# Conociendo IDA de “cabo a rabo”

## 2.--La base de IDA PRO

El mejor desensamblador interactivo profesional conocido es el IDA Pro, o simplemente IDA, es un producto actualmente de Hex-Rays empresa sita en Liège (Belgica). El genio que programó el IDA fue **Ilfak Guilfanov**, conocido como **Ilfak**. IDA inició sus andanzas hace una década como una aplicación de consola de entorno **MS-DOS**, lo cual nos ayudará a comprender la naturaleza de la interfaz de usuario del IDA. Entre otras cosas IDA continua utilizando el estilo de consola interactiva derivada de las versiones originales en DOS.

Su corazón, es un desensamblador de descendencia recursivo; el esfuerzo de IDA siempre ha sido mejorar la lógica, para mejorar el proceso de descendencia recursiva. Para poder solventar la deficiencia más grande de dicho método de desensamblado, IDA emplea muchas técnicas heurísticas para identificar el código adicional el cual no haya sido encontrado en el proceso de descendencia recursiva. Además del proceso de desensamblado, IDA es excelente distinguiendo los **bytes de datos** de los **bytes de código** asimismo para determinar el tipo de los datos, que representan dichos bytes de datos. Aunque el código que nos muestra IDA está en lenguaje ensamblado, una de las metas de IDA es mostrarnos una idea lo más cercana posible del código fuente. IDA hace esto posible anotando la información de los **tipos de datos**, los **nombres de las variables derivadas** y los de la **función**. Estas anotaciones minimizan la cantidad de hexadecimales crudos y aumentan al máximo la cantidad de información simbólica al usuario.

## 2.1-- Instalación de IDA

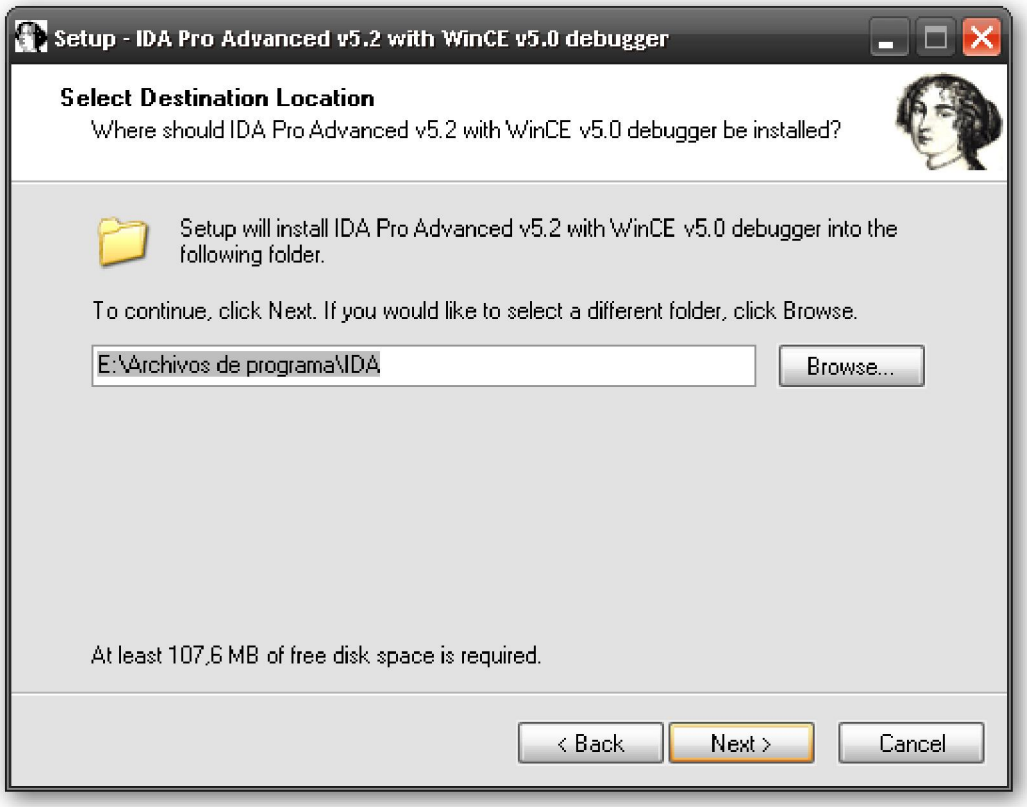
El archivo ejecutable, **ida52.exe**, es el archivo de instalación de Windows. Los archivos comprimidos gzip y tar contienen las instalaciones para OS X y Linux. El **IDA SDK** contiene otras utilidades de las cuales ya hablaremos.

### 2.1.1.-- Instalación en Windows

Ejecutamos el instalador de Windows y nos vamos desplazando por los diálogos colocando lo adecuado a nosotros. En una de las ventanas se te proporciona la oportunidad de instalar el IDA en la localización que desees. Se te da una dirección por defecto la cual utilizaremos en todos los escritos posteriores, directorio **\Archivos de Programa\IDA**. En dicho directorio hallarás el archivo clave**, ida.key**, y también los ejecutables:

** **Idag.exe** (Es la versión Windows GUI de IDA)

** **idaw.exe** (Es la versión Windows modo texto de IDA)




## 2.1.2.-- Instalación OS X y Linux

Para instalarlo en OS X o Linux, haremos gunzip y untar en los archivos apropiados. Instalarlo en Linux puede ser parecido a esto:

**# tar –xvzf linux52adv.gz**

Instalarlo en un sistema OS X, podría ser esto:

