3.课程内容：综合实践三—补丁

1 补丁的概念
2 补丁工具
3 冷不丁
4 热补丁

(4) 三级--补丁-要求

- ①基础[T3-4-1]：对二进制程序进行冷补丁，修正程序错误（初级）
- ②进阶[T3-4-2]：对二进制程序进行热补丁，不停机修正程序错误（中、高级）
- ③编写check程序（高级），修复后的程序的运行，进行验证，可行性、有效性做对比、评价

（4）补丁的概念

- 补丁是指衣服、被褥上为遮掩破洞而钉补上的小布块。现在也指对于大型软件系统(如微软操作系统)在使用过程中暴露的问题（一般由软件安全公司、黑客或病毒设计者发现）而发布的解决问题的小程序。

（4）补丁-类型

- “高危漏洞”的补丁，这些漏洞可能会被木马、病毒利用，应立即修复。--所有者
- 软件更新的补丁，用于修复一些流行软件的安全漏洞。---开发者
- 功能性更新补丁，主要用于更新系统或软件的功能，可根据需要选择性进行安装。---开发者
- 恶意修复补丁：修改正常合法文件变成执行非法工作---敌对方

（4）补丁-角度—谁完成补丁

- 有源码—下载更新，比如，微软每个月第二个星期二周期性发布------

  程序的开发者

- 无源码--二进制程序
  - 二进制程序所有者补救措施
  - 第三方或安全公司
  - 敌对方

打补丁的方式

- 冷补丁-[T3-4-1]
- 被打补丁的程序以静态二进制文件的形式，采取相应的工具，对其进行修改
- 工具：二进制文件的编辑软件都可以

- 热补丁[T3-4-2]
- 被打补丁的程序运行中打补丁
- 工具：内存管理软件，hook技术，进程管理软件

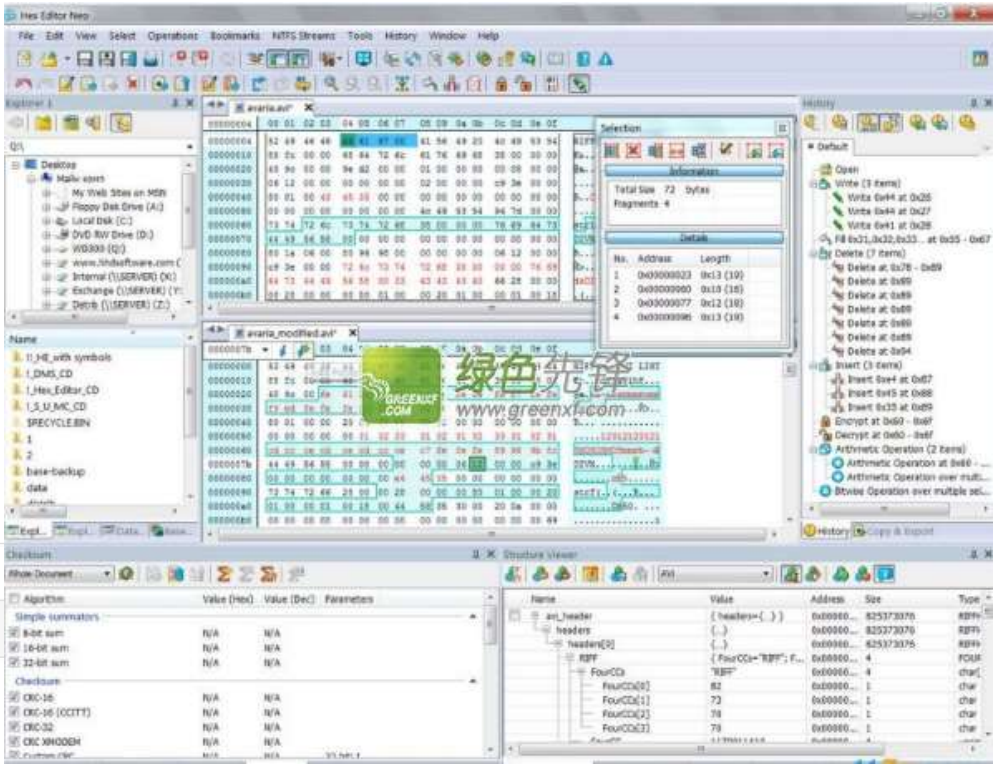
冷补丁[T3-4-1]二进制编辑工具

- **UltraEdit**

十六进制编辑模式通常用于非ASCII 文件，或二进制文件。这些文件一般都包含不可打印的字符，并且不是文本文件。

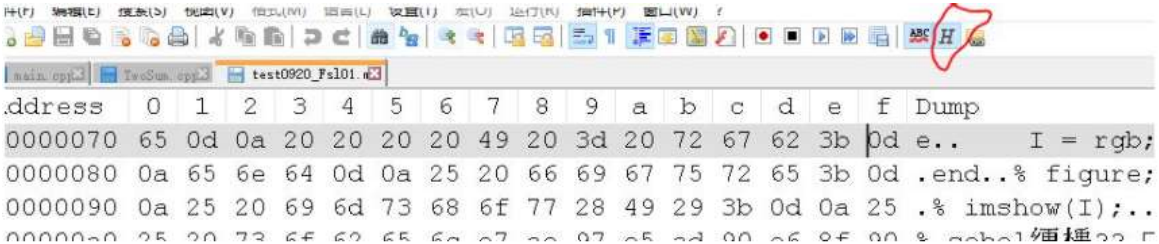
将屏幕范围分割成如下三个区域，其中文件偏移范围显示位于行首的字符相对于文件头部的字节偏移。

文件偏移:	十六进制表示	;ASCII 表示
000000h:	30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35	;123456789012345



