Reversing with GDB and GEF

Starting GDB

`gdb` *program* [*core*|*pid*]
`gdb` *gdb-options* [`--args` *program args*]

At startup, GDB reads the following init files and executes their commands: `/etc/gdb/gdbinit` and `~/.gdbinit`, plus `./.gdbinit`, if `set auto-load local-gdbinit` is set to `on`.

Some options are:

`-q/--quiet/--silent`	don't print version number on startup
`-h/--help`	print help
`--tty=`*TTY*	use *TTY* for I/O by debugged program
`--nh`	do not read `~/.gdbinit`
`-x` *FILE*	execute GDB commands from *FILE*
`-ix` *FILE*	like `-x` but execute before loading inferior
`-ex` *CMD*	execute a single GDB command; may be used multiple times and in conjunction with `-x`
`-iex` *CMD*	like `-ex` but before loading inferior
`-s` *SYMFILE*	read symbols from *SYMFILE*
`--write`	set writing into executable and core files

To quit, `q`[`uit`] or *Ctrl-D*.

You can invoke commands on the standard shell by using:
`shell` *command-string*
or simply: `!`*command-string*

You can abbreviate a gdb command to the first few letters of the command name, if that abbreviation is unambiguous; and you can repeat certain gdb commands by typing just *Return*. You can also use the *TAB* key to get gdb to fill out the rest of a word in a command (or to show you the alternatives available, if there is more than one possibility).

You can always ask gdb itself for information on its commands, using the command `h`[`elp`].

Getting information

`i`[`nfo`] is for describing the state of your program. For example, you can show the arguments passed to a function with `info args`; you can get a complete list of the info sub-commands with `help info`.

You can assign the result of an expression to an environment variable with `set`. For example, you can set the gdb prompt to a $-sign with `set prompt $`.

In contrast to `info`, `show` is for describing the state of gdb itself. You can change most of the things you can show, by using the related command `set`; for example, you can control what number system is used for displays with `set radix`, or simply inquire which is currently in use with `show radix`

To display all the settable parameters and their current values, you can use `show` with no arguments; you may also use `info set`: both produce the same display.

Logging output

`set logging on`	enable logging
`set logging off`	disable logging
`set logging file` *file*	change the current logfile default logfile is `gdb.txt`
`set logging overwrite [on|off]`	
`set logging redirect [on|off]`	output to both terminal and logfile?
`show logging`	show current logging settings

Starting your program

page 26