Tutorial notes from SecurityTube "GDB Expert Course" on youtube: http://goo.gl/tPTa7t

Working in 32bit Kubuntu 12.04LTS VBM

Install these packages (were already installed by default in my case)

```
        build-essential
        make
        libglib2.0-dev

        Can check their installation with:   '% apt-cache pkgnames'
```

----- From Video 1

Include debug symbols into executable file during compile time:

```
        % gcc -ggdb test.c -o test

        There are various symbol file types
            - DWARF 2
            - COFF
            - XCOFF
            - Stabs

        Just use -g for operating system default, or -ggdb for those optimal for gdb (
as above)
```

Can also include them explicitly at gdb run time,

----- From Video 2

```
  (gdb) list [line_number]                ; this will list the source code

      But note that the source code is not in the executable, it is referenced from th
e executable
      back into the .c source file (which if missing will cause a problem)

  (gdb) info source
      ; gives information about nature of source, format, etc
  (gdb) info sources
      ; shows files to which source references will go

  (gdb) info variables
      ; gives information on global and static variables (not locals)

  (gdb) info functions
      ; this will list out the functions from symbol table

      Interesting to note that the '_start' function is in the nondebug symbol list be
cause glib
      will start there, and then later call my 'main()' function after the startup cod
e

  (gdb) info scope function
      ; this gives local variables and functions inside particular scope

  (gdb) info scope [tab]
      ; this will provide a list of all possible scopes that could be specified
```

How to get debugging symbols out from and exe file?

```
  % objcopy --only-keep-debug program program_symbols
      ; this copies out symbols from 'program' into 'program_symbols'
      ; can check the resulting binary symbol file with emacs by using function 'hexl-
find-file'
```

```
  % strip --strip-debug program
      ; this will strip out and remove my debugging symbols from
  % strip --strip-debug --strip-unneeded
      ; will do a hard strip of even the non-debugging symbols
      ; THIS IS THE SECURE THING TO DO PRIOR TO DISTRIBUTING ANY BINARY EXECUTABLE!!!
```

Now, how to explicitly add symbols to an executable that doesn't have any inside? Insi
de gdb:

```
  (gdb) symbol-file program_symbols
  (gdb) info functions ; to check it
```

To add symbols back into the binary executable?

```
  % objcopy --add-gnu-debuglink=program_symbols program
  % gdb program

  (gdb) info functions
```

We can also dump symbol table into a test formated file as follows:

```
  (gdb) maint print symbols file.txt
```

From within gdb we can get a listing of all the sections defined in the executable fil
e:

```
  (gdb) info files
      ; this doesn't seem to depend on whether or not symbols are embedded
```

----- From Video 3