

**Immunity Debugger Cheat Sheet**

PAUSE	Pause Execution
RUN	Run Program
G [expression]	Run until address
GE [expression]	Pass exception to handler and run until address
SI	Step into
NI	Step over
TI [expression]	Trace into until address
TO [expression]	Trace over until address
TC condition	Trace into until condition
TOC condition	Trace over until condition
TR	Execute until return
TU	Execute until user code
<b>Python Scripts</b>	
!heap	Analyze Process Heaps
!search	Search for instructions in all segments and mem
!searchcode	Search for instructions in .text segments
!safeseh	Locate registered Exception Handlers
!getrpc	Get the Methods for the RPC Interfaces of a single or all loaded DLL's
!gflags	Set Windows Global Flags
!activex	Get the exposed COM Functions for an ActiveX control (requires comtypes library)
!hippie	Heap Allocation Routine hooker

**User Interface**

New Process	F3
Exit Process	Alt+F2
Restore Active Windows	F5
Close IDbg	Alt+X
Breakpoints Window	Alt+B
CPU Window	Alt+C
Loaded Modules Window	Alt+E
Log Window	Alt+L
Memory Window	Alt+M
Embedded Commandline	Alt+F1

**Disassembly Editing**

Label	:
Comment	;
Edit Memory	Ctrl+E
Assemble	Space
Undo Changes	Alt+Backspace

**Execution**

Step Into	F7
Animate Into	Ctrl+F7
Step Over	F8
Animate Over	Ctrl+F8
Run	F9
Pass Exception Back to Program and Run	Shift+F9
Execute until Return	Ctrl+F9
Execute until User Code	Alt+F9
Trace Into	Ctrl+F11
Trace Over	Ctrl+F12
Pause	F12
Pause Trace Conditional	Ctrl+T
Run to Selection	F4

**Breakpoints**

Set/Unset Breakpoint	F2
Set/Edit Conditional Breakpoint	Shift+F2
Set/Edit Conditional Log Breakpoint	Shift+F4
Temporarily Disable/Enable Breakpoint	Space

**Analysis**

Analyze Executable Code	Ctrl+A
Scan Object Files	Ctrl+O
Display Symbolic Names	Ctrl+N

**Searching**

Search for selected address XREFS	Ctrl+R
Search for jumps to selected line	Ctrl+J
Search for sequence	Ctrl+S
Search for binary	Ctrl+B
Search for a command	Ctrl+F
Repeat last search	Ctrl+L

**Code Navigation**

Go to origin	*
Follow Expression/Address	Ctrl+G
Go to previous address	-
Go to next address	+
Go to previous procedure	Ctrl+-
Go to next procedure	Ctrl++
Go to previous reference	Alt+F7
Go to next reference	Alt+F8
Follow jump or call	Enter
View Call Stack	Alt+K

**Immunity Debugger Commandline (Alt+F1) – Commands are not case sensitive**

<b>Expressions</b>	
<b>CALC expression</b>	Calculate value of expression (e.g. calc 2+2)
<b>WATCH expression</b>	Watch expression
<b>Disassembler/Assembler/Comments/Labels</b>	
<b>U expression</b>	Follow address in Disassembler
<b>*</b>	Follow EIP in Disassembler
<b>L expression,label</b>	Assign symbolic label to address
<b>C expression,comment</b>	Set comment at address
<b>A expression[,command]</b>	Assemble at address
<b>Data Dumping</b>	
<b>DD expression</b>	Dump in DWORD format
<b>DW expression</b>	Dump in WORD format
<b>DB expression</b>	Dump in BYTE format
<b>DA expression</b>	Dump in assembler format
<b>DC expression</b>	Dump as ASCII text
<b>DS expression</b>	Dump as addresses in Stack format
<b>DU expression</b>	Dump as UNICODE text
<b>STK expression</b>	Follow address in Stack
<b>Breakpoints</b>	
<b>BP expression[,condition]</b>	Set breakpoint at address
<b>BC expression</b>	Delete breakpoint at address
<b>BPX label</b>	Set breakpoint on all calls to 'label' within the current module
<b>BPD label</b>	Delete breakpoint on all calls to 'label' within the current module
<b>BR expression1[,expression2]</b>	Set memory breakpoint on Read from range
<b>BW expression1[,expression2]</b>	Set memory breakpoint on Write to range
<b>BMD</b>	Remove memory breakpoint
<b>HR expression</b>	Set hardware breakpoint on Read from address
<b>HW expression</b>	Set hardware breakpoint on Write to address
<b>HE expression</b>	Set hardware breakpoint on Execute at address
<b>HD expression</b>	Delete hardware breakpoint at address
<b>Tracing</b>	