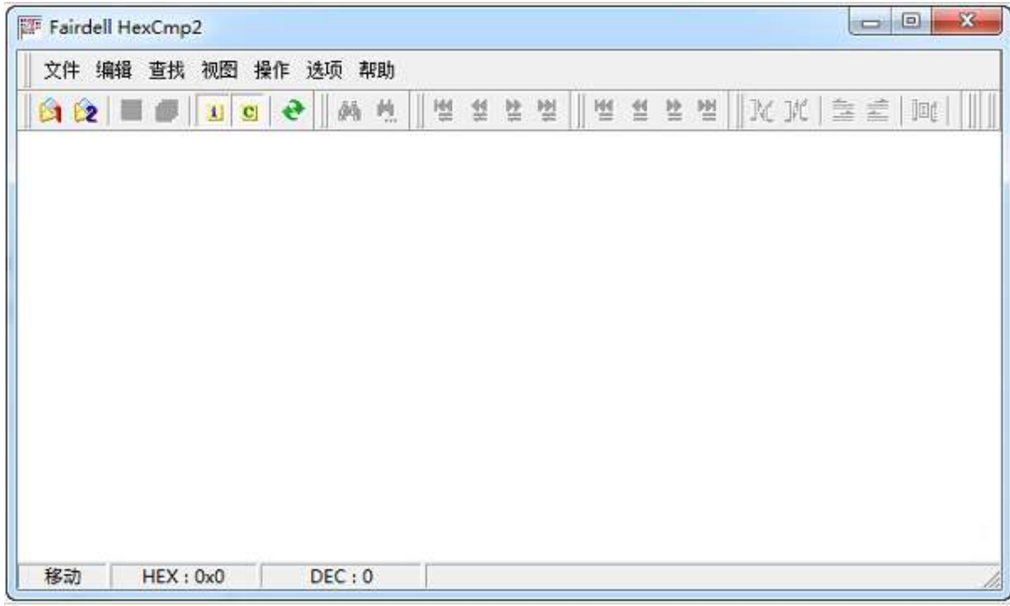
冷补丁二进制编辑工具

- **Notepad++查看、编辑二进制文件+安装附加组件HexEditor：**



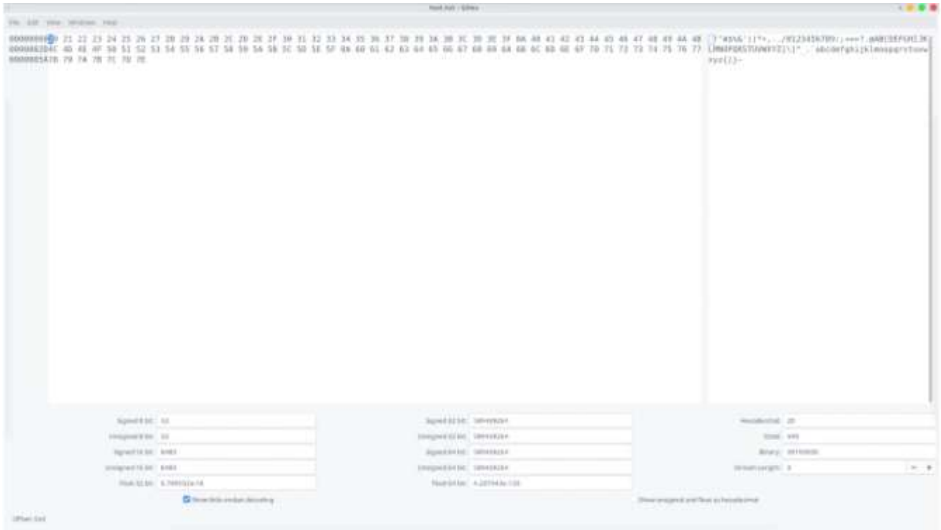
冷补丁二进制编辑工具

- **二进制文件比较编辑软件(fairdell hexcmp)二进制文件比较编辑工具**是一款文件管理的软件，享有实时高效的运行操作方式，支持同步文件运行管理，能够进行文件传输以及数据分析操作，程序可以帮助你快速的轻松的比较两个文件，并可以对你进行编辑操作,程序还拥有强大的搜索功能,可以执行基于十六进制或者字符的搜索操作。用于编辑、比较和修复。



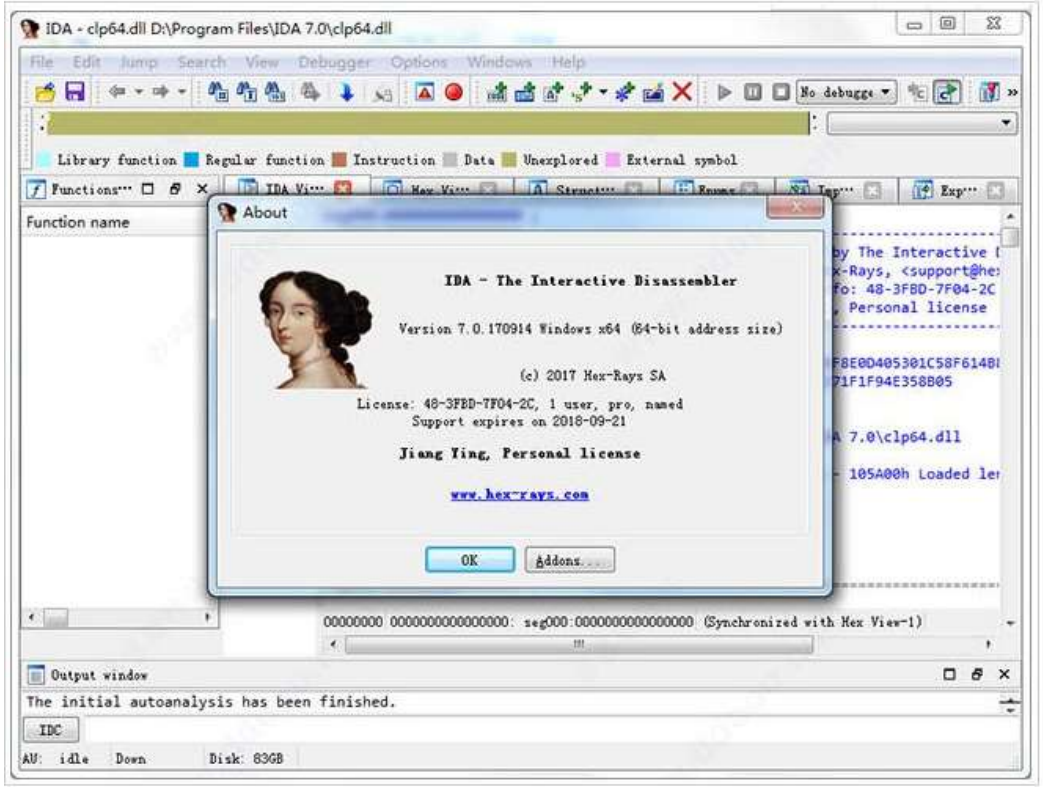
冷补丁二进制编辑工具

- Linux下二进制文件编辑神器—Ghex （免费）
- GHex是一个简单的二进制文件编辑器。它允许用户使用多级撤消/重做机制查看和编辑hex和ascii中的二进制文件。功能包括查找和替换功能，二进制，八进制，十进制和十六进制值之间的转换，以及使用另一种用户可配置的多文档界面概念，该概念允许用户使用多个视图编辑多个文档。



冷补丁二进制编辑工具

- IDA Pro是一款世界顶级的交互式反汇编工具，IDA Pro全名Interactive Disassembler Professional(交互式反汇编器专业版)，是Hex-Rays公司的旗舰产品，目前最新版为IDA Pro7.0。主要用在反汇编和动态调试等方面，支持对多种处理器的不同类型的可执行模块进行反汇编处理，具有方便直观的操作界面，可以为用户呈现尽可能接近源代码的代码，减少了反汇编工作的难度，提高了效率。

