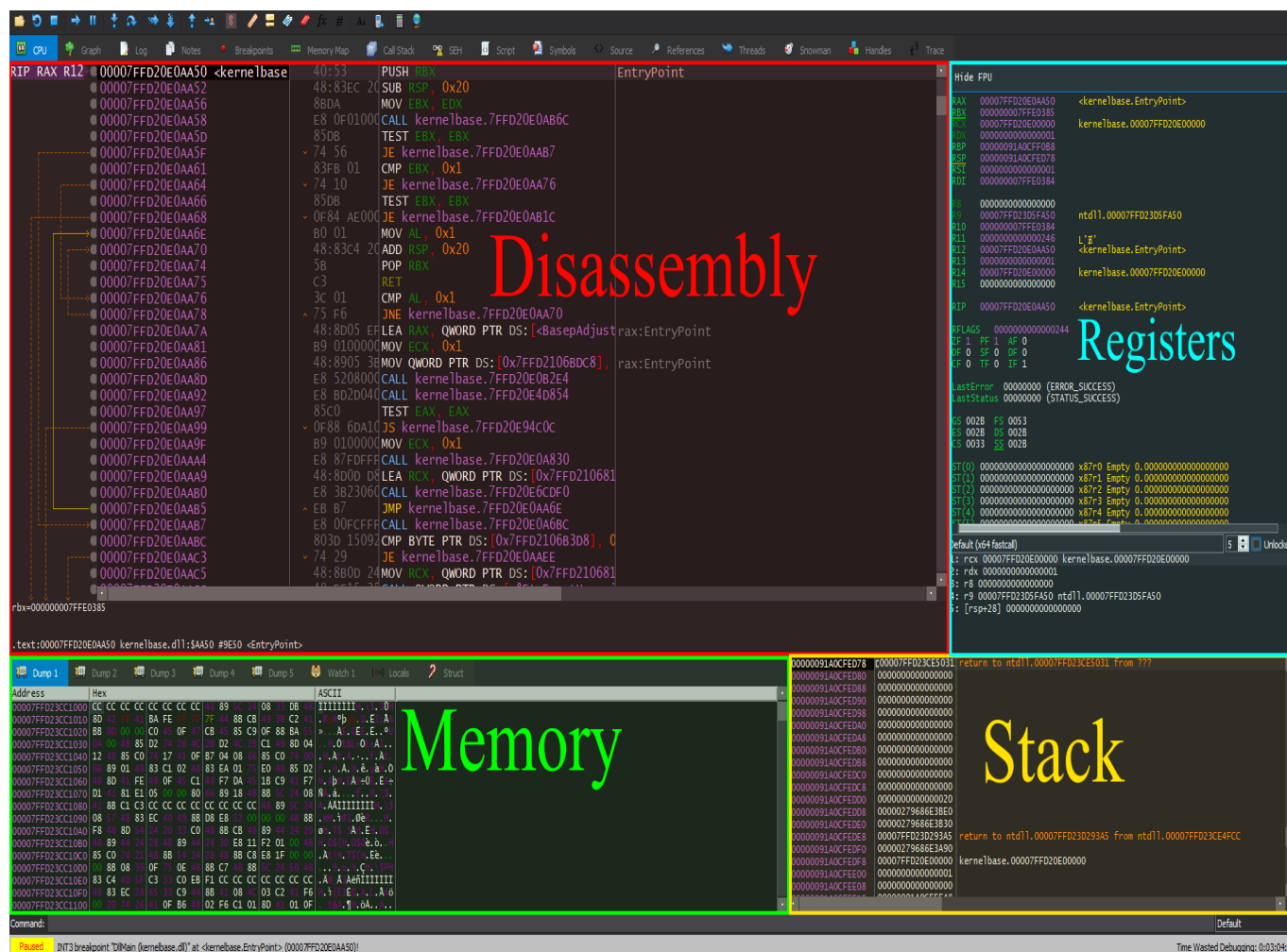# Tool Guides

It's important to know how to use the software you are using. This section will cover some tool basics. It's best to use this section as a reference when the tools first get introduced.

x64dbg

x64dbg has a feature called "Analysis" which will analyze a portion of the disassembly. This analysis will give us more information such as comments on the right with information. There is a plugin that you can use for even better analysis called xAnalyzer. I use it and I would recommend it, although it is not required for this course.

Main Tabs

Near the top of x64dbg, you will see many tabs which will aid us in our reversing.

- **CPU** this is the disassembly.
- The **graph** tab is useful for conditionals. To graph, you have to select the area you want to graph then right click and choose "Graph" or just press "G".
- **Log** is just logging information for x64dbg and can't be ignored for pretty much everything.

