开源代码分析技巧之——gdb 单步调试

在海量的源码面前，如何更好的切入到源码，锁定我们需要的那一行，有时显得格外重要。

而通常来讲，我们知道源码包，或多或许我们也同时有一些参考文档（可能不全），并且知道源码能提供的一些功能。比如以 Samba4.0.0 的源码为例，在 samba4.0.0/source4 下就有介绍 source4 新功能说明的帮助文档；在 kerberos 相关文件夹下就有介绍 kerberos 协议实现相关的文档；在 Samba 官网上有其从 80-90 年代一路走来各个版本的文档，信息量非常大，筛选需要花些时间。而对于一些功能的提供，往往是一些提供给我们可用的工具，比如安装 Samba 后在/usr/local/samba/bin 下有 samba-tool、tdbtool、rpcclient 等工具。这样工具的用法，我们可以 man 一下，不同的参数便一目了然。

顺着这个思路，我们就可以加上参数、子参数，运行设定的工具，这时候就有了执行这些程序的进程，可通过 ps –aux 查看。有了这些进程，我们就可以展开我们的单步调试了。

1、从 SAMBA 源码现身说法

对于比如/usr/local/samba/bin 目录下的文件，可以通过 File 查看文件属性，有的是编译好的二进制可执行文件，有的则是一些 python 脚本，可以 vi 打开查看源码的。但是二者都可以运行。对于前者二进制程序我们可以直接 gdb 可执行文件名称，如 gdb rpcclient 直接调试。而对于后者，我们只能通过 pdb（python 的单步调试方法）和 gdb 结合的方式运行：

第一步，pdb 可执行文件名称，可以参数执行的进程号；

第二步，通过 ps –aux 查看执行的进程号；

第三步，gdbattach 进程号进行调试。

2、gdb 常用参数

这个可以参考网络上有网友总结的很详细的版本。我列举下自己用的比较多的，抛砖引玉一下。

序号	单步调试指令	简写	释义
1	run	r	运行
2	stop	stop	暂停程序执行
3	break	b	设置断点，可用形式（b 文件名:行号 / b 行号）
4	continue	c	继续，直至下一个断点
5	step	s	在某个函数断点处，执行 s 可以步入该函数单步调试
6	next	n	执行到当前模块下的下一条指令
7	enable		使得断点有效（enable 断点号）
8	disable		使断点无效（disable 断点号）

9	delete		删除断点（delete 断点号）
10	clear		清除断点
11	ignore		忽略断点（ignore 断点号忽略断点次数）
12	list	l	显示当前执行程序的代码行，默认 10 行，可手动修改
13	info		比如: info break 可以查看当前所有断点信息
14	print	p	打印参数信息（p 参数/变量名/数组成员/结构体成员）
15	ptype/whatis		显示数据类型
16	backtrace	bt	查看调用堆栈信息
17	command	command	在 break 后加 command，会提示输入信息:此时类似一个脚本程序，你可以输入想要在断点处打印的信息，输入 quit 退出。（非常适合看循环单步调试的信息，不用每次手动输入）
18	search text		至上往下搜索，显示在当前文件中包含 text 的代码行
19	reverse-search text		至下往上搜索，显示包含 text 的代码行
20	attach		attach 加进程号，进程相关调试
21	kill		结束当前的调试
22	quit		退出 debug

注意点：对于含参数指令的单步调试方法，比如 rpcclient –U administrator%123456 –c “dsgetncchanges” 192.168.123.1，指令为 rpcclient，后面的都是参数。这时候我们可以先运行 gdb rpcclient ，此时调试并没有运行，可以通过 run –U administrator%123456 –c “dsgetncchanges” 192.168.123.1 进入调试运行状态，加断点调试等。

3、开源代码 gdb 之我的思考

在开源代码中，当我们不知道要 break 哪里才能有找到我们需要的的代码行或者代码片段时，我们可以结合以下几点定位：

第 1 点：结合打印日志信息，从中提取搜索关键词，放入代码中搜索，便于定位断点区域；

第 2 点：根据参考文档（自带的或者官网上的）,锁定我们调试的代码行；

第 3 点：没有可供参考的文档，只有源码，我们可以根据函数名或者文件名猜测其可能在的区域，在可能的

区域多加几个断点试试。

4、结语

gdb 相关调试博大精深，需要在实践中揣摩，有了它，我们的静谧的源码才能动起来，动起来才能更利于我们源码的进一步深入分析。