10.1.—Archivos de configuración IDA

La mayor parte de opciones por defecto de IDA, son controladas y habilitadas por el contenido de distintos archivos de configuración. La mayor parte de dichos archivos están guardados en el directorio **Archivos de Programa\IDA\cfg**, con la excepción de los archivos de configuración para los **plug-ins**, los cuales se encuentran en **Archivos de Programa\IDA\plugins\plugins.cfg**, el archivo **plugins.cfg** lo trataremos individualmente más adelante. La mayoría de los archivos **cfg** son utilizados por los módulos de procesador y son aplicables sólo cuando se están analizando ciertos tipos de **CPU**. Los tres principales archivos de configuración son **ida.cfg, idagui.cfg** y **idatui.cfg**. Las opciones que se aplican a cualquier version IDA se encuentran en el archivo **ida.cfg**, mientras que los archivos **idagui.cfg** e **idatui.cfg** contienen opciones específicas para las versiones GUI y las versiones modo texto respectivamente. Debido a que las versiones que no son para Windows sólo funcionan en modo consola, las distribuciones de IDA para Linux y OS X no contienen el archivo **idagui.cfg**.

10.1.1.— ida.cfg: archivo principal de configuración

El archivo principal de configuración es el **ida.cfg**. En el momento de iniciar un proceso, este archivo es leído para asignar los tipos de procesador por defecto para varias extensiones de archivos y preparar los parámetros de memoria de IDA. En el momento en que se ha especificado un tipo de procesador, el archivo es leído por segunda vez para proceder con las siguientes opciones de configuración. Como ya hemos dicho las opciones contenidas en **ida.cfg** se aplican a todas las versiones IDA sea la que sea la pantalla de usuario utilizada.

Las opciones generales interesantes en **ida.cfg** son la creación de archivos de respaldo (**CREATE_BACKUPS**) y el nombre del visualizador gráfico externo (**GRAPH_VISUALIZER**), del que hablaremos más adelante.

El archivo **ida.cfg**, contiene un gran número de opciones para controlar el formato de las líneas de desensamblado, las cuales incluyen los valores por defecto de muchas de las opciones accesibles con la acción **Options > General**. Estas incluyen el número de bytes opcodes mostrados (**OPCODE_BYTES**), la profundidad de sangrado de las instrucciones (**INDENTATION**), si el offset al **stack pointer** debe ser mostrado en cada instrucción (**SHOW_SP**) y el máximo número de referencias cruzadas que se mostrarán en una línea de desensamblado (**SHOW_XREFS**). El resto de opciones controlan el formato de desensamblado en modo gráfico.

La opción para especificar la longitud de nombres globales en las ubicaciones del programa, así como a las variables de pila, se encuentra en el **ida.cfg** con el nombre de **MAX_NAMES_LENGTH**. Esta opción por defecto tiene el valor de 15 caracteres y sobrepasar dicho valor nos produce un mensaje de atención cada vez que introduzcamos un nombre con una longitud que sobrepase el límite. Se utiliza esta longitud por defecto debido a que algunos ensambladores no pueden manipular nombres de más de 15 caracteres.

La lista de caracteres permitidos en los nombres creados por el usuario, se controla con las opciones **NameChars**. Por defecto la lista permite caracteres alfanuméricos y cuatro caracteres especiales que son **_$?@**. Si IDA no te permite colocar algún carácter en la creación de un nombre para ubicaciones o variables de pila, tendrás que añadir dichos caracteres en **NameChars**. Por ejemplo, tendremos que modificar la opción NameChars

si queremos hacer aceptable el signo (.) en la creación de nombres. Pero debemos evitar el uso de punto y coma, punto, como y espaciado entre caracteres en los nombres, ya que puede producir confusión, ya que dichos caracteres son considerados delimitadores dentro de las líneas de desensamblado.

