WINDOWS

- Legyakrabban / legutóbb használt fájlok
- Task Manager (Feladatkezelő)
- Sysinternals Suite: Procexplorer, UserAssist
- Event Viewer (Eseménynapló) cmd eventvwr
- Logok: C:\Windows\System32\winevt\Logs
- Hálózati kapcsolatok: cmd netstat -a / -o
- Registry hive- regedit
- Felhasználó könyvtárai: pl. Letöltések, Dokumentumok mappa
- E-mail csatolmányok
- Skype history
- Legutóbb megnyitott fájlok
- Webböngésző letöltések
- TEMP, LOCALTEMP: %systemdrive%\Windows\Temp
  %userprofile%\AppData\Local\Temp
- Felhő tárhely
- Lomtár

Shortcut Files

- Created by user & OS when opening files
- used for Recent Documents
- MAC times and path of original file

.lnkfile

- C:\Users\user\AppData\Roaming\Microsoft...
- -\Windows\Recent
- -\Office\Recent (Vista/Win7/8/10/11)
- Lnkfile can contain malware as a payload.
- Emotetmalware new variant uses PowerShell lnk. with payload as a dropper.
- Zimmerman's LECmd https://ericzimmerman.github.io/#!index.md
- Lecmd.exe –f "C:\path_to_file\file.lnk"

Windows Event Logs

- %systemroot%\System32\winevt\logs\logname.evtx (Vista/7/8/10/11)
- HKLM\System\CurrentControlSet\Services\EvenLog\
- 3 main logs – Security, System, Application
- Ne feledkezzünk meg a naplóesemények korrelációjáról (több naplóból származó információk összerakása)!
- Naplóbeállítások a registry-ben HKLM\SYSTEM\CurrentControlSet\Services\EventLog\Application
- HardwareEvents, Security, System

Windows Registry

LIVE System: Regedit.exe
DEAD System:

- FTK IMAGER (Password recovery and all registry files)
- Autopsy, Magnet Axiom

- Global registry files are typically stored in:
- %SystemRoot%\System32\config

Extract following: NTUSER.DAT; SOFTWARE; SYSTEM; SECURITY; SAM; UsrClass.dat
\Users\<user>\NTUSER.dat
\Windows\System32\config
\Users\<user>\AppData\Local\Microsoft\Windows\UsrClass.dat

NTUSER.dat - Registry Explorer

1. Software\Microsoft\Internet Explorer\TypedURLs
   Megjeleníti a billentyűzeten beírt URL-címeket és a hozzájuk tartozó időbélyegeket.
   Milyen látogatott weboldalakat találunk?
2. Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
   Nemrég megnyitott programok.
   Milyen programokat találunk?
3. Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery
   Legutóbbi keresések.
   Mikre keresett rá?
4. Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist*\Count
   Információ alkalmazásokról, fájlokról, egyéb objektumokról
5. Software\Microsoft\Windows\CurrentVersion\Search\RecentApps
6. SYSTEM\ControlSet001\Control\TimeZoneInformation
   Displays values from Time Zone Information key in a more usable format for timezonebias, etc.
7. SAM\Domains\Account\Users
   Displays user accounts and user account details
8. SOFTWARE \Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
   Displays recently opened documents, by extension
9. SOFTWARE \Microsoft\Office\15.0\Word\User MRU*\File MRU
   Extracts recent Office document names and last opened/closed times
10. SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles
    Lists all wireless or wired networks ever connected
11. SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
    Installed programs
12. SYSTEM\ControlSet01\Services\Tcpip\Parameters\Interfaces
    Network Interfaces and settings

Built-in ([CMD]):

• whoami /all; hostname; ipconfig /all --display currently logged user; hostname; ipconf
• doskey /history --CMD history
• net view :display currently logged user; hostname
• netstat
-b: process nevét megjeleníti
-r: megjeleníti az IP irányító tábla(útválasztó tábla, route print) tartalmát
-n: aktív TCP kapcsolatok, ugyanakkor a címek és portszámok numerikusan kerülnek kijelzésre, semmiképp nem határozza meg azok neveit
-a: megjelenít minden aktív kapcsolatot, illetve azokat a TCP és UDP portokat, amelyeken a számítógép „fülel” (LISTENING)
-o: aktív TCP kapcsolatok és a kapcsolatokhoz tartozó folyamatok ID (PID) számát is

