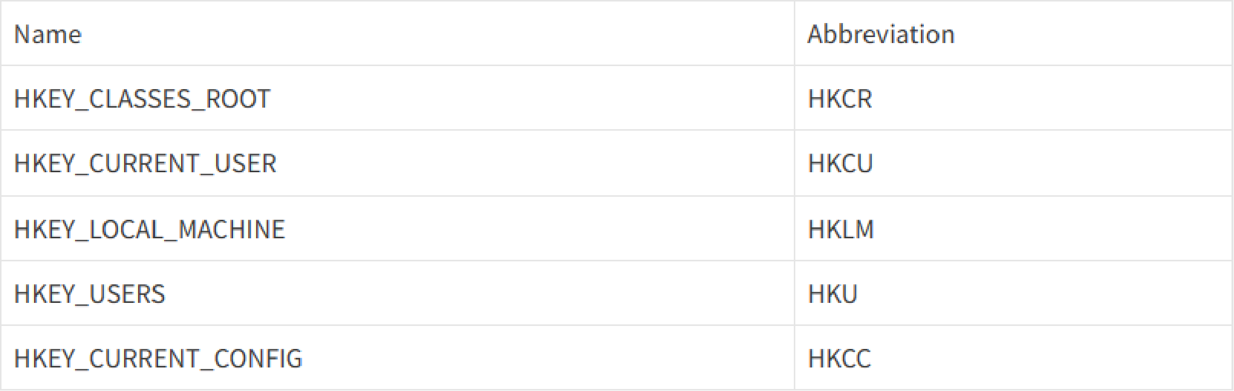
## DFIR Live Incident Response Cheat Sheet

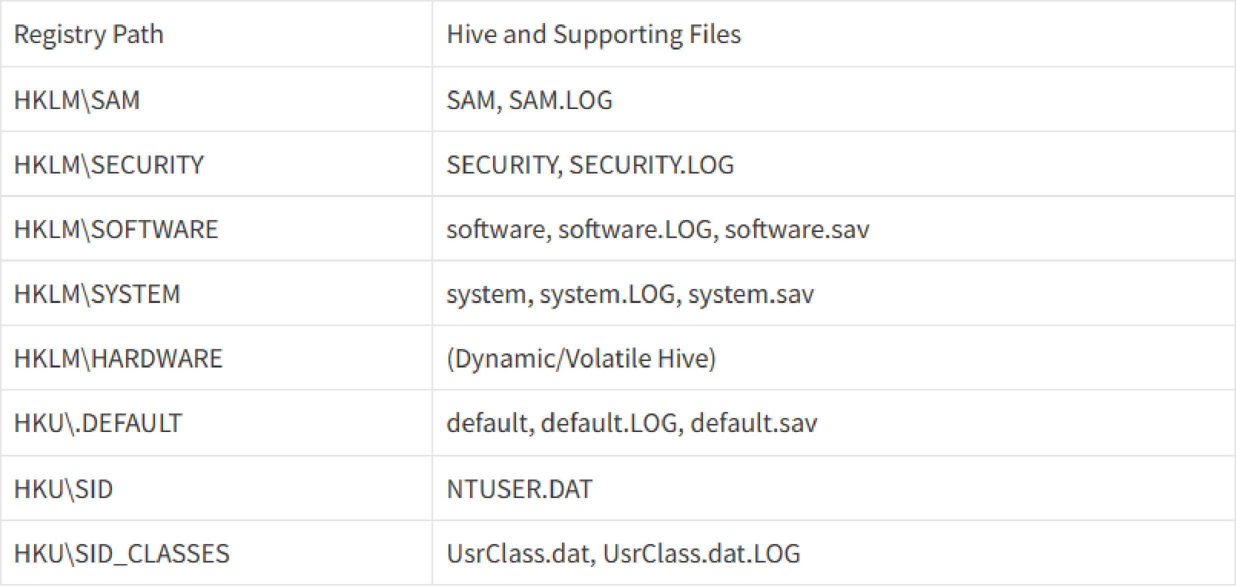
Last Update: March 20, 2024

* Please note that you should ensure the tools that you are using are considering using (e.g., Windows binaries) are safe and have not been modified. Consider running these from a clear USB stick,

### Registry



| Name | Abbreviation |
|---|---|
| HKEY_CLASSES_ROOT | HKCR |
| HKEY_CURRENT_USER | HKCU |
| HKEY_LOCAL_MACHINE | HKLM |
| HKEY_USERS | HKU |
| HKEY_CURRENT_CONFIG | HKCC |



| Registry Path | Hive and Supporting Files |
|---|---|
| HKLM\SAM | SAM, SAM.LOG |
| HKLM\SECURITY | SECURITY, SECURITY.LOG |
| HKLM\SOFTWARE | software, software.LOG, software.sav |
| HKLM\SYSTEM | system, system.LOG, system.sav |
| HKLM\HARDWARE | (Dynamic/Volatile Hive) |
| HKU\.DEFAULT | default, default.LOG, default.sav |
| HKU\SID | NTUSER.DAT |
| HKU\SID_CLASSES | UsrClass.dat, UsrClass.dat.LOG |