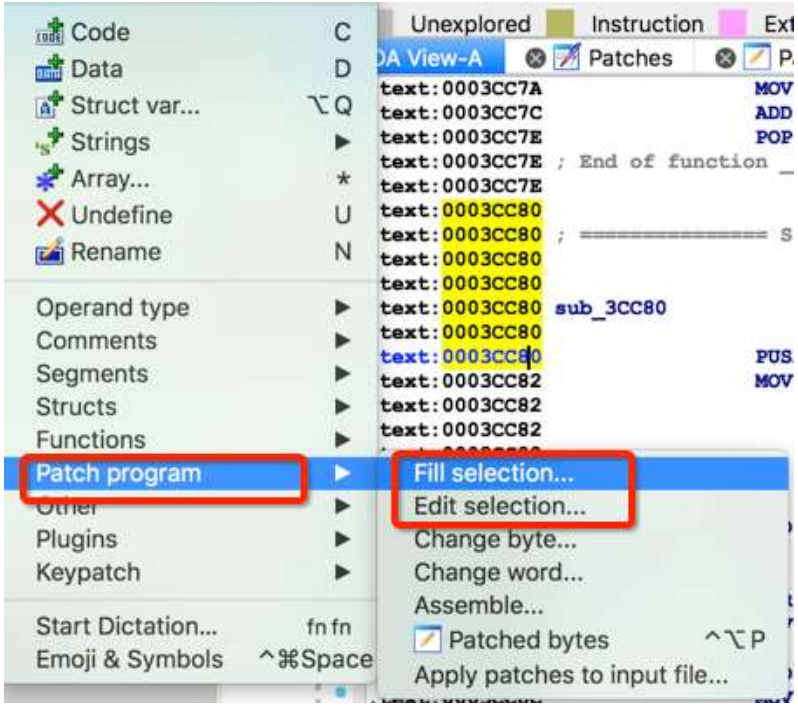


IDA Pro功能特色

- 1、可编程性
  IDA Pro包含了一个由非常强大的类似于宏语言组成的完全开发环境，可用于执行简单到中等复杂的自动化任务。对于一些高级任务，开放式插件架构对外部开发人员是没有限制的，这样可以完善IDA Pro的功能。
- 2、交互性
  IDA Pro拥有完全的互动性，IDA可以让分析师重写决策或者提供相应的线索。交互性是内置程序语言和开放式插件架构的最终要求。
- 3、调试器
  IDA Pro调试器补充了反汇编的静态分析功能：允许分析师通过代码一步一步来调查，调试器经常会绕过混淆，并得到一些能够对静态反汇编程序进行深入处理的数据，有些IDA调试器也可以运行在虚拟环境的应用上，这使得恶意软件分析更有成效。
- 4、反汇编
  IDA Pro可用的二进制程序的探索开发，也能确保代码的可读性，甚至在某些情况下和二进制文件产生的源代码非常相似。该程序图的代码可以为进一步的调查提供后期处理。

IDA Pro功能特色

- 5.补丁功能ida-patcher，支持的CPU架构：
  support Arm, Arm64 (AArch64/Armv8), Hexagon, Mips, PowerPC, Sparc, SystemZ & X86 (include 16/32/64bit).
- 6.支持的平台Windows, MacOS, Linux



IDApatcher打补丁

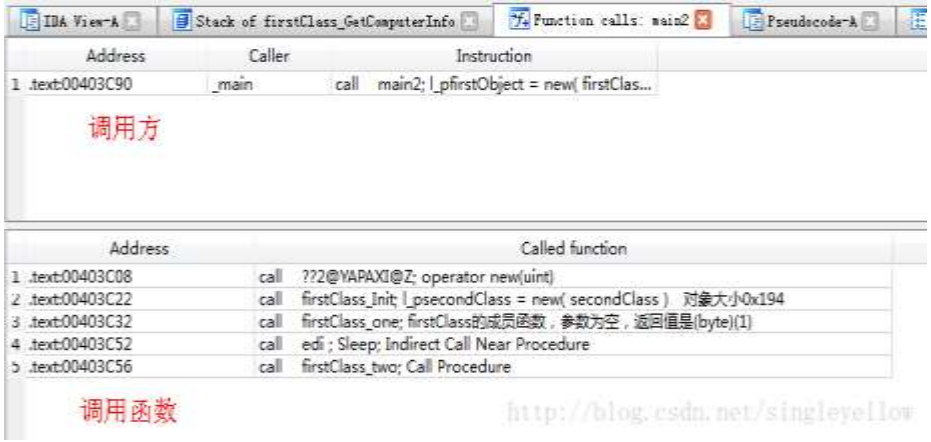
- 我们三级所讲的方式基本上为修改原程序二进制代码的方式，适用于Windows95之后的平台（Win32）的可执行文件都是PE格式，.exe、.ocx、.dll.等都是常见的PE格式的文件映像，看一个文件是否为PE文件，不是看它的扩展名，而是看它的文件头中是否有PE文件头标示和具体的文件内容。

