ESOF 422:

Advanced Software Engineering: Cyber Practices

Volatility

Reese Pearsall

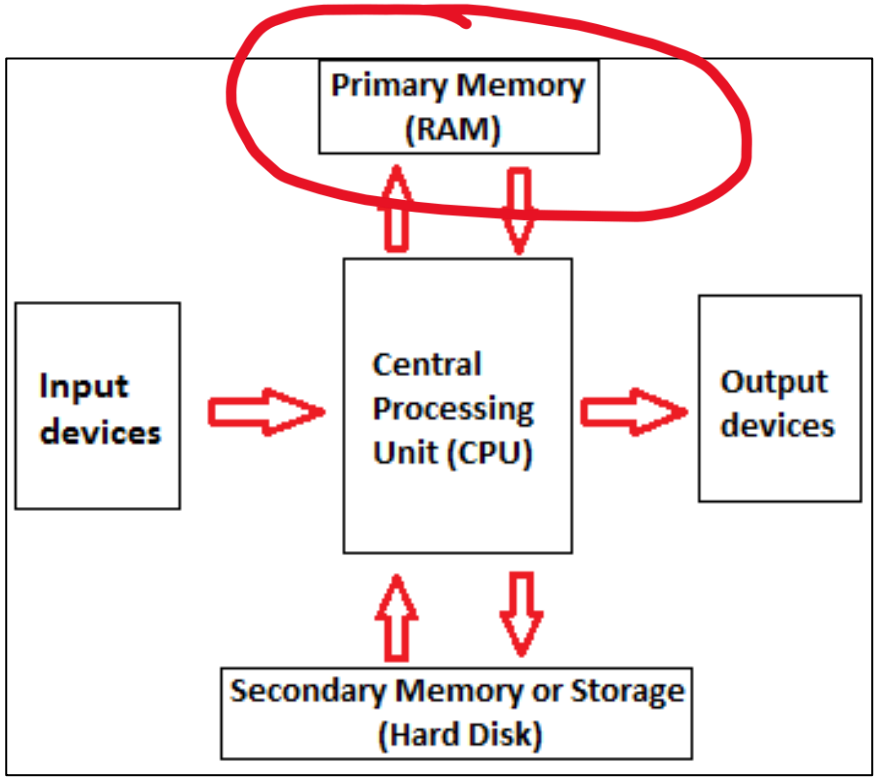
Spring 2025

Memory Forensics

Analysis of data sources from a running system's memory (RAM)

What does RAM contain?

- Programs and files that have been executed
- Running (and sometimes dead) processes
- What programs accessed what files
- Where opens files are/were location on disk
- Information from keyboard (passwords, emails, chats)
- Opened web pages
- Decrypted content
- Network connections
- Content no longer on disk
- Content that was never on disk



