Volatility 1.3 Memory Analysis Cheat Sheet

Typographical conventions

italics	parameters or variables whose actual names or values are to be supplied by the user
underlined	mandatory parameters, default values

Basic commands

python volatility *command* [*options*]

python volatility	list built-in and plugin commands

Common options

-h	detailed help for command
-b / --base=*address*	CR3 (in hex)
-t / --type=*type*	auto, pae, nopae
-H / --output-format=*fmt*	select format (text, sql, xml). Availability of formats varies depending on plugin.
-O / --outfile=*filename*	results (listings, etc.)

Convert and examine images

Converters

raw2dmp	convert from raw (dd) to crash dump
-f / --file=*filename*	raw image file
-o / --output=*filename*	crash dump
dmp2raw	convert crash dump into raw (dd)
-f / --file=*filename*	crash dump
-o / --output=*filename*	raw image
hibinfo	convert hibernation file into raw
-f / --file=*filename*	hiberfil.sys
o / --output=*filename*	raw image
dmpchk	list info about DMP file
-f / --file=*filename*	memory image file

Get basic image information

ident	identify memory image properties
-f / --file=*filename*	memory image file
datetime	date and time of memory image
-f / --file=*filename*	memory image file

Executable objects

Threads

thrdscan	scan for thread dispatcher objects
-f / --file=*filename*	memory image file
-s / --start=*offset*	start address (in hex)
-e / --end=*offset*	end address (in hex)
-l / --slow	scan in slow mode
thrdscan2	scan for threads (fast pool scanner)
-f / --file=*filename*	memory image file
thread_queues	print message queues
-f / --file=*filename*	memory image file

Processes

pslist	enumerate processes
-f / --file=*filename*	memory image file
psscan	scan for thread dispatcher objects
-f / --file=*filename*	memory image file
-s / --start=*offset*	start address (in hex)
-e / --end=*offset*	end address (in hex)
-l / --slow	scan in slow mode
-d / --dot	render process tree in DOT format
psscan2	scan for processes (fast pool scanner)
-f / --file=*filename*	memory image file
-d / --dot	render process tree in DOT format
pstree	show process hierarchy
-f / --file=*filename*	memory image file
-v / --verbose	show more information (path, command line)
procdump	dump process image
-f / --file=*filename*	memory image file
-o / --offset=*EPROCESS*	select by EPROCESS (in hex)
-p / --pid=*PID*	select by PID
-m / --mode={disk|mem}	extraction mode
-u / --unsafe	skip sanity checks
getsids	print process owner SIDs
-f / --file=*filename*	memory image file
jobscan	scan for job objects
-f / --file=*filename*	memory image file
-p / --processinfo	show processes in job

Drivers

driverscan	scan for driver objects
-f / --file=*filename*	memory image file
driverirp	show driver with IRP handlers
-f / --file=*filename*	memory image file

DLL and Modules

dlllist	list loaded DLLs
-f / --file=*filename*	memory image file
-o / --offset=*EPROCESS*	select by EPROCESS (in hex)
-p / --pid=*PID*	select by PID
dlldump	dump DLL to disk
-f / --file=*filename*	memory image file
-d / --directory=*DIR*	output directory
-p / --pattern=*REGEX*	dump matching file(s)
-I / --ignore-case	pattern is case-insensitive
-m / --mode={disk|mem}	extraction mode
-u / --unsafe	skip sanity checks

Version 1.3.1.0
2009-11-01

Volatility 1.3 Memory Analysis Cheat Sheet

modules — enumerate kernel modules
- -f / --file=*filename* — memory image file

modscan — scan for kernel modules
- -f / --file=*filename* — memory image file

modscan2 — scan for kernel modules (fast)
- -f / --file=*filename* — memory image file

moddump — dump module(s) to disk
- -f / --file=*filename* — memory image file
- -p / --pattern=*REGEX* — dump matching file(s)
- -I / --ignore-case — pattern is case-insensitive
- -o / --offset=*offset* — module offset in hex
- -m / --mode={disk|mem} — extraction mode
- -u / --unsafe — skip sanity checks

Files

files — list open files
- -f / --file=*filename* — memory image file
- -o / --offset=*EPROCESS* — select by EPROCESS (in hex)
- -p / --pid=*PID* — select by PID

fileobjscan — scan for file objects
- -f / --file=*filename* — memory image file

Network activity

Sockets

sockets — enumerate sockets
- -f / --file=*filename* — memory image file

sockscan — scan for sockets
- -f / --file=*filename* — memory image file
- -s / --start=*offset* — start address (in hex)
- -e / --end=*offset* — end address (in hex)
- -l / --slow — scan in slow mode

sockscan2 — scan for sockets (fast)
- -f / --file=*filename* — memory image file

Connections

connections — enumerate TCP connections
- -f / --file=*filename* — memory image file

connscan — scan for TCP connections
- -f / --file=*filename* — memory image file
- -s / --start=*offset* — start address (in hex)
- -e / --end=*offset* — end address (in hex)
- -l / --slow — scan in slow mode

connscan2 — scan for TCP connections (fast)
- -f / --file=*filename* — memory image file

Other objects

objtypescan — scans for object type objects
- -f / --file=*filename* — memory image file

eventscan — scans for events
- -f / --file=*filename* — memory image file

mutantscan — scans for mutants/mutexes
- -f / --file=*filename* — memory image file
- -s / --silent — suppress less meaningful results

symlinkobjscan — scans for symbolic link objects
- -f / --file=*filename* — memory image file