IDA Pro图形接口：

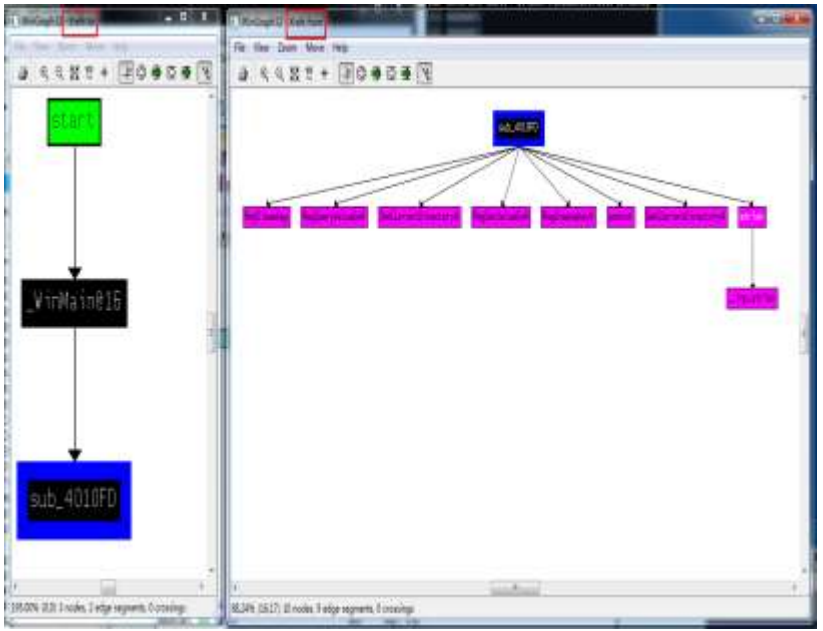
- 文本模式：文本模式左侧部分被称为箭头窗口，显示了程序的非线性流程，实线标记的是无条件跳转，虚线标记了条件跳转，向上的箭头表示一个循环。
- 图形模式：图形模式中，箭头的颜色和方向显示程序的流程，红色表示一个条件跳转没有被采用，绿色表示这个条件跳转被采用，蓝色表示一个无条件跳转被采用，向上的箭头同样表示一个循环条件。
- IDA窗口分布：**函数窗口：列举可执行文件中的所有函数，并显示每个函数的长度。这个窗口中每个函数关联了一些标志，如L代表此函数是库函数。
- 名称窗口：列举每个地址的名字，包括函数、命名代码、命名数据、字符串。
- 字符串窗口：显示所有字符串，默认显示长度超过5个字符的ASCII字符串。
- 导入表窗口：列举一个文件的所有导入函数。
- 导出表窗口：列举一个文件的所有导出函数，一般多用于分析DLL文件。
- 结构窗口：列举所有的活跃数据的结构布局。软件程序使用链接和交叉引用：常见的几个链接类型：子链接：一根函数开始的链接，如printf本地链接：跳转指令目的地址的链接，如loc_40107E偏移链接：内存偏移的链接**导航栏：**导航栏包括一个以颜色伪代号的被加载的二进制地址空间的线性视图

IDA-pro函数调用地址查找

- 1View->Open subviews->Function calls 显示出函数调用窗口，如下：



- 2点击按钮Display graph of xrefs from current identifier(从当前标识符绘制交叉引用图)



