PLUGINS	DESCRIPTION
**	**IDENTIFY ROGUE PROCESSES**
pslist	Print all running processes within the EPROCESS doubly linked list
psscan	Scan physical memory for EPROCESS pool allocations
pstree	Print process list as tree showing parent relationships (using EPROCESS linked list)
pstotal	Comparison of **psscan** and **pslist** results. Also, produces output in graphics format
malprocfind	Automatically identify suspicious system processes
processbl	Compare processes and loaded DLLs with a baseline image
**	**ANALYZE PROCESS DLLS AND HANDLES**
dlllist	Print list of loaded dlls for each process
cmdline	Display command-line args for each process
getsids	Print ownership of SIDs for each process
handles	Print list of open handles for each process
mutantscan	Scan memory for mutant objects (kMUTANT)
**	**REVIEW NETWORK ARTIFACTS**
netscan	All the above - scan both connections and sockets [Vista+]
**	**LOOK FOR EVIDENCE OF CODE INJECTION**
malfind	Find hidden and injected code and dump affected memory sections
ldrmodules	Detect unlinked DLLs and image binaries
hollowfind	Attempts to identify evidence of process hollowing
**	**CHECK FOR SIGNS OF HOOKING AND ROOTKIT**
ssdt	Display System Service Descriptor Table entries (*egrep -v '(ntoskrnl|win32k)'*)
psxview	Find hidden processes via cross-view techniques
modscan	Scan memory image to find loaded, unloaded, and unlinked kernel modules
modules	Walk linked list to identify kernel drivers loaded
apihooks	Find DLL function (Inline and Trampoline) hooks. Mainly for Address table and inline API
driverirp	Identify I/O request Packets (IRP) hooks
idt	Identify Interrupt Descriptor Table hooks
**	**DUMP SUSPICIOUS PROCESSES, DRIVERS, DLLS, FILES**
dlldump	Dump DLLs from a process
moddump	Dump a kernel driver to an executable file sample
procdump	Dump a process to an executable file sample
memdump	Dump all addressable memory for a process into one file
cmdscan	Scan for COMMAND_HISTORY buffers (if not found, run strings against process)
consoles	Scan for CONSOLE_INFORMATION output
dumpfiles	Extract files by name or physical offset
filescan	Scan memory for FILE_OBJECTs (Files opened, potentially encrypted, we can dump it)