# IDAPython脚本化软件逆向分析

## 课前准备

1. 学员基础：
   1. 熟悉C/C++
   2. 熟悉x86汇编
   3. 熟悉Python
   4. 具备《**SQLite数据库逆向分析**》
   5. 具备《x86/x64软件逆向分析》更佳
   6. 具备《X86软件逆向分析实战》更佳
2. 演示环境：
   1. Windows 10
   2. Visual Studio Community 2022 /C++桌面开发

<https://visualstudio.microsoft.com/zh-hans/>

* 1. IDA Pro 7.7

<https://hex-rays.com/>

<https://hex-rays.com/ida-pro/>

* 1. WeChat SQLite

<https://pc.weixin.qq.com/>

* 1. PyCharm Community 2022.1

<https://www.jetbrains.com.cn/en-us/pycharm/download/#section=windows>

<https://download.jetbrains.com.cn/python/pycharm-community-2022.1.exe>

* 1. Wing Pro 8

<http://wingware.com/downloads/wingide>

<https://wingware.com/pub/wingpro/8.3.0.1/wingpro-8.3.0.1.exe>

1. 课程源码：[https://github.com/**zmrbak**/IDAPython](https://github.com/zmrbak/IDAPython)
2. 课程类型：演示、实战

## 源码编译常见问题及解决

**升级项目**

右击项目，属性，

配置（所有配置），平台（所有平台），

配置属性，常规，

选择Windows SDK 版本，选择平台工具集，

确定。

**编译**

配置：debug/release

平台: x86

右击项目，生成/重新生成

## MSVC编译控制参数(1)

<https://docs.microsoft.com/zh-cn/cpp/build/projects-and-build-systems-cpp?view=msvc-170>

|  |  |
| --- | --- |
| **Debug** | **Release** |
| **/JMC**  启用本机“仅我的代码”  **/permissive-**  使某些非符合代码可编译(功能集可更改)(默认开启)  **/ifcOutput "Debug\"**  建议使用 /ifcOutput <directory> 为每个编译创建一个单独 .ifc 的文件。  **/GS**  启用安全检查  **/analyze-**  启用本机分析  **/W3**  设置警告等级(默认 n=1)    **/Zc:wchar\_t**  C++ 语言合规性，这里的参数可以是: wchar\_t 是本机类型，不是 typedef  **/ZI**  启用“编辑并继续”调试信息  **/Gm-**  启用最小重新生成  **/Od**  禁用优化(默认)  **/sdl**  支持其他安全功能和警告  **/Fd"Debug\vc143.pdb"**  命名 .PDB 文件  **/Zc:inline**  C++ 语言合规性，这里的参数可以是  如果是 COMDAT，则删除未引用的函数或数据, 或仅使用内部链接(默认关闭)  **/fp:precise**  选择浮点模型, "precise" 浮点模型；结果可预测  **/D "WIN32"**  定义宏  **/D "\_DEBUG"**  **/D "\_CONSOLE"**  **/D "\_UNICODE"**  **/D "UNICODE"**  **/errorReport:prompt**  已弃用。请将内部编译器错误报告给 Microsoft  prompt - 提示立即发送报告  **/WX-**  将警告视为错误  **/Zc:forScope**  C++ 语言合规性，这里的参数可以是:  对范围规则强制使用标准 C++  **/RTC1**  启用快速检查(/RTCsu)  **/Gd**  \_\_cdecl 调用约定  **/Oy-**  启用帧指针省略  **/MDd**  与 MSVCRTD.LIB 调试库链接  **/FC**  诊断中使用完整路径名  **/Fa"Debug\"**  命名程序集列表文件  **/EHsc**  GX[-] 启用 C++ EH (与 /EHsc 相同)  EHs 启用 C++ EH (没有 SEH 异常)  EHc 外部 "C" 默认为 nothrow  **/nologo**  取消显示版权信息  **/Fo"Debug\"**  命名对象文件  **/Fp"Debug\ida03.pch"**  命名预编译头文件  **/diagnostics:column**  控制诊断消息的格式：打印列信息 | **/permissive-**  使某些非符合代码可编译(功能集可更改)(默认开启)  **/ifcOutput "Release\"**  建议使用 /ifcOutput <directory> 为每个编译创建一个单独 .ifc 的文件。  **/GS**  启用安全检查  **/GL**  启用链接时代码生成  **/analyze-**  启用本机分析  **/W3**  设置警告等级(默认 n=1)  **/Gy**  分隔链接器函数  **/Zc:wchar\_t**  C++ 语言合规性，这里的参数可以是: wchar\_t 是本机类型，不是 typedef  **/Zi**  启用调试信息  **/Gm-**  启用最小重新生成  **/O2**  最大优化(优选速度)  **/sdl**  支持其他安全功能和警告  **/Fd"Release\vc143.pdb"**  命名 .PDB 文件  **/Zc:inline**  C++ 语言合规性，这里的参数可以是  如果是 COMDAT，则删除未引用的函数或数据, 或仅使用内部链接(默认关闭)  **/fp:precise**  选择浮点模型, "precise" 浮点模型；结果可预测  **/D "WIN32"**  定义宏  **/D "NDEBUG"**  **/D "\_CONSOLE"**  **/D "\_UNICODE"**  **/D "UNICODE"**  **/errorReport:prompt**  已弃用。请将内部编译器错误报告给 Microsoft  prompt - 提示立即发送报告  **/WX-**  将警告视为错误  **/Zc:forScope**  C++ 语言合规性，这里的参数可以是:  对范围规则强制使用标准 C++  **/Gd**  \_\_cdecl 调用约定  **/Oy-**  启用帧指针省略  **/Oi**  启用内部函数  **/MD**  与 MSVCRT.LIB 链接  **/FC**  诊断中使用完整路径名  **/Fa"Release\"**  命名程序集列表文件  **/EHsc**  GX[-] 启用 C++ EH (与 /EHsc 相同)  EHs 启用 C++ EH (没有 SEH 异常)  EHc 外部 "C" 默认为 nothrow  **/nologo**  取消显示版权信息  **/Fo"Release\"**  命名对象文件  **/Fp"Release\ida03.pch"**  命名预编译头文件  **/diagnostics:column**  控制诊断消息的格式：打印列信息 |

