努力改个网名

这短短的一生，我们最终都会失去；你不妨大胆一些，爱一个人，攀一座山，追一个梦。

昵称： 努力改个网名
园龄： 3年1个月
粉丝： 178
关注： 2
+加关注

< 2020年4月 >

日	一	二	三	四	五	六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

搜索

找找看

积分与排名

积分 - 652679
排名 - 292

随笔分类

Android(8)
Crazy Talk(5)
Eclipse(7)
Hadoop(2)
Java(12)
Kali(15)
Linux(92)
MySQL(12)
Oracle(11)
Pentest(43)
Reverse(7)
Tomcat(2)
Version Control(7)
VMware(5)
Weblogic(17)
WebSphere(9)

博客园 首页 新随笔 联系 订阅 管理
随笔- 439 评论- 112 文章- 7

BinDiff安装使用教程

一、说明

大概一两年前在《漏洞战争:软件漏洞分析精要》听到bindiff（和补丁比较法），但一直都没去使用。前两天再回头看书感觉需要使用一翻，整个过程下来还是遇到了一些问题，值得记录一番。

二、安装

2.1 jdk安装

bindiff是一款java程序，因此需要安装jdk，我装的是jdk1.8其他版本兼容性不太清楚。

jdk下载地址：
https://www.oracle.com/technetwork/java/javase/downloads/index.html

2.2 ida安装

bindiff需要借助ida pro进行分析，所以需要安装ida pro。官方说明需要6.8及以上版本，但7.x版本也尚不支持。

ida pro安装可参考：
https://www.cnblogs.com/lsdb/p/7500981.html

2.3 bindiff安装

下载地址：https://www.zynamics.com/software.html

bindiff同时支持windows、linux、mac；bindiff5只能装在win8和win10上；下载链接不明显但是是有链接的，直接中键点击文件名处即可下载。

随笔档案

BinDiff 5 (latest version)

Filename	Size	SHA256
bindiff_5_amd64.deb bindiff_5_amd64.deb.asc	24.0M	cd859562803f58331fee5f0741e081edd2a470914fa9c60ef3db6fe06f058bc4
BinDiff5.dmg	58.3M	842eb2d0a23f0f889ad19704853159f07834f6169ade537cfa2dd9b7a494bfef
bindiff5.msi	57.3M	76aff21604cbe8ee4c3fd7ee94dede43d20b006af3f33d2c9ea32345ce71f986

The code for the BinExport plugin is available on GitHub.

BinDiff 4.3.0

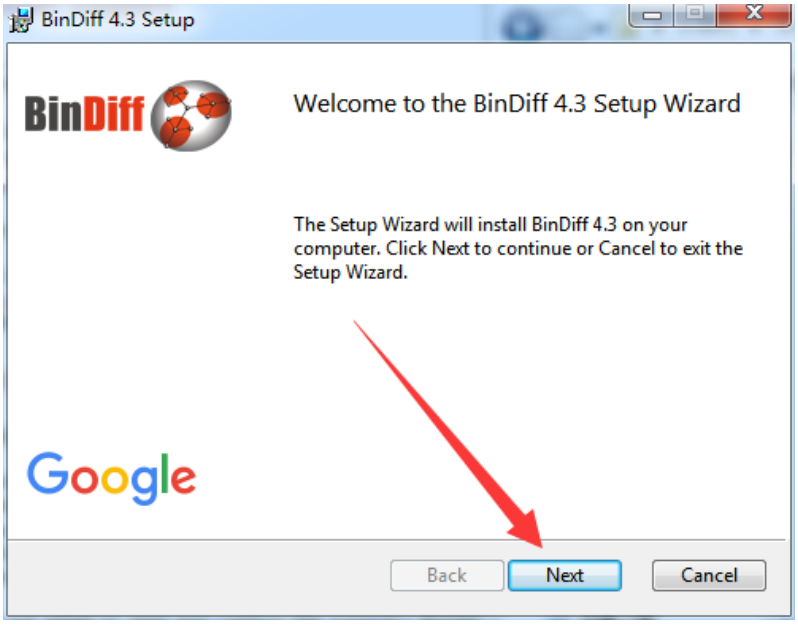
Filename	Size	SHA256
bindiff_4.3.0_amd64.deb	24.8M	98776bd9a61a29e4c8518b0ff0ae0a66518ab7c759aead5e3fcf2e6d3bcd1987
BinDiff4.3.dmg	29.7M	dd8f0df59b71d1cbc5661ac732c483883a43af43ec78a0b9293e745e23a59f84
bindiff430.msi	28.6M	e1915c18026d5a7288cca0c1ff71840bdb473b97c2235862a1241cda231791da