Memory Forensics

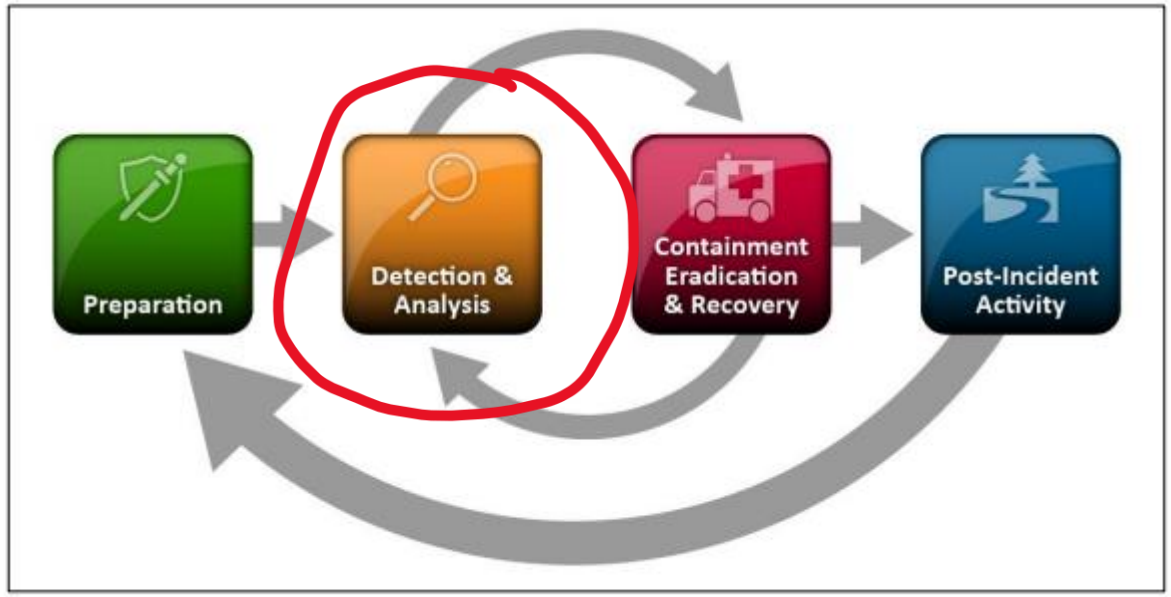


Figure 3-1. Incident Response Life Cycle

The stuff we are talking about for the remainder of the semester are parts of the **analysis** stage

Volatility is a popular, modular framework used for memory forensics

- Written in Python
- Works on memory images from Windows, Mac, and Linux systems
- Runs on Windows, Mac, and Linux
- Open source
- Extensible and scriptable API
- Lots of plugins and community modules

Volatility is not:

- A memory acquisition tool
- Not a GUI
- Bug-free
- Supportive of every single OS version

Installation

You must have access to a Linux VM to install and run volatility.

Kali Linux is great, but it could be any Linux distro

There is volatility2 and volatility3. They are both memory forensics tools.
Volatility3 is newer, and may have slightly different syntax

Kali Linux has many things already installed, but there are some dependencies that we may need to install

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install -y python3 python3-dev libpython3-dev python3-pip python3-setuptools python3-wheel
```

```
┌──(kali㉿kali)-[~]
└─$ python3 -m pip install -U distorm3 yara pycrypto pillow openpyxl ujson pytz ipython capstone
```

The source code for the volatility3 framework will come from a GitHub repo!

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/volatilityfoundation/volatility3.git
```

Memory Files

You will need to get a zip file of a few different memory files you will use for the homework

https://www.cs.montana.edu/pearsall/classes/spring2025/422/homework/hw6.zip

This is around 3GB in size. You will need to have some available space

There is nothing malicious about this file, and the commands that we are running with volatility are safe

```
┌──(kali㉿kali)-[~/hw6]
└─$ ls Lab1/ Lab2/ Lab3/
Lab1/:
desktop.ini  memory.mem

Lab2/:
ecorpoffice

Lab3/:
pos01  target1  target2
```

`.mem` and `.vmem` files are contents of a (virtual) machine's RAM contents

Can be analyzed using volatility

Getting Started

```
┌──(kali㉿kali)-[~]
└─$ python3 ./volatility3/vol.py -f <FILEPATH> -p <PROFILENAME> <pluginname>
```

Getting Started

```
┌──(kali㉿kali)-[~]
└─$ python3 ./volatility3/vol.py -f <FILEPATH> -p <pluginname>
```

vol.py will always be the script we run for volatility commands

Getting Started

```
┌──(kali㉿kali)-[~]
└─$ python3 ./volatility3/vol.py -f <FILEPATH> -p <pluginname>
```

vol.py will always be the script we run for volatility commands

We provide the path to the `.mem` or `.vmem` files for analysis with the –f flag

Getting Started

vol.py will always be the script we run for volatility commands

We provide the path to the `.mem` or `.vmem` files for analysis with the –f flag

The name of the plugin to run

Plugins

Volatility **plugins** are a module or script provided by volatility to extract certain types of data from a memory file