### Explorer

- HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer
    - \ComDlg32
    - \LastVistedPidlMRU
    - \OpenSavePidlMRU
    - \RecentDocs
    - \RunMRU

Peter Morin
Petermorin123@gmail.com

- \TypedPaths
- \UserAssist

**Persistence**:

- HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\Run
- HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\RunOnce
- HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\RunOnceEx
- HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\RunServices
- HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\RunServicesOnce

### Shellbags

- HKCU\SOFTWARE\Microsoft\Windows\Shell
    - \BagMRU
    - \Bags

### USB Mass Storage

- HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR < Class ID / Serial #
- HKLM\SYSTEM\CurrentControlSet\Enum\USB < VID / PID
- HKLM\SOFTWARE\Microsoft\Windows Portable Devices\Devices
    - Find Serial # and then look for FriendlyName to obtain the Volume Name of the USB device
- HKLM\SYSTEM\MountedDevices
    - Find Serial # to obtain the Drive Letter of the USB device
    - Find Serial # to obtain the Volume GUID of the USB device

### LNK File Analysis

- C:\username\AppData\Roaming\Microsoft\Windows\Recent
- C:\username\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations
- C:\username\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations

### Prefetcher and SuperFetch

- C:\Windows\Prefetch
- filename-hash(xxxxxxxx).pf
- Example: CALC.EXE-AC08706A.pf

The hash is a hash of the file’s path. In this example, CALC.EXE is located in C:\Windows\System32. If it were copied to another location (like the Desktop) and executed, a new .pf file would be created reflecting a hash of the new path.

### Activities Cache Database

- \Users\<username>\AppData\Local\ConnectedDevicesPlatform\<id>\ActivitiesCache.db.

**AppCompatCache (ShimCache)**

- SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache\

**Common Windows Processes to Investigate**

- Services.exe
- LSASS.exe
- system.exe
- SMSS.exe
- CRSS.exe
- Winlogin.exe
- Winit.exe
- svchost.exe
- Explorer.exe

**Unusual Network Usage**

- Look at File Shares - net view \\127.0.0.1
- Open Sessions with Machine - net session
- Session This machine has Opened - net use
- NetBIOS over TCP/IP Activity - nbtstat -S
- List Listening TCP and UDP Ports - netstat -na
- netstat -na 5
    - 5 - Continuous Scrolling every 5 seconds
- netstat -naob
    - -o flag shows process ID
    - -b flag shows executable
- Inspect Firewall rules
    - netsh advfir-ewall show currentprofile
    - netsh firewall show config

**Unusual Accounts**

- Unexpected Users in the Administtators Group - lusrmgr.msc
- List Users - net user
- List Members of Admin Group - net localgroup administrators
- List Domain Users - net user /domain

When looking at domain accounts, the command will be run on the domain contro-ller. A large domain may take some time - redirect to a text file to analyze:

- net user /domain > domainUsers.txt

Search for Startup Items (persistence)

Users' Autostart Folders

- dir /s /b "-C:-\Doc-uments and Settings\ [user name]\Start Menu\"
- dir /s /b "-C:-\Users\ [user name]\Start Menu\"

Use WMIC To find Start Up Programs

- wmic startup list full

**Search for Unusual Processes**

Task List

- tasklist
- wmic process list full

Parent Process ID

- wmic process get name,p-are-ntp-roc-essid, processid

Command-Line Options and DLLs

- tasklist /m /fi "pid eq [pid]"
- wmic process where proces-sid-=[pid] get commandline

Unusual Scheduled Tasks - schtasks

## Memory (based on Volatility3)

### Identifying the Memory Profile

- vol.py imageinfo -f file.dmp
- vol.py kdbgscan -f file.dmp

### O/S Information

- vol.py -f file.dmp windows.info.Info

### Hashes/Passwords

- vol.py -f file.dmp windows.hashdump.Hashdump #Grab common windows hashes (SAM+SYSTEM)
- vol.py -f file.dmp windows.cachedump.Cachedump #Grab domain cache hashes inside the registry
- vol.py -f file.dmp windows.lsadump.Lsadump #Grab lsa secrets