Las dos últimas opciones que vale la pena mencionar y que tienen influencia en el análisis de los encabezados C, visto en la parte 7, son. La opción **C_HEADER_PATH**, esta especifica una lista de directorios en los que IDA podrá buscar y resolver las dependencias **#include**. Por defecto es listado el directorio normal de Microsoft Visual Studio. Si utilizas un compilador distinto o tus archivos C no están en una ubicación estandar, deberás considerar editar esta opción con tus valores. La otra opción, es **C_PREDEFINED_MACROS** puede utilizarse para especificar una lista predefinida de macros de reprocesamiento que IDA incorporará mientras analiza un archivo C. Esta opción nos proporciona la facilidad de trabajar con macros definidas en los encabezados de archivo a los cuales no tenemos acceso.

El resto de **ida.cfg** contiene opciones específicas para varios módulos de procesador. La única documentación disponible de estas opciones está en forma de comentarios asociados a cada opción. Las opciones específicas habilitadas para cada procesador se habilitan en la sección **Processor options** del diálogo inicial de carga de un archivo.

El último paso en el procesamiento de ida.cfg es la búsqueda de un archivo llamado **Archivo de Programas\ IDA\cfg\idauser.cfg**. Si existe, este archivo es manipulado como una extensión de **ida.cfg**, y cualquier opción de este archivo sobrescribirá las opciones correspondientes en ida.cfg. Si no te es fácil editar ida.cfg, deberás crear **idauser.cfg** y añadir a éste todas las opciones que quieras sobrescribir. Por otra parte la creación de idauser.cfg te ofrece la facilidad de transferir tus opciones especificadas de una versión a otra. Por ejemplo, con idauser.cfg no necesitarás reeditar ida.cfg cada vez que actualices una copia de IDA. Simplemente copias tu idauser.cfg a tu nueva instalación de IDA actualizada.

10.1.2.— idagui.cfg: configuración de la GUI

Los elementos específicos para configurar la GUI Windows de IDA se localizan en el archivo: **Archivos de Programa\IDA\idagui.cfg**. Este archivo está repartido en tres secciones: Comportamiento por defecto de la GUI, mapeado de atajos de teclado y configuración de extensiones de archivo para el diálogo **File > Open**. En esta parte hablaremos de algunas opciones interesantes. Para el resto de opciones de todo el archivo **idagui.cfg**, leerse los comentarios que acompañan a las opciones en donde se describe su propósito.

Ida nos permite especificar un segundo archivo de ayuda utilizando la opción **HELPFILE**. Cualquier archivo especificado en esta opción, no reemplaza la ayuda principal de IDA. El propósito de esta opción es proporcionar acceso a información suplementaria que se pueda aplicar en situaciones específicas de ingeniería inversa. Cuando tengamos especificada una ayuda suplementaria, el atajo **CTRL-F1** produce la apertura del archivo en cuestión pudiendo buscar un nombre en concreto indicándolo con el cursor, si no se encuentra, podrás utilizar el índice del archivo de ayuda IDA. Como ejemplo, Ida no nos ofrece información de ayuda con respecto a la instrucción mnemotecnia en un desensamblado. Si estamos analizando un binario x86, podríamos tener la referencia de la instrucción con una orden. Por lo tanto si pudiéramos localizar

un archivo de ayuda en donde se nos explicara cada instrucción x86, sólo tendríamos que apretar una tecla atajo y leerlo. El único inconveniente en la utilización de un archivo de ayuda complementario es que tiene que ser del tipo **.hlp**, ya que IDA no soporta los archivos **.chm** como ayuda secundaria.

Una pregunta común respecto a la utilización de IDA es, ¿Cómo puedo "parchear" binarios con IDA? En pocas palabras la respuesta es, no puedes, pero hablaremos de este tema más adelante. Lo que sí podemos hacer con IDA es "parchear" la base de datos modificando instrucciones o datos casi de cualquier forma que nos convenga. Una vez que estudiemos los scripts, comprenderemos que modificar la base de datos no es nada difícil. Pero ¿qué ocurriría si no nos interesa o no queremos aprender el lenguaje de scripts de IDA? Pues nada, IDA proporciona un menú para parchear la base de datos el cual no se muestra por defecto. Esta opción es **DISPLAY_PATCH_SUBMENU** y se utiliza para mostrar dicho menú de parcheado realizando la acción **Edit > Patch Program**. Las opciones que se nos proporcionan en dicho menú las estudiaremos en poco más adelante.

