistema de mensajes UDP.
- *aodvuu*: aquí encontraremos los dos archivos necesarios para el funcionamiento del protocolo de encaminamiento en nuestro dispositivo, el ejecutable (*aodvd*) y el módulo que se carga en memoria (*kaodv.o*).

A.2.3. Archivos fuente

Además de los ficheros del punto de control que se sitúan en la carpeta *Maya_web* y de los de los puntos de acceso que se hallan en *MAYA*, existe otra carpeta en el directorio del proyecto llamada *sources*. Donde encontraremos los fuentes de AODV-UU y de los procesos escritos en lenguaje C para el envío de mensajes UDP. También encontraremos una versión ya compilada, tanto del protocolo de encaminamiento como de los archivos *.c* tanto para arquitectura x86 como para MIPS.

A.3. Ampliación de la herramienta

En esta sección explicaremos los pasos fundamentales a realizar para realizar alguna modificación en la herramienta, por ello se recomienda su lectura a todo usuario que desee implementar alguna nueva funcionalidad o modificar las ya existentes.

A.3.1. Añadir un nuevo parámetro

Quizá esta sea la acción más común que se pueda realizar sobre la aplicación. Añadir un parámetro a la oferta de los ya disponibles para ser configurados es una operación sencilla aunque quizá algo costosa, debido en gran parte a la diferente naturaleza de los parámetros y que implica modificar los procesos servidor y cliente que implementan la funcionalidad de los mensajes UDP para soportar el nuevo parámetro.

Las acciones que deberemos llevar a cabo son las siguientes:

1. En la interfaz de usuario (*index.php*) deberemos, añadir al desplegable donde se muestra la lista de parámetros, el nombre del parámetro que deseamos añadir.

2. Además, deberemos conocer si queremos que nuestro parámetro sea global, en caso afirmativo deberemos invocar a la función *javascript* del mismo documento llamada *check_all_disabled()* desde el atributo *OnClick* del elemento que hemos añadido, que obligará al usuario a seleccionar todos los dispositivos de la lista. En caso de que no sea global deberemos invocar a la función *check_none()* para permitir que el usuario los seleccione o no a su gusto. Por último, en caso de que tengamos que controlar alguna restricción del valor del parámetro introducido, deberemos indicarlo en la función *validate_option()*, dónde crearemos una nueva entrada *if*, con el valor de nuestro parámetro para comprobar si está seleccionado e introduciremos las instrucciones necesarias para controlar las restricciones.
3. A continuación, abriremos el archivo *changeparameters.php*, dónde añadiremos en el *switch* de selección de parámetro, el nombre del que hemos elegido y deberemos indicar el valor para los siguientes parámetros: *\$OpenWRT_param* (nombre del parámetro en el sistema OpenWRT), *\$reboot_necessary* (deberemos asignarle el valor *yes* o *no* dependiendo de si es necesario reiniciar el dispositivo tras realizar el cambio) y *\$global* (valor *yes* o *no* dependiendo de si es un parámetro global o no).
4. Para finalizar los cambios en el código PHP, no dirigiremos a la parte final del archivo mencionado en el paso anterior y, sólo en caso de que sea un parámetro global, deberemos indicar el comando necesario para cambiar ése parámetro en el punto de control mediante línea de comandos, los que ya aparecer implementados servirá de buen ejemplo para el lector.
5. El siguiente cambio lo deberemos realizar en el archivo *listener.c*, dónde deberemos declarar, de la misma forma que esta hecho para los demás parámetros, un array con la ruta dónde ubicaremos el script que se encargará de configurar el parámetro que hemos creado. Esto implicará la recompilación del archivo y su posterior actualización en todos los dispositivos del sistema.
6. En ese mismo archivo, deberemos añadir una sentencia *"strcmp(rec_param, "Nombre_de_nuevo_parámetro")"* en la parte dónde se compara el campo parámetro del mensaje UDP con todas las posibilidades existentes, dónde el nombre del nuevo parámetro es el mismo que le asignamos a la opción del desplegable en el punto 2.
7. Por último, sólo quedaría crear el archivo *.sh* que se encargue de la configuración del parámetro y ubicarlo en la ruta que hemos indicado en el punto 5. En este fichero, debemos incluir la instrucción para actualizar el parámetro en la NVRAM del dispositivo y realizar un *commit* sobre ella. En algunos casos es posible que tengamos que reiniciar la interfaz *Wi-Fi* (*"/sbin/wifi"*) o incluso el dispositivo (*"reboot"*).

A.3.2. Añadir un nuevo campo al mensaje UDP

La modificación de la estructura del paquete, puede ser otra modificación bastante probable en la herramienta debido a nuevas necesidades o ampliaciones del sistema, por ello vamos a enumerar los pasos a seguir:

1. En primer lugar modificaremos el código PHP, para ello nos dirigiremos al fichero *changeparameters.php*, en el primer bucle *while*, dónde ejecutamos la instrucción para enviar un mensaje UDP a los dispositivos, deberemos añadir un nuevo parámetro para el proceso *clientCP*. Lo mismo debe hacerse en el fichero *activateIP.php*.
2. A continuación deberemos modificar los procesos, como observaremos, será importante en la posición que coloquemos el nuevo campo y si va a pertenecer a la parte cifrada o sin cifrar. En primer lugar, abriremos el fichero *client.c*. Si se trata de un campo que viajara en la parte cifrada del paquete, tendremos que añadirlo al string *data.message* de la forma que se hace con el resto de parámetros. En caso de ir en la parte sin cifrar, no tenemos que modificar nada, ya que el *string* que almacena todos los argumentos sin cifrar ya tiene espacio de sobra para albergarlos. Sin embargo, en ambos casos deberemos asegurarnos que el argumento número 7 sigue siendo la dirección de broadcast de la red, en caso contrario se debiera modificar la línea *"he=gethostbyname(argv[6]) == NULL"*, sustituyendo el 6 por el número de argumento dónde se encuentre la dirección broadcast menos 1.
3. En el archivo *clientCP.c*, habrá que realizar en primer lugar los mismos cambios que en el punto anterior. Posteriormente habrá que comprobar que el HIT sigue en la posición en *argv[7]*, de lo contrario habrá que cambiar la línea *"pub_keyfile = fopen(argv[7], "r") == NULL"*. Por último habrá que modificar el *scanf* y el *sprintf*, que usamos para leer y escribir la información respectivamente, con los nuevos argumentos.
4. Por último en el *server.c* deberemos, al igual que en *clientCP.c*, modificar el número de parámetros en el *scanf* cuando recibimos un mensaje y en el *sprintf*, con el cual invocamos al proceso cliente.