-附录IDApro-常见符号

- IDA图形视图会有执行流，Yes箭头默认为绿色，No箭头默认为红色，蓝色表示默认下一个执行块。我们可以在左侧查看代码的运行过程，按下空格键也可以直观地看到程序的图形视图。IDA View-A是反汇编窗口，
- HexView-A是十六进制格式显示的窗口，
- Imports是导入表（程序中调用到的外面的函数），
- Functions是函数表（这个程序中的函数），
- Structures是结构，
- Enums是枚举。
- 在反汇编窗口中大多是eax, ebx, ecx, edx, esi, edi, ebp, esp等。这些都是X86 汇编语言中CPU上的通用寄存器的名称，是32位的寄存器。这些寄存器相当于C语言中的变量。
  EAX 是”累加器”(accumulator), 它是很多加法乘法指令的缺省寄存器。
  EBX 是”基地址”(base)寄存器, 在内存寻址时存放基地址。
  ECX 是计数器(counter), 是重复(REP)前缀指令和LOOP指令的内定计数器。
  EDX 则总是被用来放整数除法产生的余数。
  ESI/EDI 分别叫做”源/目标索引寄存器”(source/destination index),因为在很多字符串操作指令中， DS:ESI指向源串,而ES:EDI指向目标串。
  EBP 是”基址指针”(BASE POINTER), 它最经常被用作高级语言函数调用的”框架指针”(frame pointer)。
  ESP 专门用作堆栈指针，被形象地称为栈顶指针，堆栈的顶部是地址小的区域，压入堆栈的数据越多，ESP也就越来越小。在32位平台上，ESP每次减少4字节。

基础[T3-4-1]：对二进制程序进行冷补丁---修正程序错误（初级）

- 实验一：简单修改二进制文件实现漏洞修补：

1、格式化字符串漏洞print_with_puts

```c
#include<stdio.h>

int main() {
        puts("test1");
        char s[20];
        scanf("%s", s);
        printf(s);
        return 0;
}
```

编译：

```
root@DESKTOP-HUI9I31:/# gcc print_with_puts.c -o print_with_puts
```

漏洞验证：

```
root@DESKTOP-HUI9I31:/# ./print_with_puts
test1
%s%s%s%s%s%s%s%s%s%s%s%s
段错误（核心已转储）
```

格式化字符串漏洞发生的条件就是格式字符串要去的参数和实际提供的参数不匹配。

1、格式化字符串漏洞print_with_puts

- 在只有二进制文件而没有源代码的情况下，用IDA进行反编译，注意划线处：

主要是三个区域：地址区、OpCode区（操作码区）、反编译代码区

1、格式化字符串漏洞print_with_puts

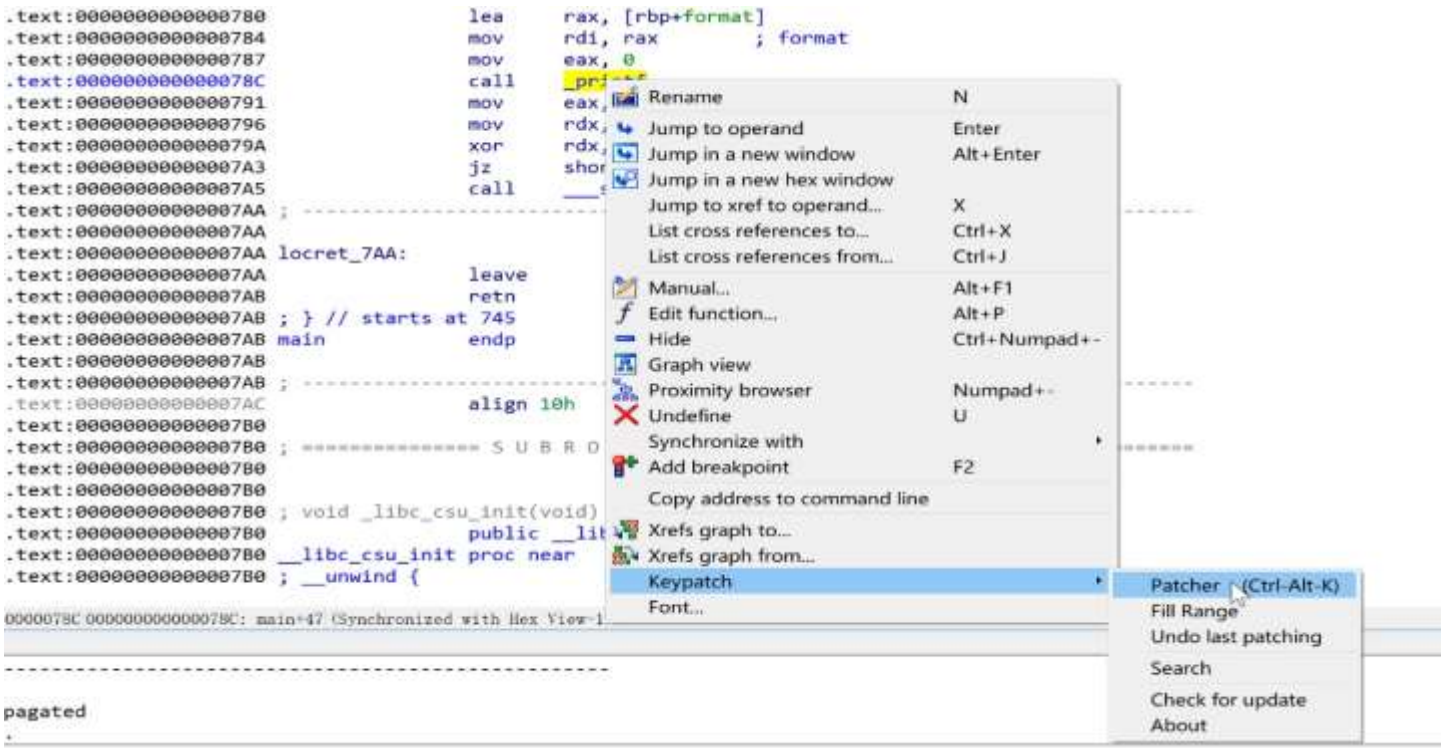
- 观察到该程序中存在puts函数，且调用puts函数与调用printf函数的指令均为五个字节，想到可以将call _printf简单修改为call _puts，方法如下：

