上一章介绍的是IDA的交叉引用功能，交叉引用反映的是地址之间的关系，但是可能还是不够描绘出我们对整个程序的大局观，因此，我们可以以交叉引用为起点，通过IDA绘图功能描绘出各个方法、基本块（几条指令的组合）和指令之间的关系。

IDA总共提供了两种绘图功能，分别为第三方提供的外部图形功能和IDA内置的图形功能。其中我们常用的通过空格键在反汇编切换出来的图形就是IDA内置的图形功能。

内置图形功能

集成绘图模式提供了另外一种界面，以替代标准的文本式反汇编代码清单。在图形模式中，经过反汇编的函数以类似于外部流程图的控制流图形显示。

由于这种模式使用的是面向函数的控制流图形，因此，它一次只能显示一个函数。而且，图形模式不能用于显示函数以外的指令。

要在文本视图与图形视图之间切换，可以按下空格键，或者右击反汇编窗口，然后在上下文菜单中选择Text View或Graph View。

对于较大的图形，使用GraphOverview（图形概览）窗口进行平移会更加方便。“图形概览”窗口中始终显示有一个虚线矩形框，框中的图形与当前反汇编窗口中显示的内容相对应。你可以随时单击并拖动这个虚线框，以重新定位图形视图。

在图形试图中依然可以双击导航、访问详细栈帧视图等等。同时，在某个基本块中，可以发现左上角有三个按钮，分别可用于更改节点的背景颜色，分配或更改节点名称，以及访问以该节点为目标的交叉引用列表。

更改节点的颜色非常有用，常说的IDA染色法就是通过这个地方进行染色，要给节点分配了颜色，文本模式下对应的指令也会使用这种颜色作为背景色，要取消颜色的分配，右击节点的标题栏，在上下文菜单中选择Set node colorto default（设置默认节点颜色）即可。

外置图形功能

使用View►Graphs（查看►图形）子菜单可以生成5种类型的图形。可在IDA中使用的外部图形包括：

- 函数流程图
- 整个二进制文件的调用图
- 目标符号的交叉引用图
- 源头符号的交叉引用图
- 自定义的交叉引用图

但是外部图形存在着与内置图形很大的区别就是外部图形无法进行交互，通常仅限于缩放和平移。

基本块

在OLLVM反混淆中经常会见到这样几个概念：基本块、真实块、控制块、虚假块。其中基本块是一条或数条指令的组合，它拥有唯一一个指向块起始位置的入口点和唯一一个指向块结束位置的退出点。基本块在行为方面有一个重要的特点，即一旦基本块中的第一条指令开始执行，块中的其他指令都会执行，直到最后一条指令。

函数流程图

将光标放在一个函数中，选择View►Graphs►Flow Chart（热键为F12）,IDA将生成并显示一个外部流程图，或者称为控制流图形，因为它们将一个函数的指令划分成基本块，并使用边来表示块之间的流。这种外部流程图与IDA的集成式反汇编图形视图非常类似。

函数调用图

函数调用图可帮助我们迅速理解程序中函数调用的层次结构。

首先为程序中的每个函数创建一个图形节点，然后再根据函数之间的调用交叉引用将函数节点连接起来，即可生成调用图。为一个函数生成调用图的过程可以看做是一个递归下降过程，即遍历该函数调用的所有函数。

一般来说，只要访问到库函数，就可以停止递归下降过程，因为库函数更底层的实现一般不在我们的逆向范围之内。但在静态链接二进制文件中，可能会由于静态链接二进制文件中包含所有链接到的所有库的代码，所以生成的函数调用库可能非常巨大。

而且几乎所有的编译器都会插入包装代码，用于初始化和终止库，并在将控制权转交给main函数之前正确配置相关参数。

自定义交叉引用图

交叉引用图顾名思义，自定义交叉引用图（View▶Graphs▶User Xrefs Chart）则可以指定地址范围，指定地址范围内的每个全局符号均以节点显示。如果起始地址是数据库中最低的地址，而结束地址是数据库中最高的地址，在这种极端情况下，生成的图形为整个二进制文件的函数调用图。

其次还可以设置递归最大深度和忽略哪些节点例如库函数，-1为不限制。该功能是最强大的外部图形功能。