- **Notes** allows you to take notes. Global notes will be seen when reversing any binary. Debuggee notes will only be seen when reversing the specific binary they were written for.
- **Breakpoints** shows all set breakpoints.
- **Memory Map** shows sections in memory and where they are located.
- **Call Stack** shows function calls and what's currently on the call stack.
- **SEH** is short for Structured Exception Handler. These are errors that are handled.
- **Script** is for scripting. We will ignore this.
- **Symbols** shows all program symbols, primarily function names.
- **Source** shows the source code if it's available.
- **References** shows any references such as string references. You can populate this tab by going to the CPU tab then right clicking and choosing either "Search for" or "Find references to".
- **Threads** shows currently running threads that have something to do with the process.
- **Snowman** is a decompiler. To populate the decompiler go to the CPU view, select a portion of code, then right-click and choose "Decompile". Snowman isn't a good decompiler, but it's not bad either.
- **Handles** shows all of the programs handles.
- **Trace** will show a trace of a section of the program. Traces allow you to "debug statically" once you have traced a section.

Memory Window

In the memory window, you have several different tabs. You have five different "Dumps" which are areas in memory. There is a "Watch" tab which allows you to watch parts memory and variables that you choose. "Locals" are local variables. "Struct" allows you to analyze structures.

Stack Window

The stack window shows the stack. The address of the current stack pointer is highlighted.

Register Window

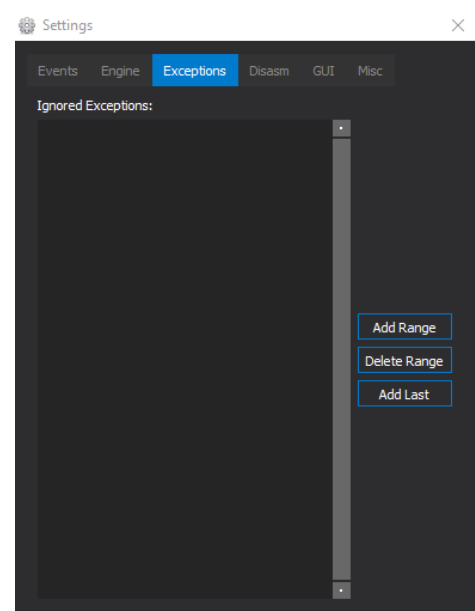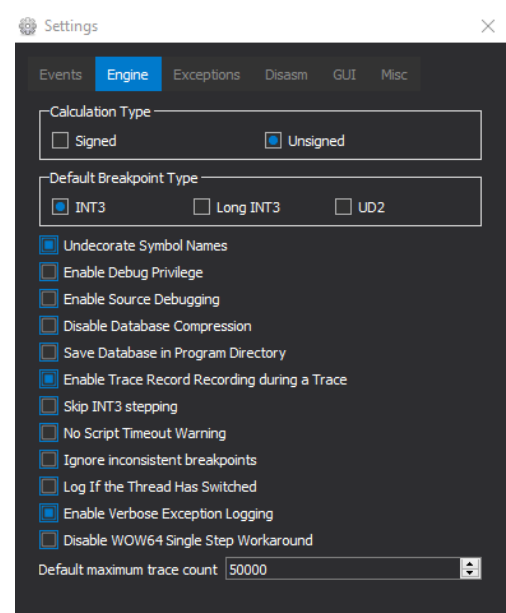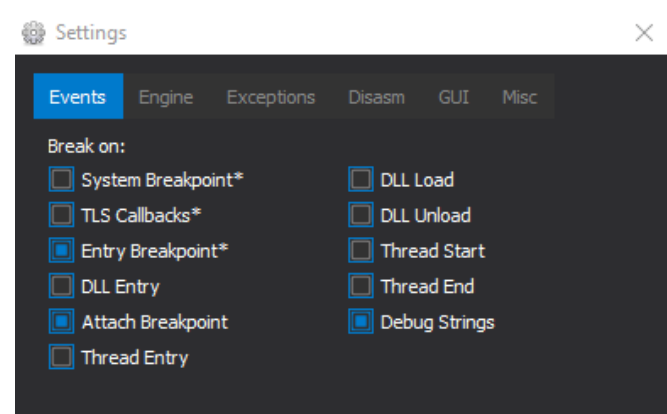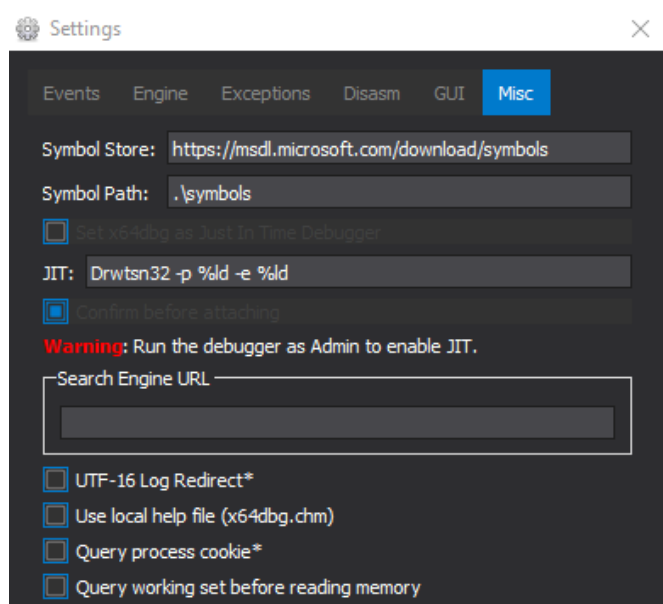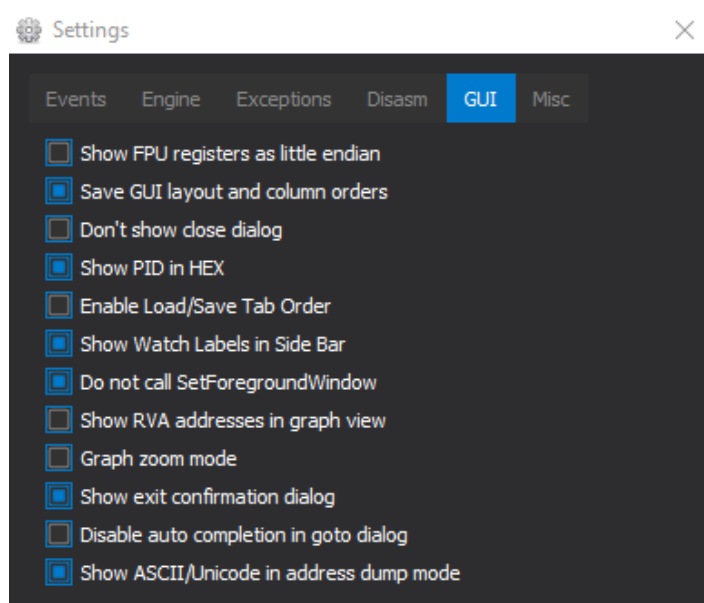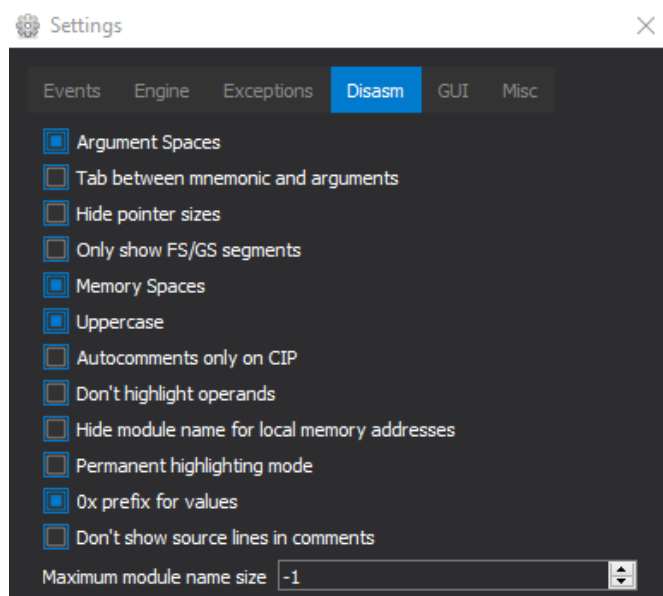Shows all registers. You can right-click the registers and do things with/to them.