BinDiff 4.2.0

Filename	Size	SHA256
bindiff420-win-pluginsonly.zip	5.8M	355ade8528084f68379909bebc78040d62a8aaac21b8b3df83a1050a5a276d5c
bindiff-license-key.zip	990	34241d13dc80dbab9453a8166588ddc7a3fe4d8996211ae456339a55adda290c

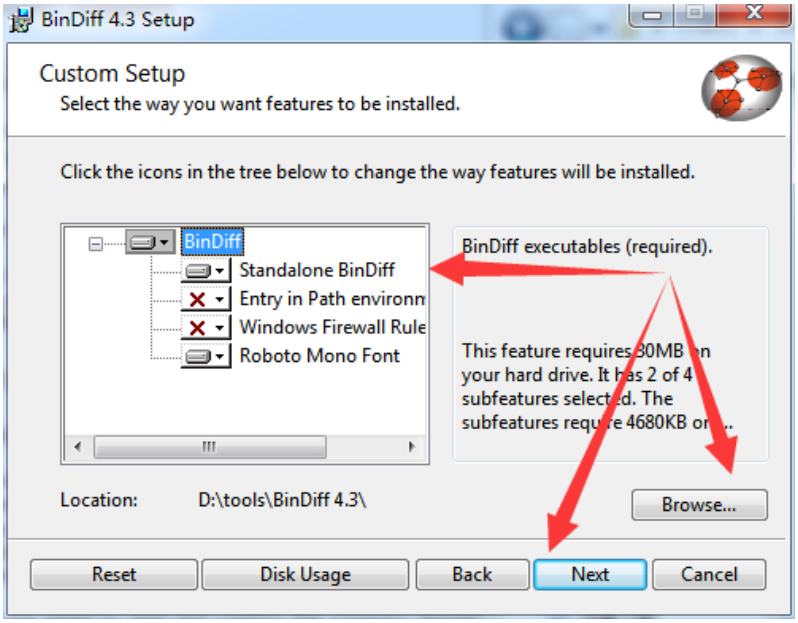
Note: We do not offer support. If you do contact us with bugs, feature requests or general questions, we will decide on a case by case basis on how to respond.