- There are over 200 different plugins, but we will use only a few of them
- May need to provide the machine OS profile with –p flag to get results

```
vol.py: error: argument PLUGIN: invalid choice imageinfo (choose from banners.Banners, configwriter.ConfigWriter, frameworkinfo.FrameworkInfo, isfinfo.IsfInfo, layerwriter.LayerWriter, linux.bas
.Bash, linux.boottime.Boottime, linux.capabilities.Capabilities, linux.check_afinfo.Check_afinfo, linux.check_creds.Check_creds, linux.check_idt.Check_idt, linux.check_modules.Check_modules, lin
x.check_syscall.Check_syscall, linux.ebpf.EBPF, linux.elfs.Elfs, linux.envars.Envars, linux.graphics.fbdev.Fbdev, linux.hidden_modules.Hidden_modules, linux.iomem.IOMem, linux.ip.Addr, linux.ip.
ink, linux.kallsyms.Kallsyms, linux.keyboard_notifiers.Keyboard_notifiers, linux.kmsg.Kmsg, linux.kthreads.Kthreads, linux.library_list.LibraryList, linux.lsmod.Lsmod, linux.lsof.Lsof, linux.mal
ind.Malfind, linux.module_extract.ModuleExtract, linux.modxview.Modxview, linux.mountinfo.MountInfo, linux.netfilter.Netfilter, linux.pagecache.Files, linux.pagecache.InodePages, linux.pagecache
RecoverFs, linux.pidhashtable.PIDHashTable, linux.proc.Maps, linux.psaux.PsAux, linux.pscallstack.PsCallStack, linux.pslist.PsList, linux.psscan.PsScan, linux.pstree.PsTree, linux.ptrace.Ptrace,
linux.sockstat.Sockstat, linux.tracing.ftrace.CheckFtrace, linux.tracing.perf_events.PerfEvents, linux.tracing.tracepoints.CheckTracepoints, linux.tty_check.tty_check, linux.vmaregexscan.VmaRegE
Scan, linux.vmayarascan.VmaYaraScan, linux.vmcoreinfo.VMCoreInfo, mac.bash.Bash, mac.check_syscall.Check_syscall, mac.check_sysctl.Check_sysctl, mac.check_trap_table.Check_trap_table, mac.dmesg.
mesg, mac.ifconfig.Ifconfig, mac.kauth_listeners.Kauth_listeners, mac.kauth_scopes.Kauth_scopes, mac.kevents.Kevents, mac.list_files.List_Files, mac.lsmod.Lsmod, mac.lsof.Lsof, mac.malfind.Malfi
d, mac.mount.Mount, mac.netstat.Netstat, mac.proc_maps.Maps, mac.psaux.Psaux, mac.pslist.PsList, mac.pstree.PsTree, mac.socket_filters.Socket_filters, mac.timers.Timers, mac.trustedbsd.Trustedbs
, mac.vfsevents.VFSevents, regexscan.RegExScan, timeliner.Timeliner, vmscan.Vmscan, windows.amcache.Amcache, windows.bigpools.BigPools, windows.callbacks.Callbacks, windows.cmdline.CmdLine, wind
ws.cmdscan.CmdScan, windows.consoles.Consoles, windows.crashinfo.Crashinfo, windows.debugregisters.DebugRegisters, windows.deskscan.DeskScan, windows.desktops.Desktops, windows.devicetree.Device