Cuando hayamos estudiado alguna capacidad para realizar scripts en IDA, quizá queramos introducir, de vez en cuando, alguna línea de órdenes como un script para realizar alguna acción que no esté permitida vía orden de menú o atajo. Para acceder a las capacidades de script de IDA, normalmente lo debemos seleccionar desde un menú e interactuar con un diálogo para introducir el script. Pero si utilizamos la opción **DISPLAY_COMMAND_LINE**, esto nos proporciona que IDA nos muestre una línea de entrada de texto debajo de la ventana de mensajes. Para activar o desactivar esta línea de órdenes haremos click derecho en la zona de herramientas principal y seleccionamos.

Esta caja de texto es la "**command line**" de IDA, la cual podemos utilizar para introducir on-line declaraciones **IDC**. Digamos también, que esta entrada de órdenes no nos permite ejecutar órdenes del sistema operativo como si estuviéramos en la **cmd**.

La sección de configuración de atajos, es utilizada para especificar el mapeado de las acciones de IDA utilizando los atajos de teclado. Reasignar atajos es útil en muchos casos, así como incluir nuevas acciones que se realicen a través de atajos, cambiar secuencias por defecto por otras que te sean más fáciles de recordar o cambiar secuencias que entren en conflicto con otras secuencias de nuestro sistema operativo, como ya hemos dicho esto normalmente ocurrirá en la versión consola IDA.

Virtualmente cualquier opción que IDA proporciona a través de elementos de los menús o por los botones de herramientas, están listados en esta sección. Desafortunadamente, los nombres de las órdenes no coinciden con el texto utilizado en el menú de IDA, por lo tanto nos tendremos que esforzar para determinar exactamente cual es la opción de configuración específica a la opción de menú. Por ejemplo **Jump > Jump to Problem**, en **idagui.cfg** es igual a la opción **JumpQ**, esta coincide con su atajo **CTRL-Q.** Además también nos encontramos que muchas órdenes contienen su comentario, pero otras no, con lo cual tendremos que determinar su comportamiento. Un truco que nos puede ayudar a deducir la asociación de un menú con la opción de configuración es buscar la acción en la ayuda de IDA. El resultado de la búsqueda normalmente nos conduce a la descripción de la acción correspondiente al elemento del menú.

Las siguientes líneas representan asignaciones de atajo de teclado en **idagui.cfg**:

```
"Abort"                 =       0               // Abort IDA, don't save changes
"Quit"                  =       "Alt-X"         // Quit to DOS, save changes
```

La primera línea es el atajo para la orden de Ida **Abort**, la cual en este caso no tiene asignado ningún atajo. El valor **0** sin comillas indica que no existe atajo asignado a esta orden. La segunda línea nos muestra el atajo asignado a la orden **Quit**. Los atajos son especificados con una secuencia clave entre comillas. Dentro del archivo idagui.cfg existen numerosas asignaciones de atajos.

La última parte de **idagui.cfg**, describe los tipos de archivo asociados a los tipos de extensiones y especifica cuales de los tipos serán listados en el listado de archivos que se muestra en el diálogo **File > Open**. Existen un gran número de tipo de archivo descritos en el archivo de configuración; sin embargo si trabajamos frecuentemente con algún tipo de archivo que no se nos proporciona, podemos editar el archivo de configuración y añadir en el listado de tipo de archivo el que deseamos. La opción **FILE_EXTENSIONS** describe todas las asociaciones de archivos conocidos por IDA. La siguiente línea es un ejemplo de asociación de tipo de archivo.

```
CLASS_JAVA,   "Java Class Files",                                   "*.cla*;*.cls"
```

La línea contiene tres componentes separados por comas: un nombre de la asociación (**CLASS_JAVA**), una descripción **("Java Class Files")** y una pauta de nombre de archivo **("*.cla*;*.cls")**. En la pauta de nombre de archivo se permiten los asteriscos, y para colocar múltiples pautas hay que separarlas con un punto y coma. Un segundo tipo de asociación, permite que varias asociaciones se agrupen en una categoría. La siguiente línea agrupa a todas las asociaciones cuyos nombres empiecen con **EXE_** dentro de una asociación llamada **EXE**.

```
EXE,            "Executable Files",                                 EXE_*
```