(1)查看puts函数的地址：结果为0x0610

```
.plt:0000000000000610 _puts           proc near               ; CODE XREF: main+1E↓p
.plt:0000000000000610                 jmp     cs:puts_ptr
.plt:0000000000000610 _puts           endp
```

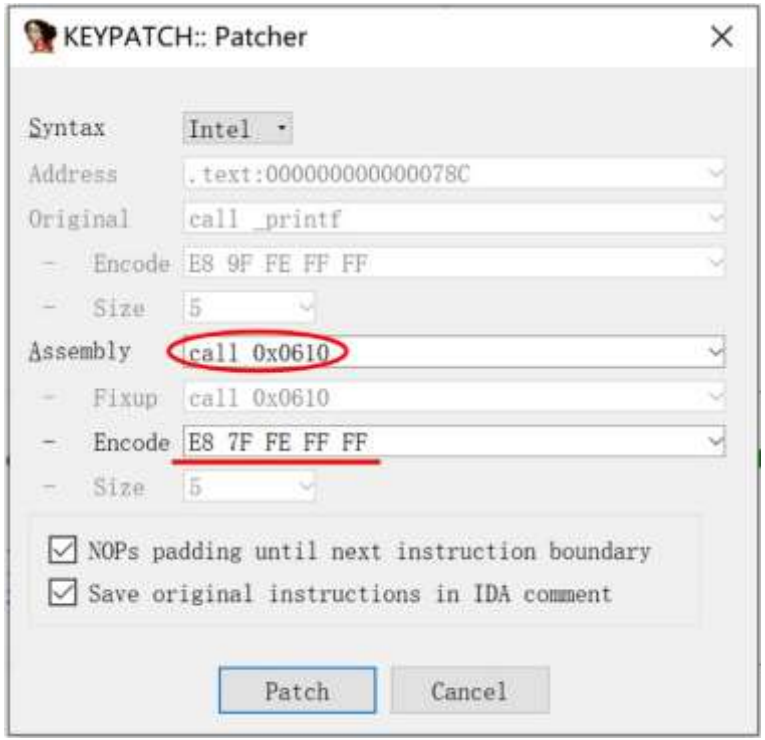
(2) 选中调用printf指令的语句，通过鼠标右键或Ctrl+Alt+K快捷键调用IDA插件Keypatch：