双击运行安装程序

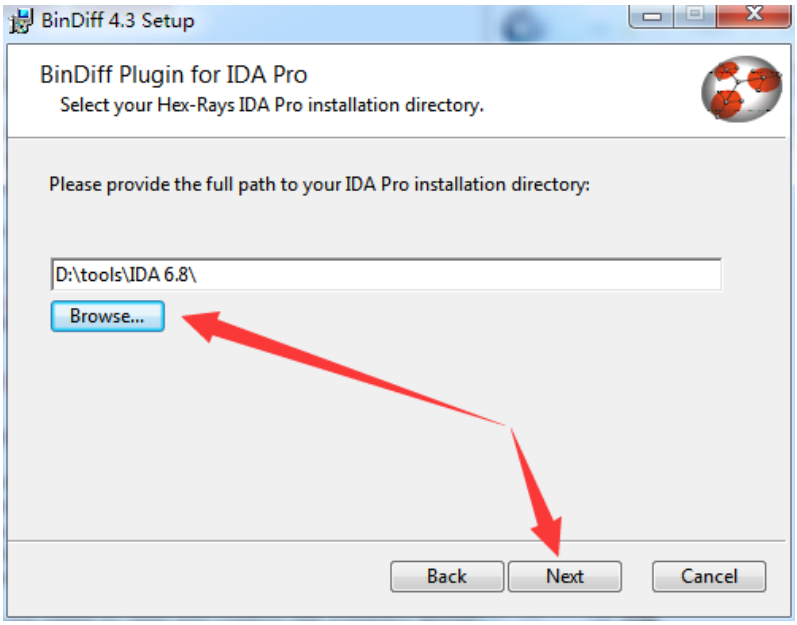


接受协议

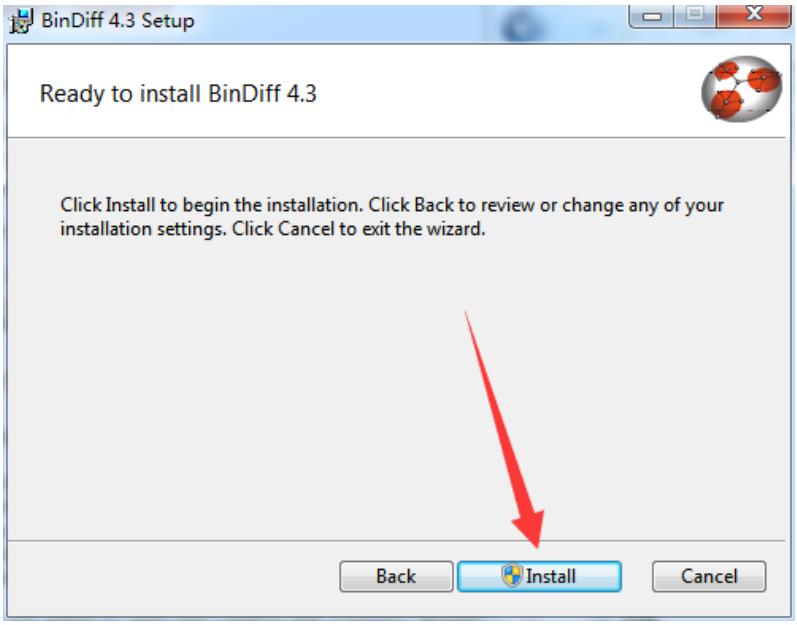
选择安装组件和路径



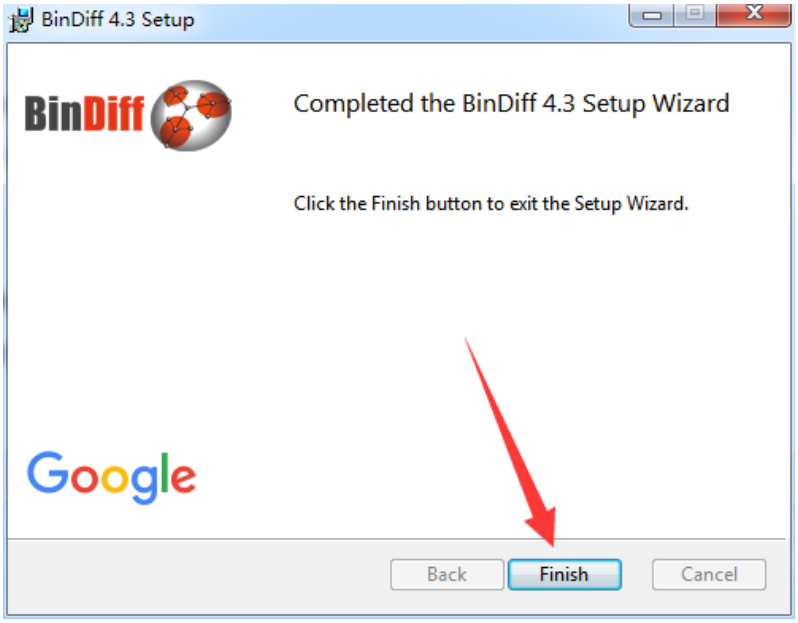
指出自己ida pro安装的目录



确认安装



安装完成



三、bindiff使用

3.1 编写比较程序

我这里使用cfree创建两个控制台项目bindiff1和bindiff2，分另写入以下两分代码并各自编译。代码的区别就只是把if语句的大于号改为小于号。

bindff1代码：

```
# include <stdio.h>
int main(){
    int a = 1;
    if (a > 1){
        printf("if brance\n");
    }
    else{
        printf("else brance\n");
    }
    getchar();
}
```

bindff2代码：

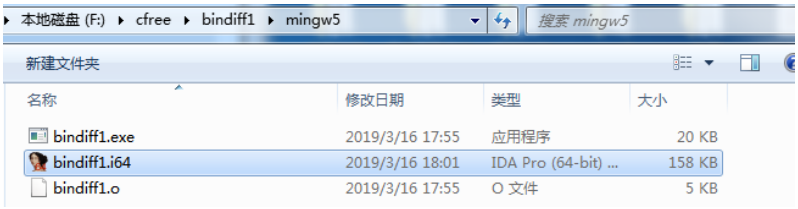
```
# include <stdio.h>
int main(){
    int a = 1;
    if (a < 1){
        printf("if brance\n");
    }
    else{
        printf("else brance\n");
    }
```