Saving and Databases

x64dbg automatically saves your progress. If you want to manually save your progress you can do so with databases. You can import and export databases under the "File" tab. Databases are also useful for sharing between computers.

Settings

Here are my settings:

⚙ Settings   ✕

Events Engine Exceptions Disasm GUI Misc

Break on:
☐ System Breakpoint*   ☐ DLL Load
☐ TLS Callbacks*   ☐ DLL Unload
◉ Entry Breakpoint*   ☐ Thread Start
☐ DLL Entry   ☐ Thread End
◉ Attach Breakpoint   ◉ Debug Strings
☐ Thread Entry

⚙ Settings   ✕

Events **Engine** Exceptions Disasm GUI Misc

┌ Calculation Type ─────────────────────────┐
│ ☐ Signed   ◉ Unsigned   │
└──┘
┌ Default Breakpoint Type ──────────────────┐
│ ◉ INT3   ☐ Long INT3   ☐ UD2   │
└──┘
◉ Undecorate Symbol Names
☐ Enable Debug Privilege
☐ Enable Source Debugging
☐ Disable Database Compression
☐ Save Database in Program Directory
◉ Enable Trace Record Recording during a Trace
☐ Skip INT3 stepping
☐ No Script Timeout Warning
☐ Ignore inconsistent breakpoints
☐ Log If the Thread Has Switched
◉ Enable Verbose Exception Logging
☐ Disable WOW64 Single Step Workaround
Default maximum trace count 50000   ⇕

⚙ Settings   ✕

Events Engine **Exceptions** Disasm GUI Misc

Ignored Exceptions:

                                    Add Range

                                    Delete Range

                                    Add Last

Ghidra - TODO

Ghidra is a good reversing framework made by our friends at the NSA. The main reason why I use Ghidra is for the decompiler which is very good when compared with other compilers.