PWNDBG CHEATSHEET

HTTPS://PWNDBG.RE/

GDB COMMANDS

file <path>
load binary file to debug

run [<args>...]
run program [with args]

starti [<args>...]
start program and stop
at its very first instruction

set args <args>...
set program arguments

break <where>
set a breakpoint

info breakpoints|threads|regs
list breakpoints/threads/register values

delete <breakpoint>
delete a breakpoint

next
go to next (source) line

step
go to next line stepping into functions

ni
go to next instruction

si
go to next instruction stepping
into functions

finish
run until current function returns

continue
continue program execution

print <what>
evaluate and print an expression

x/format <address>
examine memory with given format
(see `help x`)

apropos <topic>
find information about topic

backtrace
print backtrace (call stack)

up, down
move up/down the call stack

PWNDBG COMMANDS:

pwndbg [<topic>]
print info about pwndbg commands

config
show pwndbg configuration

theme
show pwndbg theme configuration

tip [--all]
print tips that are shown during startup

CONTEXT DISPLAY

context [<section>]
display context or a given context section
(regs, disasm, args, code, stack, backtrace,
expressions, ghidra, threads)

set context-sections [<sect1>] [<sect2>...]
set context to display only given sections

ctx-watch eval|execute <expression>
adds a given expression to be shown on context display

START COMMANDS

attachp <pid|name>
attach to given pid or process by part of its name

start [<args>...]
run and stop program at the first found symbol from:
main, _main, start, _start, init, _init or entry

entry [<args>...]
run and stop program at its entrypoint address

sstart [<args>...]
run and stop program at the __libc_start_main function

MEMORY COMMANDS

vmmap [<address|name>]
display memory mappings information
[filtered by address or name]

search <what>
search memory for a given value

telescope <where> [<count>]
examine memory dereferencing valid pointers

hexdump <where> [<count>]
print hexdump of given address

p2p <mapping_names> [<mapping_names>...]
pointer to pointer chain search (e.g. `p2p stack libc` will look for pointers to libc on the stack)

xinfo <where>
show offsets of the specified address from
various useful locations

STACK COMMANDS

retaddr
print return addresses on the stack

canary
print the global stack canary/cookie value
and finds canaries on the stack

NAVIGATION

xuntil <where>
continue until an address or function

nextcall
continue to next call instruction

nextjmp
continue to next jump instruction

nextret
continue to next return-like instruction

stepret
step until a ret instruction is found

stepuntilasm <asm code>
step until a given assembly instruction
(or mnemonic) is found

LINUX/LIBC/ELF COMMANDS

checksec
print binary mitigations status

piebase
print the relocated binary base address

got
print symbols in the .got.plt section

gotplt
print symbols in the .got.plt section

plt
print symbols in the .plt section

tls
print thread local storage address

MISC COMMANDS

distance <where1> <where2>
compute difference between two addresses

patch <where> '<instructions>...'
patch given address with given code/bytes

patch_list
list all applied patches

patch_revert <patch>
revert a patch

cymbol [...]
add, show, load, edit, or delete custom structures
in plain C (so they can be used e.g. with print command)

plist [...]
dump elements of a linked list (see `help plist`)

procinfo
display process information

errno [<errno value>]
print libc's errno error code string

GLIBC HEAP HACKING

heap_config
show glibc allocator hacking configuration

heap
iteratively print chunks on heap (glibc only)

vis_heap_chunks
visualize chunks on a heap

bins
print contents of all arena bins and thread's tcache

find_fake_fast <address>
find candidate fake fast or tcache chunks
overlapping the specified address

try_free <address>
check what would happen if free was called
with given address