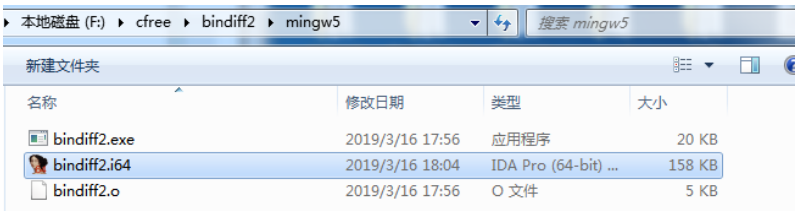
```
    getchar();
}
```

3.2 使用ida创建数据库

bindiff不能直接分析exe程序，而只能先使用ida pro的.i64数据库基础上进行分析。

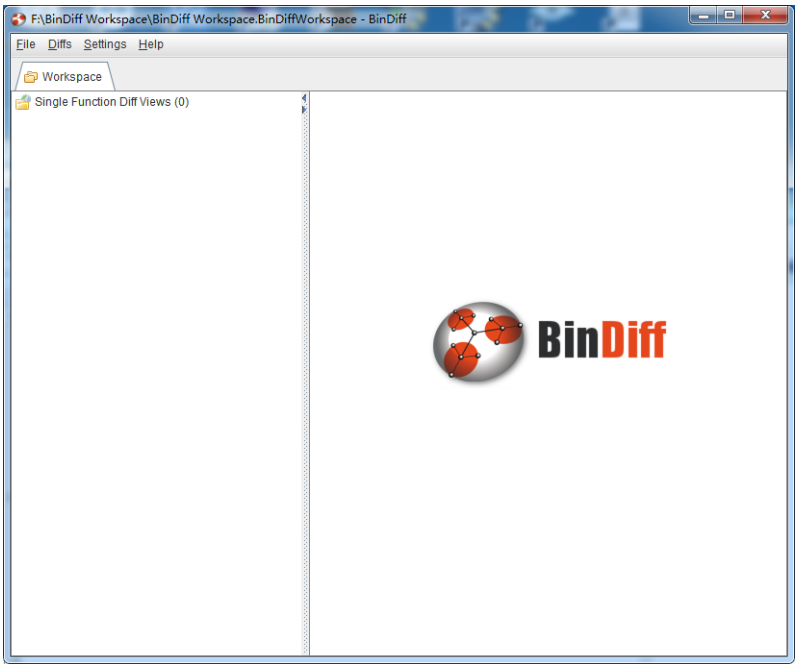
因此需要先用ida pro分别打开bindiff1.exe和bindiff2.exe再关闭，以创建.i64数据库。



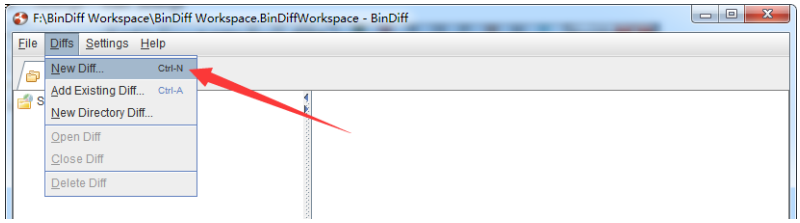


3.3 bindiff载入文件并比较

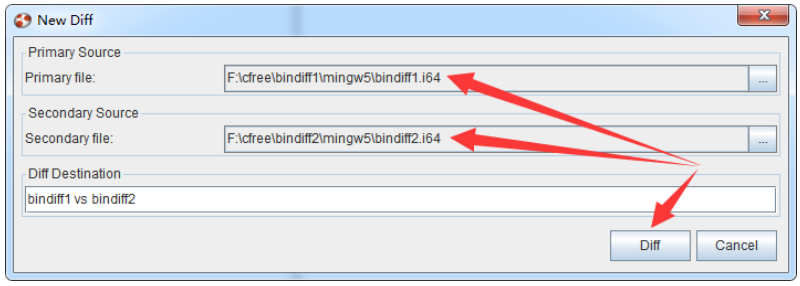
到安装目录bin文件夹下双击bindiff.jar即可启动bindiff，界面如下：



主菜单----Diffs----New Diff

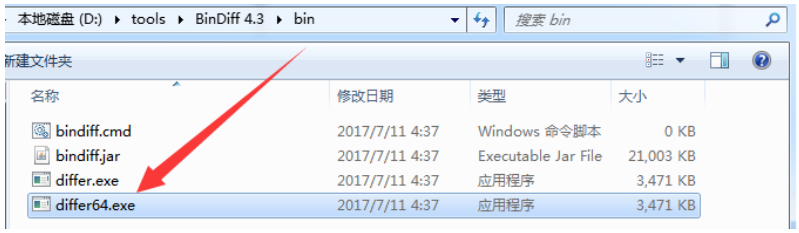


先后载入bindffi1.i64和bindiff2.i64，点击Diff进行比较



32位操作系统应该成功载入，64位操作系统可能会报错：“Can't find Diff engine at '...\differ64.exe'”