Memory

Address conversion and extraction

memdmp — list addressable memory
- -f / --file=*filename* — memory image file
- -o / --offset=*EPROCESS* — select by EPROCESS (in hex)
- -p / --pid=*PID* — select by PID

memmap — print memory map (virtual → physical)
- -f / --file=*filename* — memory image file
- -o / --offset=*EPROCESS* — select by EPROCESS (in hex)
- -p / --pid=*PID* — select by PID

strings — maps physical offset to virtual address
- -f / --file=*filename* — memory image file
- -s / --strings=*filename* — strings output file

The current implementation of Volatility's „strings" command is very slow. Suggested usage:

1. run a conventional strings command on a raw memory image, be aware of ASCII/ANSI and UNICODE encodings, e.g.
   a. strings –o *dumpfile* > *outfile*
   b. strings –td –eS *dumpfile* > *outfile*
2. ensure *outfile* is in format offset:string
3. shorten *outfile* as far as possible
4. add other interesting offsets and attach a meaningful label instead of a string
5. run Volatility's „strings" command

Virtual address descriptors

vadinfo — information about every single VAD
- -f / --file=*filename* — memory image file
- -o / --offset=*EPROCESS* — select by EPROCESS (in hex)
- -p / --pid=*PID* — select by PID

vaddump — save memory regions into files
- -f / --file=*filename* — memory image file
- -d / --directory=*DIR* — output directory
- -o / --offset=*EPROCESS* — select by EPROCESS (in hex)
- -p / --pid=*PID* — select by PID

vadwalk — produce VAD tree
- -f / --file=*filename* — memory image file
- -o / --offset=*EPROCESS* — select by EPROCESS (in hex)
- -p / --pid=*PID* — select by PID
- -d / --dot — render in DOT format
- -e / --tree — render as ASCII tree
- -l / --table — render as table

Version 1.3.1.0
2009-11-01

Volatility 1.3 Memory Analysis Cheat Sheet

Registry

Method

1. use „hivescan“ to find registry hive structures in memory
2. let „hivelist“ start from any of the found structures and produce a list of hives
3. use „hivedump“, „printkey“ or other tools to extract information from the proper hive
4. (optional) merge information from all hives into a single timeline and sort by date/time (Unix: sort –n)

Commands

hivescan	scan for CMHIVE
-f / --file=*filename*	memory image file
hivelist	list of registry hives
-f / --file=*filename*	memory image file
-o / --offset=*offset*	hive offset
printkey	print registry key
-f / --file=*filename*	memory image file
-o / --hive-offset=*offset*	hive offset
regobjkeys	list opened keys by process
-f / --file=*filename*	memory image file
-o / --offset=*EPROCESS*	select by EPROCESS (in hex)
-p / --pid=*PID*	select by PID

Secrets and encryption

keyboardbuffer	contents of real mode keyboard buffer
-f / --file=*filename*	memory image file
cryptoscan	finds TrueCrypt passphrases
-f / --file=*filename*	memory image file
suspicious	finds suspicious command lines
-f / --file=*filename*	memory image file

Registry

cachedump	decrypt domain hashes
-f / --file=*filename*	memory image file
-s / --sec-offset=*offset*	SECURITY hive offset
-y / --sys-offset=*offset*	SYSTEM hive offset
hashdump	decrypt LM and NT hashes
-f / --file=*filename*	memory image file
-s / --sam-offset=*offset*	SAM hive offset
-y / --sys-offset=*offset*	SYSTEM hive offset
lsadump	decrypt LSA secrets
-f / --file=*filename*	memory image file
-s / --sec-offset=*offset*	SECURITY hive offset
-y / --sys-offset=*offset*	SYSTEM hive offset

Malware analysis

orphan_threads	show system threads without module
-f / --file=*filename*	memory image file
-p / --pid=*PID*	System process id (4)
ssdt	enumerate SSDT entries
-f / --file=*filename*	memory image file

Hidden objects

modxview	detect hidden modules
-f / --file=*filename*	memory image file
psxview	detect hidden processes
-f / --file=*filename*	memory image file

Hooks

kernel_hooks	find IAT/EAT/in-line API hooks in kernel space
-f / --file=*filename*	memory image file
usermode_hooks	find IAT/EAT/in-line API hooks in user space
-f / --file=*filename*	memory image file

Interrupts

idt	print Interrupt Descriptor Table (IDT) entries
-f / --file=*filename*	memory image file
intobjscan	scan for interrupt handler registrations
-f / --file=*filename*	memory image file

Volshell

volshell	interactive Volatility environment
-f / --file=*filename*	memory image file
-n / --name=*IMNAME*	select context by name
-o / --offset=*EPROCESS*	select context by EPROCESS (in hex)
-p / --pid=*PID*	select context by PID

Shell commands

hh()	display help
hh(*command*)	display help on command
ps()	list processes
cc()	change context
db()	display BYTEs
dd()	display DWORDs
dt()	display type
list_entry()	traverse a doubly-linked list
quit()	exit Volshell

Imprint

The Volatility Memory Analysis Cheat Sheet was compiled and produced by Andreas Schuster

int for (ensic) {blog;}
http://computer.forensikblog.de/en/
info@forensikblog.de

Version 1.3.1.0
2009-11-01