IDA PRO – the state-of-the-art binary code analysis tool

IDA Pro is the flagship product of Hex-Rays, the software provider in reverse engineering. Being an interactive and programmable disassembler and debugger, IDA Pro provides excellent quality performance on different platforms and is compatible with many processors. IDA Pro has become the de-facto standard for the analysis of hostile code, vulnerability research and commercial off-the-shelf validation.

IDA Pro comes with different types of licenses: Named, Computer, Floating and Educational license to meet different business’ scales and demands of usage.

Key features

Multi-processor Disassembler

• Disassembler modules for a large number of processors. The free SDK even allows you to run your custom disassembler;
• Full and extensible interactivity;
• Programmable: IDA can be extended in line with user’s own requirement with IDC or IDAPython;
• Open plugin architecture: external plugins enable extension of IDA’s capability;
• FLIRT technology (Fast library identification and recognition technology);
• Code graphing;
• Lumina server holds metadata with a large number of well-known functions;

Multi-target Debugger

• The debugger adds the dynamic analysis of the information collected statically by the disassembler;
• Offers all the features expected from a debugger and more: “remote” function and tracking. Remote debugger: for Windows, Linux, Mac OS X, and other machines in any combination;

More features and upgrades are introduced along with new IDA version releases!
IDA 8.0 Highlights

As usual, IDA has improved on many fronts, but the most important feature this time around - and the one that justifies the major version bump - is the introduction of the brand-new IDA Teams.

IDA Teams is out!

We’ve been hard at work putting the finishing touches to our new product: an integrated set of tools to bring seamless collaboration among teams of reverse-engineers.

Inspired by the best revision control tools out there, IDA Teams brings their key concepts to IDA databases & workflows:

- revision control,
- changes attribution,
- diffing & merging

We’re very proud - and excited! - by what IDA Teams can accomplish, and hope you will appreciate its potential. But we won’t stop there: we already have great ideas for the next versions!

iOS 16 dyld shared cache support

Once again, Apple has modified the internal format of the system files on their mobile operating systems to improve performance and memory usage, so we have updated IDA to support the new tweaks.
Outlined functions

Another new trick used by Apple on recent iOS versions is outlined functions. This is a code size optimization where the compiler extracts common instruction sequences used in several functions into a new pseudo-function which is then “called” from multiple places.

This behavior tends to make decompiled pseudocode ugly, especially when the outlined functions use arbitrary registers or stack variables from the parent function. Now you can mark such functions with a special attribute and the decompiled code will look much nicer.

For example, here’s a fragment of a file which has been compiled with such optimization:

In initial pseudocode, we can see explicit calls to the compiler-generated _OUTLINED_FUNCTION_NN
fragments, and some variables marked in orange because the decompiler considers them to be spoiled by the calls (which is the case with calls to regular functions).

After marking those pseudo-functions with ‘outline’ attribute and refreshing the pseudocode, the decompiler inlines everything and there are no more orange variables:

Golang 1.18

The new version of the language introduced changes to the layout of the metadata. New IDA version can parse it and nicely recover function names.

For example, here’s a stripped golang 1.18 binary in IDA 7.7:
And the same binary in IDA 8.0:

ARC decompiler

ARC processors are used in many embedded devices such as SSD drives or Wi-Fi chipsets. It was also the core used in the infamous Intel ME until several years ago. The new decompiler supports all ARC instruction set variants currently supported by IDA: classic 32-bit ISA, 32/16-bit ARCompact and the new AR Cv2. Delay slots, conditional instructions and zero-overhead loops are supported out of the box.
No more Python 2

It's been over two years since Python 2.7 support has ended, and it's time to let go. IDA 8.0 will support only Python 3.x, including the latest 3.10 (Windows installer includes Python 3.10.5).

Better firmware analysis thanks to the function finder plugin (patfind)

Firmware binaries usually do not have any symbols or other metadata which would help IDA to find code in the unmarked loaded data, so users had to do it manually. In the new release, we've added a plugin which makes use of the pattern format used in Ghidra (with minor extensions). This plugin is enabled by default for binary and binary-like formats and helps IDA discover more code automatically. It can also be invoked manually for normal structured files to find otherwise unreferenced code.

Comparison of the number of functions discovered after initial autoanalysis with default settings:

<table>
<thead>
<tr>
<th>File type</th>
<th>IDA 7.7</th>
<th>IDA 8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortex M0 firmware (8KB)</td>
<td>0</td>
<td>59</td>
</tr>
<tr>
<td>ARM9 firmware (23KB)</td>
<td>0</td>
<td>97</td>
</tr>
<tr>
<td>ARM11 firmware (300KB)</td>
<td>84</td>
<td>1697</td>
</tr>
<tr>
<td>PowerPC firmware (2MB)</td>
<td>0</td>
<td>2223</td>
</tr>
</tbody>
</table>

FLAIR pattern generator (makepat)

The FLAIR toolkit, available to all IDA Pro customers, allows creating FLIRT signatures from static libraries. However, such libraries are not always available - sometimes all you have is just the final linked binary. The
makepat plugin supports creating .pat files for functions in an arbitrary IDA database. It uses the functionality added for Lumina to mark variable bytes in the patterns. The .pat can then be compiled to a signature file using sigmake and applied to other files.

Full changelist: https://www.hex-rays.com/products/ida/news/8_0/
Previous releases

IDA Version 7.7 – Service Pack 1 - Release date: 18th January 2022

Highlights: The Service Pack 1 of IDA 7.7 is primarily a bug fixes release that provides fixes for a few errors that might affect many users.

Full changelis: https://www.hex-rays.com/products/ida/news/7_7sp1/

IDA Version 7.7 – Release date: 24th December 2020

Highlights: IDA 7.7 introduced DA Teams, iOS 16 dyld shared cache support, Outlined functions Golang 1.18, New decompiler: ARC, Better firmware analysis thanks to the function finder plugin (patfind), FLAIR pattern generator (makepat) And much more!

Full changelis: https://www.hex-rays.com/products/ida/news/7_7/

IDA Version 7.6SP1 - Release date: 28th April 2021

Highlights: The Service Pack 1 of IDA 7.6 is primarily a bug fixes release that provides fixes for a few errors that might affect many users.

Full changelis: https://www.hex-rays.com/products/ida/news/7_6sp1/

IDA Version 7.6 - Release date: 22nd March 2021

Highlights: IDA 7.6 introduced Apple Silicon’s full support, Golang analysis, thorough improvements for Hex-Rays Decompiler, two new processor: RISC-V and RL78, further supports for compressed macOS and iOS kernelcache and for Python 3.9

Full changelis: https://www.hex-rays.com/products/ida/news/7_6/

IDA Version 7.5 - Release date: May 2020

Highlights: IDA 7.5 introduced the tree-like folder view that helped with information organization and hence incredibly increase efficiency when it comes to large binaries analysis. MIPS decompiler was added to the lineup, with Lumina now available for both MIPS and PPC binaries. IDA 7.5 comprised many iOS/macOS improvements such as the just-added type libraries with the most major APIs and additional frameworks from macOS and iPhone SDKs.

Full changelis: https://www.hex-rays.com/products/ida/news/7_5/

IDA Version 7.5 – Service Pack 1 - Release date: 19th June 2020

Highlights: This Service Pack was released to improve user experience especially for newly released features such as the tree-like folder view function and the MIPS Decompiler.
Full changelist: https://www.hex-rays.com/products/ida/news/7_5sp1/