看意思是differ64.exe找不到，打开bindiff安装目录的bin文件夹，将bindiff.exe复制一份命名为bindiff64.exe，此时再重新载入比较即可。

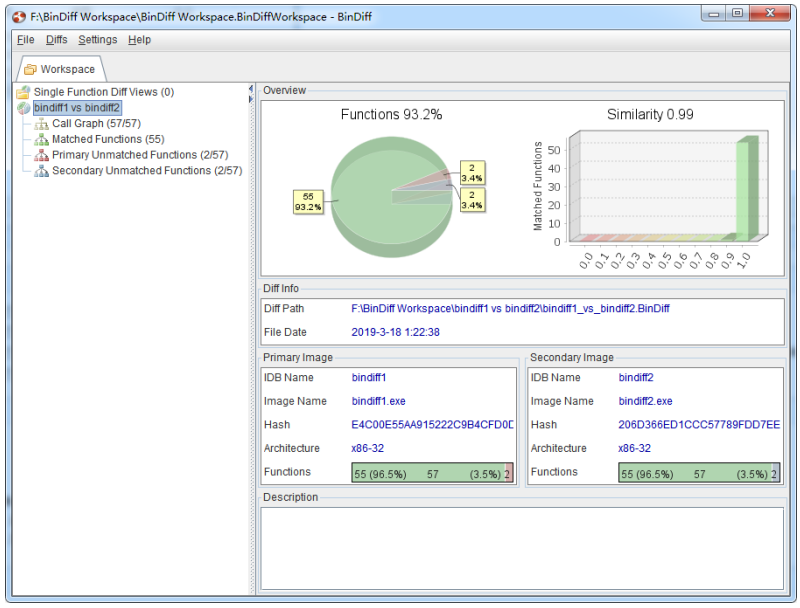


Call Graph----两个文件的函数调用图

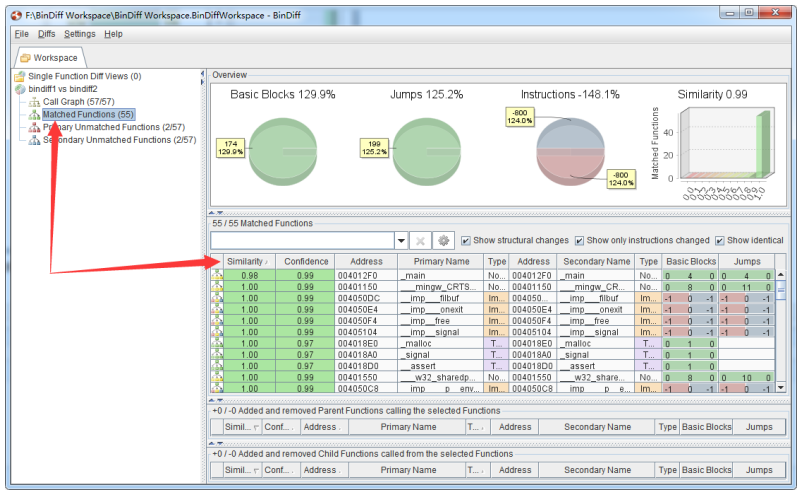
Matched Functions----两个文件的函数匹配度

Primary Unmatched Functions----

Secondary Unmatched Functions----



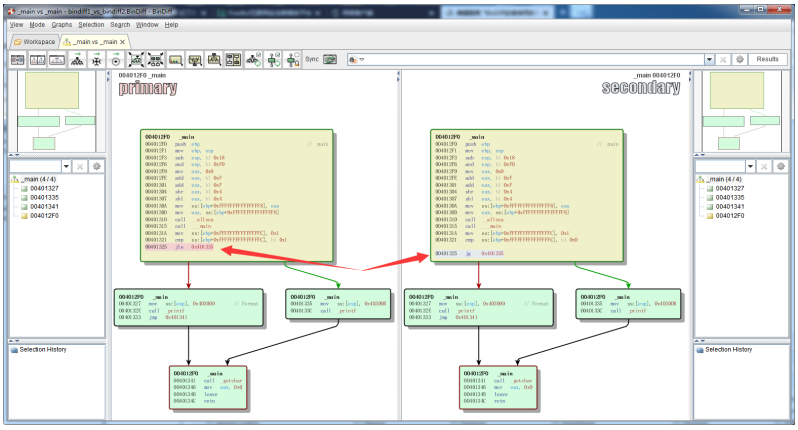
bindiff的主要简单使用是双击打开"Matched Functions"项，按相似度（Similarity）从低到高排序，相似度不为1的函数即为两个文件被改动的位置。