- systeminfo -- displays OS configuration, local or remote
- [CMD] set [variable=[string]] -- displays, sets, or removes system environment variables
- net (accounts, localgroup) -- displays plethora of user, computer and network aspects
- [CMD] doskey /history -- displays command line history (not a system config per se)
- fsmgmt.msc -- opened/mapped shares
- schtasks -- scheduled tasks

System time [CMD]:

- date /T -- displays date
- time /T -- displays time

External:

SysInternals Suite (Z:\V. modul\SysinternalsSuite)

- procexp.exe -- Process Explorer
- tcpvcon.exe; tcpview.exe -- show TCP/UDP endpoints
- PsLoggedon.exe -- show who is logged on (CMD)
- psfile.exe -- lists files and directories opened remotely (CMD)
- pslist.exe -- process information lister (CMD)
- PsService.exe (CMD) -- lists or controls services on a local or remote system
- PsInfo.exe (-s, -d, -h) -- displays installed SW, disk info, hotfixes
- autoruns.exe -- shows programs configured to autostartduring boot
- Procmon.exe -- process monitor utility (regs., network, disk, threads)
- Fport -- (application) to map ports to applications

Windows PowerShell

Windows Management Instrumentation Command-line (WMIC)

- [CMD/PS] wmic [alias] [where clause] [verb clause]
- [CMD/PS] wmic alias list brief

Useful aliases:

- process, service, share, nicconfig, useraccount, qfe, startup, diskdrive, volume, product, netlogin

Remotely:

- [CMD] wmic /node:[targetIPaddr] /user:[User] /password:[Passwd]process list full

Basic system information:

- [CMD/PS] wmic computersystem list brief /format:list

Running services related information.

- [CMD] wmic service where state=”Running” list brief
- [PS] get-service | where {$_.Status -eq "Running"}

List of the running processes in .csv file:

- [CMD/PS] wmic /output: process_list.csv process

Run commands in wmic console and redirect output to all_in_one.txt:

- [CMD/PS] wmic
- [CMD/PS] /record:all_in_one.txt
- [CMD/PS] process list brief
- [CMD/PS] service list brief

PowerShell version:

- [CMD] powershell –command get-host
- [PS] $PSVersionTable

Basic process information (név, ID, elérési út):

- [PS] Get-WmiObject win32_process | select processname, ProcessId, CommandLine

Hotfix info (Windows frissítés):

- [PS] Get-HotFix | selectHotFixID, Description, InstalledOn

Running services related information:

- [PS] get-service | where {$_.Status -eq "Running"}

Active Network Connections and related processes:

- [CMD] netstat –naob
- [PS] netstat -naob| select-string "ESTABLISHED"

Windows event logs:

- [PS] get-eventlogSYSTEM |select -last 10
- [PS] get-eventlogSECURITY |select -last 10

Scheduled jobs:

- [CMD/PS] schtasks

System time zone [PS]:

- Get-TimeZone
- systeminfo.exe

[WIN-CMD] fsutil fsinfo ntfsinfo c:
[WIN-CMD] chkdsk

Skype

- /AppData/Local/Packages/Microsoft.SkypeApp_kzf8qxf38zg5c/LocalState/live#ageorge.lucas_10/skybe.db

LINUX

- 1. Logok: /var/log/auth.log, syslog, mail.log,
- 2. Hálózati kommunikáció: ss, ss -antp
  -a all sockets
  -n numeric, don't resolve service names
  -t tcp, display only TCP sockets
  -p processes, show process using socket (PID)
- 3. Futó folyamatok: ps, ps -aux, pstree, top
  to see every process on the system using BSD syntax
- 4. Kritikus mappák: /tmp, /home, /Downloads, /var/www/html, /var/log/apache2/access.log
  ls -lat időrendben mi módosult
- 5. Ütemezett feladatok: crontab -l, crontab -e

RegRipper

PRINT available plugins

- rip.pl -l
- rip.pl –l |grep NTUSER # OR (SOFWARE,SYSTEM,UsrClass.dat…)
- rip.exe –l |findstrNTUSER

Example: list connected USB devices

- rip.pl –r System –p usbstor(SIFT)