ree, windows.direct_system_calls.DirectSystemCalls, windows.dlllist.DllList, windows.driverirp.DriverIrp, windows.drivermodule.DriverModule, windows.driverscan.DriverScan, windows.dumpfiles.Dump
iles, windows.envars.Envars, windows.filescan.FileScan, windows.getservicesids.GetServiceSIDs, windows.getsids.GetSIDs, windows.handles.Handles, windows.hollowprocesses.HollowProcesses, windows.
at.IAT, windows.indirect_system_calls.IndirectSystemCalls, windows.info.Info, windows.joblinks.JobLinks, windows.kpcrs.KPCRs, windows.ldrmodules.LdrModules, windows.malfind.Malfind, windows.mbrs
an.MBRScan, windows.memmap.Memmap, windows.mftscan.ADS, windows.mftscan.MFTScan, windows.mftscan.ResidentData, windows.modscan.ModScan, windows.modules.Modules, windows.mutantscan.MutantScan, wi
dows.netscan.NetScan, windows.netstat.NetStat, windows.orphan_kernel_threads.Threads, windows.pe_symbols.PESymbols, windows.pedump.PEDump, windows.poolscanner.PoolScanner, windows.privileges.Pri
s, windows.processghosting.ProcessGhosting, windows.pslist.PsList, windows.psscan.PsScan, windows.pstree.PsTree, windows.psxview.PsXView, windows.registry.amcache.Amcache, windows.registry.certi
icates.Certificates, windows.registry.getcellroutine.GetCellRoutine, windows.registry.hivelist.HiveList, windows.registry.hivescan.HiveScan, windows.registry.printkey.PrintKey, windows.registry.
cheduled_tasks.ScheduledTasks, windows.registry.userassist.UserAssist, windows.scheduled_tasks.ScheduledTasks, windows.sessions.Sessions, windows.shimcachemem.ShimcacheMem, windows.skeleton_key_
heck.Skeleton_Key_Check, windows.ssdt.SSDT, windows.statistics.Statistics, windows.strings.Strings, windows.suspended_threads.SuspendedThreads, windows.suspicious_threads.SuspiciousThreads, wind
ws.svcdiff.SvcDiff, windows.svclist.SvcList, windows.svcscan.SvcScan, windows.symlinkscan.SymlinkScan, windows.thrdscan.ThrdScan, windows.threads.Threads, windows.timers.Timers, windows.truecryp
.Passphrase, windows.unhooked_system_calls.unhooked_system_calls, windows.unloadedmodules.UnloadedModules, windows.vadinfo.VadInfo, windows.vadregexscan.VadRegExScan, windows.vadwalk.VadWalk, wi
dows.vadyarascan.VadYaraScan, windows.verinfo.VerInfo, windows.virtmap.VirtMap, windows.windows.Windows, windows.windowstations.WindowStations, yarascan.YaraScan)
```