1、格式化字符串漏洞print_with_puts



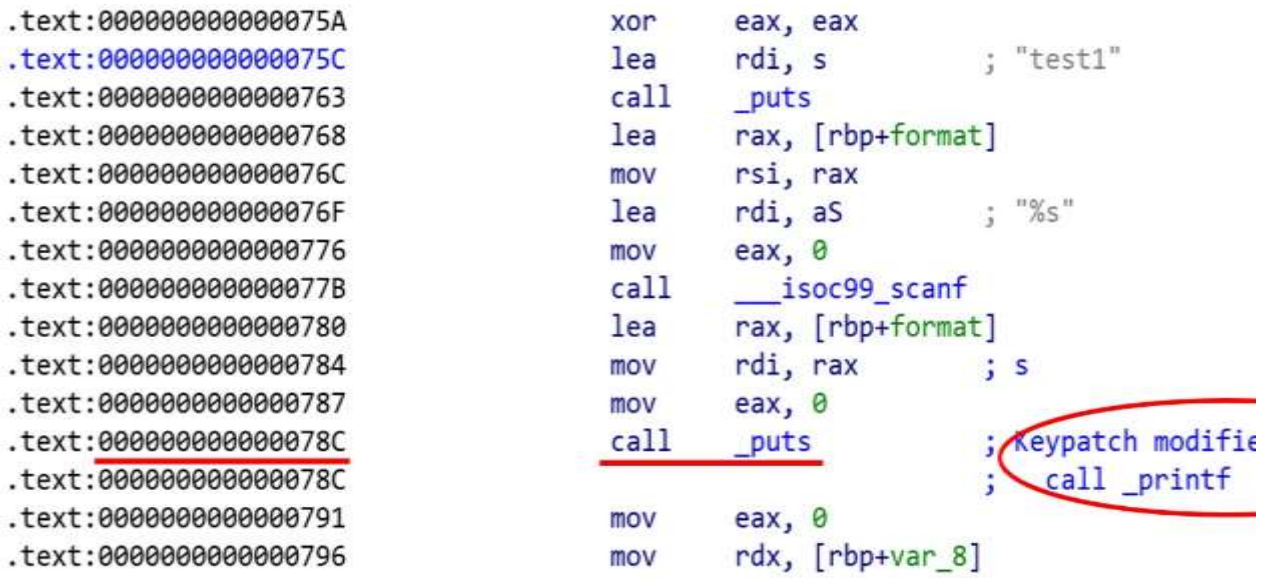
1、格式化字符串漏洞print_with_puts

(3)将调用printf函数的语句修改为调用puts函数的语句，注意，由于Keypatch不能识别符号地址跳转，因此修改时不能使用call _puts这样的语句，而应该直接给定跳转地址，这也是第(1)步中必须准备好puts函数地址的原因：



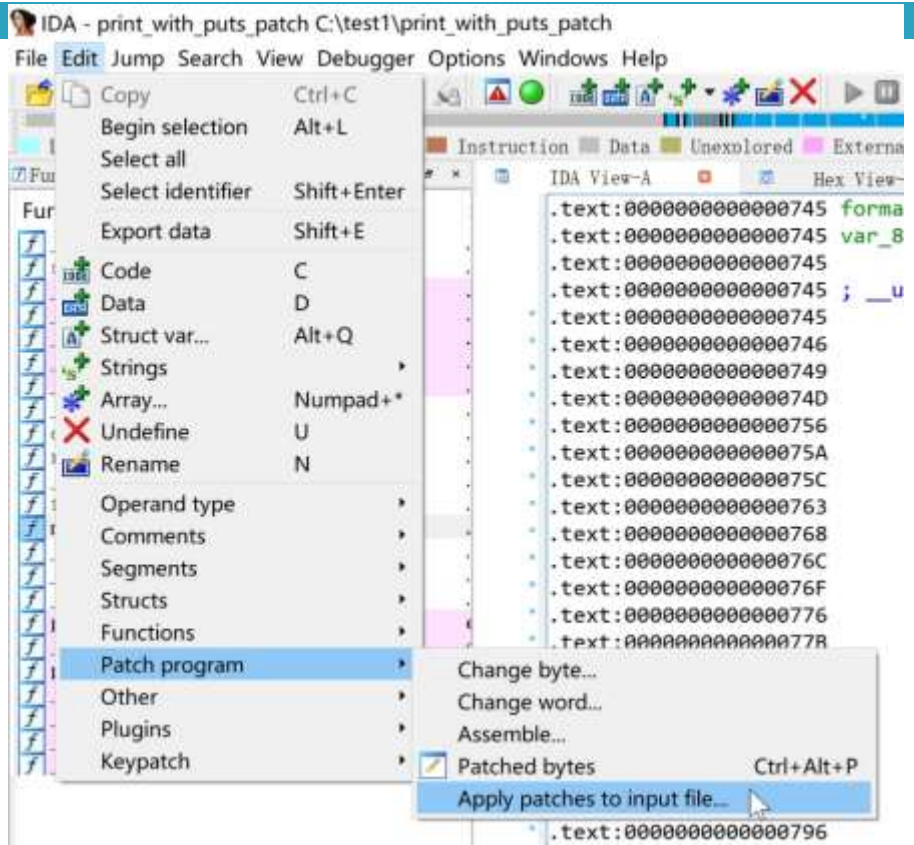
1、格式化字符串漏洞print_with_puts补丁

- Keypatch修改后的结果为：



1、格式化字符串漏洞print_with_puts补丁

(4)通过IDA将Keypatch的修改结果保存到二进制文件：



补丁后：

```
root@DESKTOP-HUI9I31:/# ./print_with_puts_patch
test1
%s%s%s%s%s%s%s%s%s%s%s%s
%s%s%s%s%s%s%s%s%s%s%s%s
```