Some other examples:

- rip.pl –r ntuser.dat –p runmru #start-run commands (nemrég megnyitott)

rip.pl –r ntuser.dat –p userassist #application executed
rip.pl –r ntuser.dat –p user_run / run #auto starts after user login (User)
rip.pl -r SAM -p samparse -users details
rip.pl -r SYSTEM -p compname -computer name
rip.pl -r SOFTWARE -p winver -windows version
rip.pl -r ./NTUSER.DAT -p osversion
rip.pl -r SYSTEM -p timezone
rip.pl -r SOFTWARE -p networklist
rip.pl -r SOFTWARE -p networklist | grep "wireless" -B6
rip.pl -r SYSTEM -p networksetup2
rip.pl –r system –p nic2
rip.pl -r software –p networkcards
rip.pl -r system –p shares
rip.pl –r NTUSER.dat –p wc_shares
rip.pl –r software –p uninstall
rip.pl –r software –p apppaths
rip.exe –r system –p shutdown
rip.pl –r ntuser.dat –p wordwheelquery -search history
rip.pl –r NTUSER.DAT –p typedurls
rip.pl -r NTUSER.DAT -p recentdocs
rip.exe –r ntuser.dat –p comdlg32 -dialog boxes executables
rip.exe –r ntuser.dat –p runmru -last commands
rip.exe –r ntuser.dat –p userassist -program runtime, count
rip.exe Software –p run -autorun
rip.pl -r SYSTEM -p bthport -Bluetooth

The Volatility Framework

- Display profiles, plugins, address spaces
  $ vol.py --info
- Global command line options
  $ vol.py -h / --help
- Plugin specific arguments
  $ vol.py [plugin] -h / --help
- e.g: $ vol.py pslist -h
- $ vol.py –f [image] --profile=[PROFILE] [plugin]
- $ vol.py –f [image] --kdbg=[KDBG value] [plugin]
- OS architecture and profile suggestions, DTB –KDBG value info

- $ vol.py –f /path/to/memory.img imageinfo
- Parse the debugger data block(normally faster than imageinfo)
- $ vol.py –f /path/to/memory.img kdbgscan
- TIPP: Rendszerinformációban manuálisan megnézni

Example:

- $ vol.py –f /path/to/memory.img --profile=Win7SP1x64 pslist
- $ vol.py –f /path/to/memory.img --kdbg=0x80545be0 pslist
- $ vol.py –f /path/to/memory.img --profile=Win7SP1x64 pslist
- $ vol.py –f /path/to/memory.img --kdbg= 0x80545be0 pslist
- We can make this command shorter by setting environment variables (use unset command to unset the environment variables):
- export VOLATILITY_LOCATION=file:///path/to/memory.img
- export VOLATILITY_PROFILE=Win7SP1x64
- export VOLATILITY_KDBG= 0x80545be0
- (unset VOLATILITY_LOCATION/PROFILE/KDBG)
- Finally, the command looks like this: vol.py pslist

Rosszindulatú folyamatok azonosítása Plugins:

- pslist -Print all running processes by following the EPROCESS lists
- pstree -Print process list as a tree
- psscan -Pool scanner for process objects
- psxview(--apply-rules) -Find hidden processes with various process listings
- Examples:
- $ python vol.py -f /path/to/memory.img --profile=Win7SP0x64 pslist
- $ ./vol.py -f /path/to/memory.img mac.pstree.PsTree
- XDOT –provides “nicer” visual representation of results:
- $ python vol.py –f /path/to/m2.img <profile> <plugin> --output=dot –output_file=file_name.dot

PROCESS OBJECTS Plugins:

- dlllist -Print list of loaded dlls for each process
- cmdline -Display process command-line arguments
- getsids -Print the SIDs owning each process
- handles -Print list of open handles for each process (Specify handles’ type by handles -t [TYPE] (e.g., File, Mutant etc.)
- mutantscan -Pool scanner for mutex objects (_KMUTANT)

Examples:

- $ python vol.py-f /path/to/memory.img --profile=Win7SP0x64 dlllist
- $ ./vol.py -f /path/to/memory.img windows.cmdline.CmdLine

NETWORK ARTIFACTS Plugins:

