Sysinternals Tools

The following table correlates System Internals tools with native commands for common tasks or gathering critical information.

Sysinternals Tool	Description	Command Usage	Example	Native tool similarity
sigcheck	Sigcheck is a command-line utility that shows file version number, timestamp information, and digital signature details, including certificate chains	usage: sigcheck [-a][-h][-i][-e][-l][-n][[-s]|[-c|-ct]|[-m]][-q][-r][-u][-vt][-v[r][s]][-f catalog file] <file or directory> usage: sigcheck -d [-c|-ct] <file or directory> usage: sigcheck -o [-vt][-v[r]] <sigcheck csv file> usage: sigcheck -t[u][v] [-i] [-c|-ct] <certificate store name|*>	sigcheck -e c:\windows\system32	wmic datafile where name='c:\\windows\\system32\\notepad.exe'
pslist	pslist is a command line utility that provides process details	PsList -m (details about virtual and physical memory) -d (details about the threads running within processes) -x (dumps process, memory, and thread detail)	C:\> pslist \\admin-pc	tasklist / taskkill SC
Psgetsid	PsGetsid allows you to translate SIDs to their display name and vice versa. It works on builtin accounts, domain accounts, and local accounts	*psgetsid [\\computer[,computer[,...] | @file] [-u username [-p password]]] [account|SID]* *-u* Specifies optional user name for login to remote computer. *-p* Specifies optional password for user name. If you omit this you will be prompted to enter a hidden password. *Account* PsGetSid will report the SID for the specified user account rather than the computer. *SID* PsGetSid will report the account for the specified SID.	psgetsid \\admin-pc	wmic useraccount where (name='uname' and domain='domain_name') get name,sid

		Computer Direct PsGetSid to perform the command on the remote computer or computers specified. *@file* PsGetSid will execute the command on each of the computers listed in the file.		
Listdlls	Listdlls- reports DLLs loaded within a process	listdlls [-r] [-v | -u] [processname|pid] listdlls [-r] [-v] [-d dllname] *processname* Dump DLLs loaded by process (partial name accepted). *pid* Dump DLLs associated with the specified process id. *dllname* Show only processes that have loaded the specified DLL. *-r* Flag DLLs that relocated because they are not loaded at their base address. *-u* Only list unsigned DLLs. *-v* Show DLL version information.	listdlls -v outlook	tasklist /m /fi "imagename eq outlook.exe"
Psloggedon	displays information of who is logged into the system; remotely or locally	psloggedon [-] [-l] [-x] [\\computer | username] *computer* The computer on which the process is running. *-l* Show only local logons instead of both local and network resource logons. *-x* Will not show logon times. *username* Search the network for computers to which that user is logged on.	psloggedon \\admin-pc	query user /server:admin-pc
Psloglist	Psloglist – a tool to parse system event log records	psloglist [-] [\\computer[,computer[,...] *computer* The computer on which the log resides. *-p passwd* Specify a	*psloglist \\admin-pc -h 24 application* (lists all	wevtutil <command> /r:<remote_computer_name> /u:<user_name> /p:<password>

		password for user (Passed as clear text). *-u user* Specify a user name for login to remote computer(optional). *@file* Execute the command on each of the computers listed in the file. *-a* Dump records timestamped after specified date. *-b* Dump records timestamped before specified date. *-c* Clear the event log after displaying. *-d #* Only display records from previous # days. *-e ID* Exclude events with the specified ID or IDs (up to 10). *-f filter* Filter event types with filter string (e.g. "-f w" to filter warnings). *-h #* Only display records from previous # hours. *-i ID* Show only events with the specified ID or IDs (up to 10). *-l event_log_file* Dump records from the specified event log file. -m # Only display records from previous # minutes. *-n #* Only display # number of most recent entries. *-o* event source (Show only records from the specified event source (e.g. \"-o cdrom\"). *-q* event source (Omit records from the specified event source or sources (e.g. \"-q cdrom\"). *-r* Dump log from least	content of application log on admin-pc for the last 24 hours	

		recent to most recent. *-s* Print Event Log records one-per-line, with comma delimited fields. *-w* Wait for new events, dumping them as they generate (local system only). *-x* Dump extended data. *eventlog* application, system or security, only the first few letters need be used.		
psexec	this command executes a process on a remote system Note: File and Printer sharing must be enabled on the remote system for psexec to function	psexec \\computer[,computer[,..] [options] command [arguments] psexec @run_file [options] command [arguments] *command* Name of the program to execute *arguments* Arguments to pass (file paths must be absolute paths on the target system) *-c* Copy the program (command)to the remote system for execution. *-c -f* Copy even if file already exists on the remote system. *-c -v* Copy only if the file is a higher version or is newer than the remote copy. *-d* No wait for the application to terminate. *-e* Do NOT load the specified account’s profile. *-f* Copy the specified program even if the file already exists on the remote system. *-h* Run with the account's elevated token, if	psexec \\admin-pc "c:\Windows\system32\notepad.exe"	WMI: $command = "notepad.exe" $process = [WMICLASS]"\\$System_Name\ROOT\CIMV2:win32_process" $result = $process.Create($command) Powershell: Invoke-Command –ComputerName $ClientName –ScriptBlock {Start-Process notepad.exe}

		available. (Vista or higher) -*i* Interactive - Run the program so that it interacts with the desktop on the remote system. -*p psswd* Specify a password for (sent as clear text). -*r* name of the remote service to create or interact with. -*s* Run remote process in the SYSTEM account -*u user* Specify a user name for login to remote computer. -*w directory* Set the working directory of the process (relative to the remote computer)		