Plugins

Volatility **plugins** are a module or script provided by volatility to extract certain types of data from a memory file

- There are over 200 different plugins, but we will use only a few of them
- For volatility2, you may need to provide the OS profile info w/ the command

Key plugins:

- windows.netscan- lists network connections
- windows.pslist- lists running processes
- windows.filescan- finds file objects in memory
- windows.pstree- shows a process tree
- windows.psscan- finds hidden or terminated processes via memory scanning
- windows.malfind- attempts to find malicious processes
- windows.registry.printkey- prints the value of a registry key
- windows.hashdump- dumps the NT hashes for logged in users
- windows.cmdscan- extracts windows command prompt history

Netscan

Netscan will print out connection information, but also open ports

```
┌──(kali㉿kali)-[~]
└─$ python3 ./volatility3/vol.py -f hw6/Lab1/memory.mem windows.netscan
```

```
Volatility 3 Framework 2.26.2
Progress:  100.00               PDB scanning finished
Offset  Proto   LocalAddr       LocalPort       ForeignAddr     ForeignPort     State   PID     Owner   Created

0xb68cb05a9300  TCPv4   0.0.0.0 49668   0.0.0.0 0       LISTENING       1592    spoolsv.exe     2021-08-06 15:26:34.000000 UTC
0xb68cb05a9300  TCPv6   ::      49668   ::      0       LISTENING       1592    spoolsv.exe     2021-08-06 15:26:34.000000 UTC
0xb68cb0751010  TCPv4   192.168.144.131 80      0.0.0.0 0       LISTENING       508     svchost.exe     2021-08-06 15:26:45.000000 UTC
0xb68cb0766d40  UDPv6   ::1     1900    *       0               3176    svchost.exe     2021-08-06 15:27:19.000000 UTC
0xb68cb079bbd0  UDPv6   ::1     60614   *       0               3176    svchost.exe     2021-08-06 15:27:19.000000 UTC
0xb68cb0a9fec0  UDPv4   0.0.0.0 0       *       0               4344    powershell.exe  2021-08-06 15:54:58.000000 UTC
0xb68cb0f05350  UDPv4   192.168.144.131 138     *       0               4       System  2021-08-06 15:26:24.000000 UTC
0xb68cb1254ec0  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING       824     svchost.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1cd12f0  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING       824     svchost.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1cd12f0  TCPv6   ::      135     ::      0       LISTENING       824     svchost.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1e278a0  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING       568     wininit.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1e278a0  TCPv6   ::      49664   ::      0       LISTENING       568     wininit.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1ec97a0  TCPv4   192.168.144.131 139     0.0.0.0 0       LISTENING       4       System  2021-08-06 15:26:24.000000 UTC
0xb68cb1ed8ec0  UDPv4   0.0.0.0 0       *       0               2136    svchost.exe     2021-08-06 15:26:41.000000 UTC
0xb68cb1ed8ec0  UDPv6   ::      0       *       0               2136    svchost.exe     2021-08-06 15:26:41.000000 UTC
0xb68cb1f099e0  UDPv4   192.168.144.131 137     *       0               4       System  2021-08-06 15:26:24.000000 UTC
0xb68cb1f6e740  TCPv4   127.0.0.1       49676   127.0.0.1       49675   ESTABLISHED     4752    java.exe        2021-08-06 15:30:28.000000 UTC
0xb68cb1fc0dc0  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING       568     wininit.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb22479f0  UDPv4   192.168.144.131 60615   *       0               3176    svchost.exe     2021-08-06 15:27:19.000000 UTC
```