- connections -Print list of open connections (x86/x64 WinXP and Win2003 Server)
- connscan -Pool scanner for tcp connections (x86/x64 WinXP and Win2003 Server)
- sockets -Print list of open sockets (x86/x64 WinXP and Win2003 Server)
- sockscan -Pool scanner for tcpsocket objects (x86/x64 WinXP and Win2003 Server)

- netscan -All the above in one plugin (Vista, Win2008 Server, Win7, Win8, Win10)

Examples:

- $ python vol.py -f /path/to/memory.img --profile=Win7SP0x64 connscan
- $ ./vol.py -f /path/to/memory.img windows.netscan.NetScan

CODE INJECTIONS Plugins:

- malfind -Find hidden and injected code
- ldrmodules -Detect unlinked DLLs
- hollowfind -Attempts to identify evidence of process hollowing

Examples:

- $ python vol.py -f /path/to/memory.img --profile=Win7SP0x64 malfind
- $ ./vol.py -f /path/to/memory.img windows.malfind.Malfind

ROOTKITS Plugins:

- Rootkit indicators:
- psxview -Find hidden processes with various process listings
- apihooks -Detect API hooks in process and kernel memory
- ssdt -Display SSDT entries (Eliminate legitimate entries pointing within kernel drivers ntoskrnl.exe and win32k.sys by using I egrep -v‘(ntoskrnl l win32k)’
- modscan -Pool scanner for kernel modules

Examples:

- $ python vol.py -f /path/to/memory.img --profile=Win7SP0x64 psxview
- $ ./vol.py -f /path/to/memory.img windows.ssdt.SSDT

DUMP SUSPICIOUS DATA Plugins:

- filescan -Pool scanner for file objects
- dumpfiles -Extract memory mapped and cached files
- dlldump -Dump DLLs from a process address space
- procdump -Dump a process to an executable file sample
- memdump -Dump the addressable memory for a process
- moddump -Dump a kernel driver to an executable file sample
- cmdscan -Extract command history by scanning for _COMMAND_HISTORY buffers for user typed commands
- consoles -Extract command history by scanning for _CONSOLE_INFORMATION output (Provide what was displayed within the console for applications Iikecmd.exeand PowerShell.exe)
- Volatility v3 uses only dumpfiles – combines plugins above into one
- Examples:
- $ python vol.py -f /path/to/memory.img --profile=Win7SP0x64 procdump–pid=<pid>
- $ ./vol.py -f /path/to/memory.img windows.dumpfiles.DumpFiles–pid<pid>
- $ python vol.py -f /path/to/memory.img --profile=Win7SP0x64 cmdscan

REGISTRY ARTIFACTS Plugins:

- hivelist -Print list of registry hives.
- hivescan -Pool scanner for registry hives
- hivedump -Prints out a hive

- printkey -Print a registry key, and its subkeys
- hashdump -Dumps passwords hashes (LM/NTLM) from memory
- userassist -Print userassistregistry keys and information

Examples:

- $ python vol.py -f /path/to/memory.img --profile=Win7SP0x64 pslist printkey -K "SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN“
- $ ./vol.py -f /path/to/memory.img windows. printkey.PrintKey–key "SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN“
- shimcache -Parses the Application Compatibility Shim Cache registry key
- shimcachemem-Parses the Application Compatibility Shim Cache stored in kernel memory
- amcache -Print AmCacheinformation
- dumpregistry -Dumps registry files out to disk
- autoruns -Searches the registry and memory space for applications running at system startup and maps them to running processes
- mimikatz -Dumps plain text passwords from memory

Examples:

- $ python vol.py -f /path/to/memory.img --profile=Win7SP0x64 mimikatz

TIMELINE

- Memory contains timestamps that can not be found anywhere else:

  Image date Processes Threads
  Event logs IE history Registry entries
  DLL load time MFT Entries Registry keys
- Can be combined with disk timestamps for more powerful results
- Most Windows timestamps are in UTC format Time zone change with –tz flag.