Observemos que en este caso el modelo especificado no está entre comillas. Podríamos incluso definir nuestra propia asociación de la siguiente forma:

```
TUTES_CLS,      "Tutes crackslatinos",                              "*.CLS"
```

Podemos elegir el nombre que queramos para la asociación, mientras no exista; sin embargo, sólo añadiendo la nueva asociación al listado **FILE_EXTENSIONS** no es suficiente para que la asociación aparezca en el diálogo **File > Open**. La opción **DEFAULT_FILE_FILTER** lista los nombres de todas las asociaciones que aparecerán en el diálogo **File > Open**. Por lo tanto para completar el proceso de hacer una nueva asociación, añadiremos **TUTES_CLS** a la lista **DEFAULT_FILE_FILTER**.

De forma similar al archivo **idauser.cfg**, si prefieres no realizar los cambios directamente a idagui.cfg, en la última línea de idagui.cfg contiene una directiva que incluirá un archivo en **Archivos de programa\IDA\cfg\idauserg.cfg**. Con lo cual si no quieres editar idagui.cfg, deberás crear **idauserg.cfg** y añadirle todas las opciones que quieras sobrescribir.

10.1.3.— idatui.cfg: Configuración de la cónsola

Análogamente a **idagui.cfg**, para los usuarios de la versión consola de IDA su archivo de configuración es **idatui.cfg**. Dicho archivo es similar en su esquema y funcionalidad a idagui.cfg. A parte de otras cosas, la especificación de atajos se realiza de la misma

forma que en idagui.cfg. Por lo tanto como los dos archivos son similares, sólo detallaremos las diferencias.

Primera, **DISPLAY_PATCH_SUBMENU y DISPLAY_COMMAND_LINE** no son permitidas como opciones en la versión consola y no se incluyen en **idatui.cfg**. El diálogo **File > Open** utilizado en la versión consola es mucho más simple que el de la versión GUI, por lo tanto todas las órdenes de asociación de archivos no existen en idatui.cfg.

Por otro lado, algunas opciones sólo son permitidas en consola. Por ejemplo, podemos utilizar la opción **NOVICE** para iniciar a IDA en modo principiante, lo cual inhabilita algunas de sus funcionalidades más complejas intentando hacer el aprendizaje de IDA más fácil. Una de las diferencias notables es que en modo principiante no podemos abrir las sub vistas.

Los usuarios de consola, probablemente, utilicen mucho las secuencias de atajos. Para facilitar dichas secuencias, el modo consola de IDA nos proporciona una sintaxis para la definición de macros. Varios ejemplos de macros pueden encontrarse en **idatui.cfg**; sin embargo la ubicación ideal para colocar las macros es crear **idausert.cfg** y colocarlo dentro de la carpeta **cfg**, es equivalente a **idauserg.cfg**. Un ejemplo de macro contenido en idatui.cfg, puede ser el siguiente, en idatui.cfg esta macro se encuentra como un comentario:

```
MACRO   "Alt-H"         // this sample macro jumps to "start" label
{
        "G"
        's' 't' 'a' 'r', 't'
        "Enter"
}
```

Las definiciones de macro se introducen con la palabra clave **MACRO** seguida del atajo **“Alt-H”** asociado con la macro. La secuencia de la macro se especifica entre **{ }** con una secuencia de nombre, cadenas o caracteres. Que pueden representar secuencias de órdenes o atajos. La macro de ejemplo, se activa utilizando **ALT-H**, y abre el diálogo **Jump to Address** utilizando el atajo **G**, introduciendo una etiqueta **start** en el diálogo carácter a carácter, y cerramos el diálogo utilizando la clave **ENTER**. Digamos que no utilizamos las sintaxis **“start”** para introducir dicho nombre de símbolo, porque se hubiera podido tomar como el nombre de un atajo y darnos un error.

Para finalizar diremos, que si IDA encuentra cualquier error en sus archivos de configuración, nos mostrará una advertencia de éste, intentado indicarnos de donde proviene. Mientras no lo resolvamos IDA no se ejecutará.

Performance Bigundill@