Netscan

We can use grep to scan for just the listening ports (open ports)

```
┌──(kali㉿kali)-[~]
└─$ python3 ./volatility3/vol.py -f hw6/Lab1/memory.mem windows.netscan | grep "LISTENING"
```

```
0xb68cb05a9300.0TCPv4   0.0.0.0 49668can0.0.0.0n0shed   LISTENING       1592    spoolsv.exe     2021-08-06 15:26:34.000000 UTC
0xb68cb05a9300  TCPv6   ::      49668   ::      0       LISTENING       1592    spoolsv.exe     2021-08-06 15:26:34.000000 UTC
0xb68cb0751010  TCPv4   192.168.144.131 80      0.0.0.0 0       LISTENING       508     svchost.exe     2021-08-06 15:26:45.000000 UTC
0xb68cb1254ec0  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING       824     svchost.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1cd12f0  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING       824     svchost.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1cd12f0  TCPv6   ::      135     ::      0       LISTENING       824     svchost.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1e278a0  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING       568     wininit.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1e278a0  TCPv6   ::      49664   ::      0       LISTENING       568     wininit.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb1ec97a0  TCPv4   192.168.144.131 139     0.0.0.0 0       LISTENING       4       System  2021-08-06 15:26:24.000000 UTC
0xb68cb1fc0dc0  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING       568     wininit.exe     2021-08-06 15:26:17.000000 UTC
0xb68cb2344cf0  TCPv4   0.0.0.0 49669   0.0.0.0 0       LISTENING       652     services.exe    2021-08-06 15:26:56.000000 UTC
0xb68cb2344cf0  TCPv6   ::      49669   ::      0       LISTENING       652     services.exe    2021-08-06 15:26:56.000000 UTC
0xb68cb2352330  TCPv4   0.0.0.0 49669   0.0.0.0 0       LISTENING       652     services.exe    2021-08-06 15:26:56.000000 UTC
0xb68cb24aeec0  TCPv4   0.0.0.0 445     0.0.0.0 0       LISTENING       4       System  2021-08-06 15:26:38.000000 UTC
0xb68cb24aeec0  TCPv6   ::      445     ::      0       LISTENING       4       System  2021-08-06 15:26:38.000000 UTC
0xb68cb24e88f0  TCPv4   0.0.0.0 47001   0.0.0.0 0       LISTENING       4       System  2021-08-06 15:26:39.000000 UTC
0xb68cb24e88f0  TCPv6   ::      47001   ::      0       LISTENING       4       System  2021-08-06 15:26:39.000000 UTC
0xb68cb24ff8f0  TCPv4   0.0.0.0 5985    0.0.0.0 0       LISTENING       4       System  2021-08-06 15:26:39.000000 UTC
0xb68cb24ff8f0  TCPv6   ::      5985    ::      0       LISTENING       4       System  2021-08-06 15:26:39.000000 UTC
0xb68cb26862d0  TCPv4   0.0.0.0 49668   0.0.0.0 0       LISTENING       1592    spoolsv.exe     2021-08-06 15:26:34.000000 UTC
0xb68cb268ca30  TCPv4   0.0.0.0 49665   0.0.0.0 0       LISTENING       996     svchost.exe     2021-08-06 15:26:22.000000 UTC
0xb68cb268ca30  TCPv6   ::      49665   ::      0       LISTENING       996     svchost.exe     2021-08-06 15:26:22.000000 UTC
0xb68cb268e9c0  TCPv4   0.0.0.0 49665   0.0.0.0 0       LISTENING       996     svchost.exe     2021-08-06 15:26:22.000000 UTC
0xb68cb26cdcf0  TCPv4   0.0.0.0 49666   0.0.0.0 0       LISTENING       664     lsass.exe       2021-08-06 15:26:26.000000 UTC
0xb68cb26cdcf0  TCPv6   ::      49666   ::      0       LISTENING       664     lsass.exe       2021-08-06 15:26:26.000000 UTC
0xb68cb271e340  TCPv4   0.0.0.0 49666   0.0.0.0 0       LISTENING       664     lsass.exe       2021-08-06 15:26:26.000000 UTC
```

Processes

We can use pslist or pstree to see active processes

```
└─$ python3 ./volatility3/vol.py -f hw6/Lab1/memory.mem windows.pslist
```