### Memory Dump

- vol.py -f file.dmp --profile=Win7SP1x86 memdump -p 2168 -D conhost/

### Processes

- vol.py -f file.dmp windows.pstree.PsTree # Get processes tree (not hidden)
- vol.py -f file.dmp windows.pslist.PsList # Get process list (EPROCESS)
- vol.py -f file.dmp windows.psscan.PsScan # Get hidden process list(malware)

### Dump Proc

- vol.py -f file.dmp windows.dumpfiles.DumpFiles --pid <pid> #Dump the .exe and dlls of the process in the current directory

### Command Line Execution

- vol.py -f file.dmp windows.cmdline.CmdLine #Display process command-line arguments

### Environment

- vol.py -f file.dmp windows.envars.Envars [--pid <pid>] #Display process environment variables

### Token Privileges

- vol.py -f file.dmp windows.privileges.Privs [--pid <pid>] #Get enabled privileges of some processes
- vol.py -f file.dmp windows.privileges.Privs | grep "SeImpersonatePrivilege\|SeAssignPrimaryPrivilege\|SeTcbPrivilege\|SeBackupPrivilege\|SeRestorePrivilege\|SeCreateTokenPrivilege\|SeLoadDriverPrivilege\|SeTakeOwnershipPrivilege\|SeDebugPrivilege" #Get all processes with interesting privileges

### SIDS

- vol.py -f file.dmp windows.getsids.GetSIDs [--pid <pid>] #Get SIDs of processes
- vol.py -f file.dmp windows.getservicesids.GetServiceSIDs #Get the SID of services

**Handles**

- vol.py -f file.dmp windows.handles.Handles [--pid <pid>]

**DLLs**

- vol.py -f file.dmp windows.dlllist.DllList [--pid <pid>] #List dlls used by each
- vol.py -f file.dmp windows.dumpfiles.DumpFiles --pid <pid> #Dump the .exe and dlls of the process in the current directory process

**Strings per processes**

What strings belong to what

- strings file.dmp > /tmp/strings.txt
- vol.py -f /tmp/file.dmp windows.strings.Strings --strings-file /tmp/strings.txt

Search strings inside a process

- vol.py -f file.dmp windows.vadyarascan.VadYaraScan --yara-rules "https://" --pid 3692 3840 3976 3312 3084 2784
- vol.py -f file.dmp yarascan.YaraScan --yara-rules https://

**UserAssist**

- vol.py -f file.dmp windows.registry.userassist.UserAssist

**Services**

- vol.py -f file.dmp windows.svcscan.SvcScan #List services
- vol.py -f file.dmp windows.getservicesids.GetServiceSIDs #Get the SID of services

**Network**

- vol.py -f file.dmp windows.netscan.NetScan

**Registry Hive**

List Hives

- vol.py -f file.dmp windows.registry.hivelist.HiveList #List roots
- vol.py -f file.dmp windows.registry.printkey.PrintKey #List roots and get initial subkeys

Get a value

- vol.py -f file.dmp windows.registry.printkey.PrintKey --key "Software\Microsoft\Windows NT\CurrentVersion"

Dump

- vol.py --profile=Win7SP1x86_23418 hivedump -o 0x9aad6148 -f file.dmp #Offset extracted by hivelist #Dump a hive
- vol.py --profile=Win7SP1x86_23418 hivedump -f file.dmp #Dump all hives

### Filesystem

- vol.py --profile=SomeLinux -f file.dmp linux_mount
- vol.py --profile=SomeLinux -f file.dmp linux_recover_filesystem #Dump the entire filesystem (if possible)

### Scan/Dump

- vol.py -f file.dmp windows.filescan.FileScan #Scan for files inside the dump
- vol.py -f file.dmp windows.dumpfiles.DumpFiles --physaddr <0xAAAAA> #Offset from previous command

### Malware

- vol.py -f file.dmp windows.malfind.Malfind [--dump] #Find hidden and injected code, [dump each suspicious section]
- vol.py -f file.dmp windows.driverirp.DriverIrp #Driver IRP hook detection
- vol.py -f file.dmp windows.ssdt.SSDT #Check system call address from unexpected addresses
- vol.py -f file.dmp linux.check_afinfo.Check_afinfo #Verifies the operation function pointers of network protocols
- vol.py -f file.dmp linux.check_creds.Check_creds #Checks if any processes are sharing credential structures
- vol.py -f file.dmp linux.check_idt.Check_idt #Checks if the IDT has been altered
- vol.py -f file.dmp linux.check_syscall.Check_syscall #Check system call table for hooks
- vol.py -f file.dmp linux.check_modules.Check_modules #Compares module list to sysfs info, if available
- vol.py -f file.dmp linux.tty_check.tty_check #Checks tty devices for hooks

### Timeline

- vol.py -f file.dmp timeLiner.TimeLiner