Plugin: timeliner

Examples:

- $ vol.py –f /path/to/memory.img --profile= Win7SP0x64 timeliner --output=body --machine=machine > timeline.txt
- $ vol.py –f /path/to/memory.img --profile= Win7SP0x64 mftparser --output=body --machine=machine > mftparser.txt
- $ vol.py –f /path/to/memory.img --profile= Win7SP0x64 shellbags --output=body --machine=machine > shellbags.txt
- $ cat *.txt > timeline_total.txt
- $mactime –b timeline_total.txt–d –z UTC
  $./vol.py -f /path/to/memory.img timeLiner.TimeLiner

STRINGS

- Additional sense of the functionality of an unknown program ASCII and Unicode What can be found?
- user prompts, error messages, passwords, accessed files, folders, network artifacts strings utility
- a -scan entire file not only the data part
- t -print location of the string in: base 8 (o), base 10 (d) or base 16 (x)

Examples:

- $ strings –a –td [mem.img] > strings.txt(ASCII)

- $ strings –a –td –el [mem.img] >> strings.txt(ASCII + Unicode)
- $ strings.exe–q –o [mem.img] > strings.txt(Windows)

BULK_EXTRACTOR

- Very handy tool:
- Efficient data „combing", Can process any data type, Similar to strings, Puts results into categories (emails, websites, IP addresses, etc.), Creates histograms from results, Can provide more artifacts for investigation

Examples:

- $ bulk_extractor -o output /path/to/memory.img

Command line:

- bulk_extractor[options] –o output_dir[image| -R [dir]]
- -o output directory;
- -f regex to be used;
- -e enables selected scanner (e.g. wordlist, facebook, net);
- -x disables selected scanner;
- -E disables all scanners.

GUI

- BE Viewer
- https://github.com/simsong/bulk_extractor/wiki/BEViewer

Examples:

- $ bulk_extractor -o output /path/to/memory.img

The Sleuth Kit Tools (TSK) Files System tools

File System layer: fsstat - megmutatja a FS részleteit és statisztikáit, beleértve az elrendezést, méreteket és címkéket is

Filename layer:

ffind - megtalálja azokat a kiosztott és ki nem osztott fájlneveket, amelyek egy adott metaadatstruktúrára mutatnak

fls - listázza a kiosztott és törölt fájlneveket egy könyvtárban

Metadata layer:

icat - kivonja a metaadatok címe által meghatározott fájl adategységeit

ifind - megtalálja azt a metaadatstruktúrát, amelynek adategységéhez adat van hozzárendelve, vagy amely egy adott fájlnévvel rendelkezik

ils - megnyitja a megnevezett kép(ek)et és listázza az inode információkat

istat - megjeleníti az összes lemezegységet egy struktúrában és felsorolja az inode információkat

Content (block) layer:

blkstat - megjeleníti az adott adategység kiosztási (allokációs) státuszát

blkls - megnyitja a megnevezett lemezkép(ek)et, és másolja a fájlrendszer-adategységeket

blkcalc - létrehoz egy lemezegységszám-leképezést két lemezkép között, az egyik normál, a másik pedig csak a ki nem osztott (unallocated) egységeket tartalmazza.

Volume tools:

mmls	- megjeleníti a partíciók elrendezését a kötetrendszerben, beleértve a partíciós táblákat és a lemezcímkéket is
mmstats	- megjeleníti a kötetrendszer általános adatait, beleértve a partíciós táblákat és a lemezcímkéket is
mmcat	- egy adott kötet tartalmának kiadása az stdout-ra

Disk tools:

disk_stat	-megmutatja, hogy létezik-e HPA (Host Protected Area)
disk sreset	- ideiglenesen eltávolíthat egy HPA-t, ha van ilyen.

Image File tools:

img_cat	- a kép formátumának részletei
img_stat	- egy képfájl nyers tartalmának megjelenítése

FS Journal tools:

jcat	- egy adott naplóblokk tartalmának megjelenítése
jls	- a fájlrendszer napló bejegyzéseinek felsorolása

- mmls -V -Version
- img_stat [image]
- mmls [image]
- fsstat -o [offset] [image]
- [LINUX] git clone https://github.com/sleuthkit/sleuthkit.git
- [LINUX] sudo fdisk -l
- [LINUX] sudo fdisk -l /dev/sda -u=cylinders (u=sectors)

- Keresse meg az összes *.exe fájlt a " C:\Users\George Lucas\" mappában, a Temp mappában lévő fájlokat kilistázva:
  fls -o 1026048 image.E01 188625 -r -p | grep exe$