The process IDs

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File output
4	0	System	0xb68cb04ac040	113	-	N/A	False	2021-08-06 15:26:02.000000 UTC	N/A	Disabled
292	4	smss.exe	0xb68cb168f800	2	-	N/A	False	2021-08-06 15:26:02.000000 UTC	N/A	Disabled
412	400	csrss.exe	0xb68cb1cb3080	11	-	0	False	2021-08-06 15:26:10.000000 UTC	N/A	Disabled
504	292	smss.exe	0xb68cb1ccf080	0	-	1	False	2021-08-06 15:26:11.000000 UTC	2021-08-06 15:26:11.000000 UTC	Disabled
512	504	csrss.exe	0xb68cb17d9540	12	-	1	False	2021-08-06 15:26:11.000000 UTC	N/A	Disabled
560	504	winlogon.exe	0xb68cb1ea5080	2	-	1	False	2021-08-06 15:26:11.000000 UTC	N/A	Disabled
568	400	wininit.exe	0xb68cb1ea3800	1	-	0	False	2021-08-06 15:26:11.000000 UTC	N/A	Disabled
652	568	services.exe	0xb68cb1efd080	3	-	0	False	2021-08-06 15:26:13.000000 UTC	N/A	Disabled
664	568	lsass.exe	0xb68cb1f2a7c0	8	-	0	False	2021-08-06 15:26:13.000000 UTC	N/A	Disabled
764	652	svchost.exe	0xb68cb1e2d800	14	-	0	False	2021-08-06 15:26:16.000000 UTC	N/A	Disabled
824	652	svchost.exe	0xb68cb1e29800	8	-	0	False	2021-08-06 15:26:17.000000 UTC	N/A	Disabled
912	560	dwm.exe	0xb68cb1ff5080	15	-	1	False	2021-08-06 15:26:18.000000 UTC	N/A	Disabled
972	652	svchost.exe	0xb68cb1fe2800	17	-	0	False	2021-08-06 15:26:19.000000 UTC	N/A	Disabled
996	652	svchost.exe	0xb68cb1fde800	14	-	0	False	2021-08-06 15:26:19.000000 UTC	N/A	Disabled
508	652	svchost.exe	0xb68cb2653500	40	-	0	False	2021-08-06 15:26:21.000000 UTC	N/A	Disabled
400	652	svchost.exe	0xb68cb1fd6800	17	-	0	False	2021-08-06 15:26:21.000000 UTC	N/A	Disabled
944	652	svchost.exe	0xb68cb1fd4800	18	-	0	False	2021-08-06 15:26:22.000000 UTC	N/A	Disabled
1092	652	svchost.exe	0xb68cb2675800	6	-	0	False	2021-08-06 15:26:23.000000 UTC	N/A	Disabled
1100	652	svchost.exe	0xb68cb1fd03c0	23	-	0	False	2021-08-06 15:26:23.000000 UTC	N/A	Disabled
1592	652	spoolsv.exe	0xb68cb272b800	9	-	0	False	2021-08-06 15:26:31.000000 UTC	N/A	Disabled
1668	652	svchost.exe	0xb68cb28a6800	11	-	0	False	2021-08-06 15:26:31.000000 UTC	N/A	Disabled
1732	652	svchost.exe	0xb68cb28c0800	7	-	0	False	2021-08-06 15:26:32.000000 UTC	N/A	Disabled
1764	652	svchost.exe	0xb68cb28bc800	4	-	0	False	2021-08-06 15:26:32.000000 UTC	N/A	Disabled
1852	652	wlms.exe	0xb68cb28ba800	2	-	0	False	2021-08-06 15:26:33.000000 UTC	N/A	Disabled
1876	652	MsMpEng.exe	0xb68cb28b8800	5	-	0	False	2021-08-06 15:26:33.000000 UTC	N/A	Disabled
1896	652	vm3dservice.ex	0xb68cb28b6800	2	-	0	False	2021-08-06 15:26:33.000000 UTC	N/A	Disabled
1904	652	vmtoolsd.exe	0xb68cb28b4800	11	-	0	False	2021-08-06 15:26:33.000000 UTC	N/A	Disabled

Processes

pstree will try to capture the relationships of process (who spawned who)

```
-$ python3 ./volatility3/vol.py -f hw6/Lab1/memory.mem windows.pstree
```

System
- smss.exe
  - smss.exe
    - csrss.exe
      - winlogon.exe

```
PID     PPID    ImageFileName   Offset(V)       Threads
4       0       System  0xb68cb04ac040  113     -
* 292   4       smss.exe        0xb68cb168f800  2
emRoot\System32\smss.exe
** 504  292     smss.exe        0xb68cb1ccf080  0
*** 512 504     csrss.exe       0xb68cb17d9540  12
tDirectory=\Windows SharedSection=1024,20480,768 Windo
ndows\system32\csrss.exe
*** 560 504     winlogon.exe    0xb68cb1ea5080  2
inlogon.exe
**** 912        560     dwm.exe 0xb68cb1ff5080  15
wm.exe
**** 2824       560     fontdrvhost.ex  0xb68cb1ff3080
:\Windows\system32\fontdrvhost.exe
```

If you detect a malicious process, this is very helpful to determine how that malicious process was created!