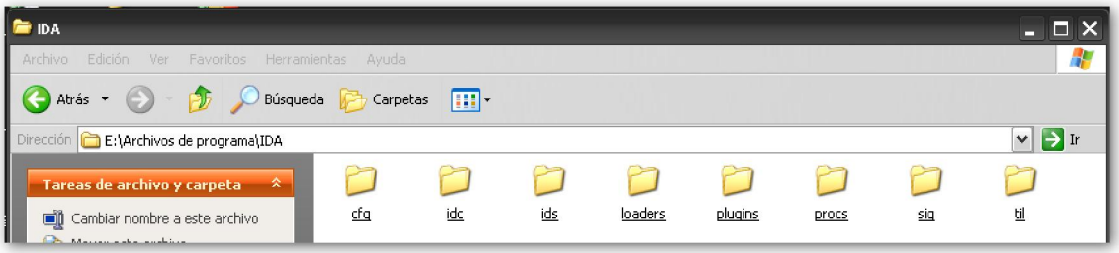
**# tar –xvzf OSX52adv.gz**

En cualquier caso, tendrás un directorio llamado **idaadv** o **idastd**, dependiendo de la versión. Una vez tengamos desempaquetado nuestro IDA, necesitarás copiar tu archivo clave de tu instalación Windows al nuevo directorio. Como alternativa, puedes crear un directorio llamado **$HOME/.idapro** y colocar una copia del archivo clave en este nuevo directorio.

Para ambos entornos OS X y Linux, el nombre del archivo ejecutable es **idal**, si lo ejecutamos nos mostrará una ventana en modo texto, similar a la de **idaw.exe**. La versión Linux depende de la librería **libstdc++.so.5**, tendremos que cerciorarnos de que la tenemos instalada en el sistema.


## 2.2.-- Los directorios de IDA

Antes de empezar a utilizar IDA, primero hemos de familiarizarnos con los contenidos derivados de su instalación. Vamos a darle un vistazo a los distintos directorios. La comprensión de la estructura de los directorios de IDA, se volverá muy importante cuando queramos utilizar sus características avanzadas, las cuales iremos abordando poco a poco. Vamos a realizar una pequeña explicación de cada subdirectorio creado en la instalación:




**Cfg**

El directorio cfg contiene distintos archivos de configuración, el archivo de configuración básica llamado **ida.cfg**, el de configuración del GUI el **idagui.cfg** y el archivo de configuración de la interfaz de usuario en modo texto **idatui.cfg**. A parte de estos existen otras capacidades de configuración, pero de momento tengamos en cuenta estos tres archivos.

**Idc**

El directorio idc contiene los archivos de núcleo necesarios para la programación de scripts en IDA con el lenguaje **IDC**. Realizar scripts con IDC lo explicaremos también más adelante.

**Ids**
El directorio ids contiene los archivos de símbolos, estos describen el contenido de las librerías compartidas que pueden ser referenciadas por los binarios cargados en IDA. Estos archivos IDS contienen un resumen listado de la información de todas las entradas que son exportadas desde una librería concreta. Estas entradas incluyen la información que describe el tipo y número de parámetros que una función pide, el tipo retornado (si existe) de una función y la información respecto a la convención de la llamada utilizada por la función.

**Loaders**
El directorio loaders contiene las extensiones IDA que se utilizarán durante el proceso de cargado del archivo para reconocer y analizar su formato de archivo como **PE** o **ELF**. Los cargadores de IDA se abordarán más adelante.

**Plugins**
El directorio plugins contiene los módulos de IDA diseñados para proporcionar características adicionales a IDA definidos por el usuario. Distintos plugins se explicarán más adelante.

**Procs**
El directorio procs contiene los módulos de procesamiento soportados por la versión instalada de IDA. Los módulos de procesamiento proporcionan la capacidad de traslación de lenguaje máquina a lenguaje ensamblado en IDA y son los responsables de generar el lenguaje ensamblado mostrado en la pantalla de usuario de IDA. También más adelante estudiaremos dichos módulos de procesamiento.

**Sig**
El directorio sig contiene las firmas de código existente, IDA las utiliza para distintas operaciones de comparación de modelos. Es a través de estas comparaciones de modelos por las que IDA identifica secuencias de código en la librería de código, lo que permite el ahorro de mucho tiempo en el proceso de análisis. Las firmas utilizadas son generadas utilizando la característica de IDA, **Fast Library Identification and Recognition Technology (FLIRT)**, la cual también estudiaremos más adelante.

**Til**
El directorio til contiene información del tipo de librería que IDA ha utiliza para registrar el esquema de las estructuras de datos de distintas librerías de compilación. Especificar estas librerías tipo también se estudiarán más adelante.

## 2.3.-- Concepción de la pantalla de usuario de IDA

La herencia de MS-DOS en IDA es evidente. Además de la pantalla, text o GUI, que utilices, IDA utiliza extensivas teclas de atajo. Aunque esto no es malo, pueden producir resultados inesperados puedes pensar que estás en modo de texto y al teclear producirte una acción de una tecla atajo.

Desde la perspectiva, introducir datos, IDA acepta cualquier forma de entrada de diálogo, de manera que si quieres introducir datos a IDA, asegúrate de tener un cuadro de diálogo donde entrar dichos datos.

Para finalizar acuérdate de esto: ¡IDA no permite la acción deshacer! Con lo cual si inadvertidamente pulsas una tecla de acción, no busques deshacer, no existe. Tampoco busques un historial de acciones anteriores, no lo hay.

Si no eres usuario de Windows y quieres utilizar la GUI, tienes dos opciones. Los usuarios de Linux pueden utilizarlo con **WINE** da un buen resultado. La segunda opción es ejecutarlo en una máquina virtual con Windows. Sin embargo si elijes utilizar IDA en depuración local sólo podrás depurar ejecutables Windows.

Performance Bigundill@