%comspec% /k "C:\Program Files\Microsoft Visual Studio\2022\Preview\Common7\Tools\VsDevCmd.bat"

## MSVC编译控制参数(2)

## IDA Pro中的脚本插件（1）

目录结构

cfg 配置文件

dbgsrv 动态调式所需的server文件

**idc** 内置脚本语言IDC所需的核心文件

ids 描述加载的二进制文件共享库内容

loaders 用于加载识别解析文件格式的扩展文件

platforms 平台相关的窗口库

plugins 扩展插件的文件夹

procs 处理器支持模块，机器码转汇编

python IDAPython库

https://github.com/idapython/src

python38 IDA内置的Python程序

sig 模式匹配的签名

themes IDA外观样式

til 类型库信息

github树状插件

<https://www.octotree.io/>

IDAPython API随机文档

IDA\_Pro\_7.7/python/examples/index.html

IDA\_Pro\_7.7/idahelp.chm

启动IDA时Python插件出错

为PATH添加Python路径 IDA\_Pro\_7.7\python38

## IDA Pro中的脚本插件（2）

## 程序的Debug/Release以及符号文件对IDA的影响（1）

编译器会生成与应用程序同名的PDB文件，作为符号文件，有助于IDA对程序中的函数名进行识别。

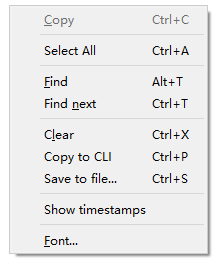
Debug模式中，编译器为程序添加了一些额外的调试代码，有助于对程序的调试。

Release模式中，编译器对程序做了一些优化，一些函数被内联到其它函数之中，未被引用到的函数会在链接阶段被移除。

## 程序的Debug/Release以及符号文件对IDA的影响（2）

## 使用Wing Pro调试IDAPython脚本

Output窗口快捷键



Wing Pro 调试设置

import wingdbstub

import idc

import idautils

wingdbstub.Ensure()

print('Test From Wing Pro 8')

for seg in idautils.Segments():

name = idc.get\_segm\_name(seg)

start = idc.get\_segm\_start(seg)

end = idc.get\_segm\_end(seg)

print(name,start,end)

接受调试连接

开始调试

IDA中，运行脚本

wingdbstub.py

C:\Program Files (x86)\Wing Pro 8

## 使用PyCharm开发IDAPython脚本

运行脚本的方式

Shift+F2 打开脚本编辑窗口，粘贴代码，点击运行。

将代码粘贴到python cli，按两次回车

Alt+F7，选择脚本文件，执行脚本文件

PyCharm配置

创建项目

将IDA\_Pro\_7.7\python包复制到项目中

将IDA\_Pro\_7.7\python包中的3设置为“源 根”

