
U.S. ARMY CYBER SCHOOL
DEFEND
ATTACK
EXPLOIT

Security & Exploitation: Reverse Engineering Guides

Version Date: 26 MAR 2020

GDB Student Resource and Cheat Sheet

What is GDB?

GDB stands for GNU Project debugger. It is the defacto standard for debugging processes in Linux. Remember that a debugger is the primary tool used in dynamic software analysis and reverse engineering. They allow you to see what is happening in memory and the registers as a process is running.

Getting setup

Installation of packages

You will need ensure the following packages are installed on your Linux host before you can use gdb:

```
GDB
GCC
libc6:i386 - for 32 bit binaries
```

To install the 32 bit binaries:

```
dpkg --add-architecture i386
apt update
apt install libc6:i386
```

Start up GDB with on of these commands:

```
gdb /path/to/file
```

or

```
gdb
```

Common uses and commands

Command	Description	Example	Caveats
quit	Exits GDB	quit	Can be invoked as a shortened "q"

run	Use this to execute the file being analyzed.	run	Arguments or input can be redirected here as "run argument1" or "run < input" or "run <<< $(python input.py)" Can be invoked as a shortened "r"
file	This sets the file to be analyzed if you did not pass it as an argument to GDB when it was started	file /path/to/file	This can also change which file is being analyzed as needed
help	Displays the help information for a command	help run	
x	Shows the contents of memory at the address referenced	x/256h $esp - shows us 256 bytes of memory space starting from $esp in hexadecimal format - effectively shows us the contents of the stack	Ths command can be formatted many ways. Reference them here: https://visualgdb.com/gdbreference/commands/x
break	Sets a break point at a given location	break *0x05ffde00 - sets a break point at memory location 0x05ffde00	Can be invoked as a shortened "b" Can also be used with function names "break main" or a specific amount of bytes past the start of a function "break main+4" There are additional methods for where to break. Reference them here: https://visualgdb.com/gdbreference/commands/break
disassemble	Disassembles the instructions found at a given location	disassemble main	Can be invoked as a shortened "diasass"

info	info registers	Info is generic command that shows all of the information related to the contents of a given query. It has many subcommands that can be found using "help info"	"info registers" is one of the most common info subcommands. It can be invoked as a shorted "info r"
info functions	This is a subcommand of info. It displays all of the function names.	info functions	
next	Moves the program forward by one step, but does not enter subroutine if it is called.	next	Can be invoked as a shortened "n".
step	Moves the program forward by one step and follows subroutines if called.	step	Can be invoked as a shortened "s".
info proc map	A subcommand of info, but important enough to be mentioned seperately. This shows a list of mapped memory regions.	info proc map	Knowing the memory layout of a program can be incredibly useful.
find	This searches throughout a given memory range for a specific sequence of bytes or op codes.	find /b 0x05ffde00, 0x05ffff00de, 0xcc - searches for a breakpoint op code between 0x05ffde00 and 0x05ffff00de	Invoke "help find" for specific information like what the "/b" means.