由上图可以看到，bindiff1.exe和bindiff2.exe只有_main相似度不为1，双击_main打开，两个函数不同的位置会被以有底色形式标出

如下图可以看到只有一条指令不同：bindiff1.exe是"jle 0x401335"（小于等于则进入else）而bindiff2是"jg 0x401335"（大于则进入else）

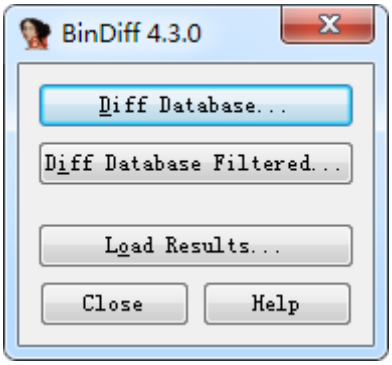
由此我们可以推断出bindiff2.exe相对于bindiff1.exe做的改动是：bindiff1.exe是"if > else"而bindiff2.exe是"if < else"。与我们的改动完全一致。



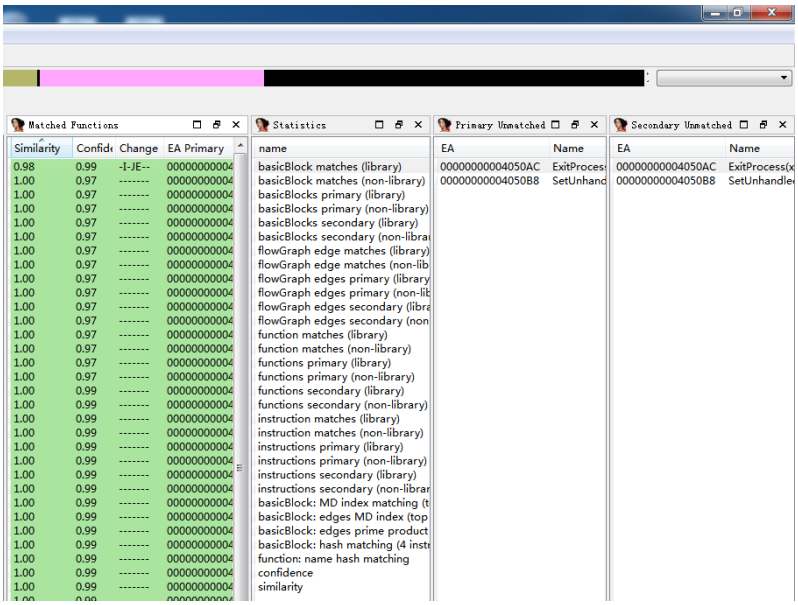
3.4 将bindiff以ida插件形式使用

在上边的操作中我们需要先用ida打开文件创建数据库，再使用bindiff打开比较，这是比较麻烦的。bindiff允许直接以ida插件的形式使用。（在安装bindiff时已同步安装为插件所以不需要另行安装）

先使用ida打开bindiff1.exe，然后使用“Ctrl+6”打开窗口，如下图所示



点击“Diff Database...”载入bindiff2.i64。如下图，仍是类似单独使用时的那几个窗口。当然这只是简单查看比较最后还是要打开bindiff。



3.5 bindiff使用注意点

第一点，相同的高级语言代码使用不同编译器编译出来的内容是有区别的。比如我这里使用cfree编译，倘若同样的代码你用VC++去编译那函数数量可能会多不少。但当然识别出的改动位置还是一个意思的。

第二点，相同的高级语言代码使用相同编译器不同的编译模式编译出来的内容是有区别的。比如VC++优化模式和普通模式编译出的汇编代码是有区别的。

第三点，bindiff只能识别出汇编指令的区别不能识别出汇编指令操作数的区别。即如上边“jle 0x401335”和“jg 0x401335”会被不同底色标出，但如果是“jle 0x401335”和“jle 0x401336”那将不会被标出。

参考：

https://www.zynamics.com/bindiff/manual/index.html

分类：Pentest