Processes

You can use the –r flag to **r**ender the output as csv, and save it to an output file (output.csv)

```
└─$ python3 ./volatility3/vol.py -r csv -f hw6/Lab1/memory.mem windows.pstree > output.csv
```

(It may be easier to sift through a CSV file than a terminal output)

File Edit Search View Document Help

```
1 TreeDepth,PID,PPID,ImageFileName,Offset(V),Threads,Handles,SessionId,Wow64,CreateTime,ExitTime,Audit,Cmd,Path
2 0,4,0,System,0×b68cb04ac040,113,-,N/A,False,2021-08-06 15:26:02.000000 UTC,N/A,-,-,-
3 1,292,4,smss.exe,0×b68cb168f800,2,-,N/A,False,2021-08-06 15:26:02.000000 UTC,N/A,\\Device\\HarddiskVolume4\\Windows\\System32\\smss.exe,\\SystemRoot\\System32\\smss.exe,\\SystemRoot\\System32\\smss.exe
4 2,504,292,smss.exe,0×b68cb1ccf080,0,-,1,False,2021-08-06 15:26:11.000000 UTC,2021-08-06 15:26:11.000000 UTC,\\Device\\HarddiskVolume4\\Windows\\System32\\smss.exe,-,-
5 3,512,504,csrss.exe,0×b68cb17d9540,12,-,1,False,2021-08-06 15:26:11.000000 UTC,N/A,\\Device\\HarddiskVolume4\\Windows\\System32\\csrss.exe,"%SystemRoot%\\system32\\csrss.exe ObjectDirectory=\\Windows
  SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=sxssrv,4 ProfileControl=Off MaxRequestThreads=16",C:\\Windows\\system32\
  \csrss.exe
6 3,560,504,winlogon.exe,0×b68cb1ea5080,2,-,1,False,2021-08-06 15:26:11.000000 UTC,N/A,\\Device\\HarddiskVolume4\\Windows\\System32\\winlogon.exe,winlogon.exe,C:\\Windows\\system32\\winlogon.exe
7 4,912,560,dwm.exe,0×b68cb1ff5080,15,-,1,False,2021-08-06 15:26:18.000000 UTC,N/A,\\Device\\HarddiskVolume4\\Windows\\System32\\dwm.exe,""dwm.exe"",C:\\Windows\\system32\\dwm.exe
8 4,2824,560,fontdrvhost.ex,0×b68cb1ff3080,5,-,1,False,2021-08-06 15:35:54.000000 UTC,N/A,\\Device\\HarddiskVolume4\\Windows\\System32\\fontdrvhost.exe,""fontdrvhost.exe"",C:\\Windows\\system32\\fontdrvhost.exe
9 4,1140,560,userinit.exe,0×b68cb2b73280,0,-,1,False,2021-08-06 15:29:16.000000 UTC,2021-08-06 15:29:40.000000 UTC,\\Device\\HarddiskVolume4\\Windows\\System32\\userinit.exe,-,-
10 5,2676,1140,explorer.exe,0×b68cb2d36800,50,-,1,False,2021-08-06 15:29:16.000000 UTC,N/A,\\Device\\HarddiskVolume4\\Windows\\explorer.exe,C:\\Windows\\Explorer.EXE,C:\\Windows\\Explorer.EXE
11 6,2688,2676,mmc.exe,0×b68cb382a5c0,14,-,1,False,2021-08-06 15:56:56.000000 UTC,N/A,\\Device\\HarddiskVolume4\\Windows\\System32\\mmc.exe,""C:\\Windows\\system32\\mmc.exe"" ""C:\\Windows\\system32\\taskschd.msc"" /s",C:
  \\Windows\\system32\\mmc.exe
12 6,4356,2676,cmd.exe,0×b68cb317d340,1,-,1,False,2021-08-06 15:29:59.000000 UTC,N/A,\\Device\\HarddiskVolume4\\Windows\\System32\\cmd.exe,""C:\\Windows\\System32\\cmd.exe"" /C ""C:\\Users\\Administrator\\Desktop\
  \wls1411\\user_projects\\domains\\base_domain\\bin\\startNodeManager.cmd"" ",C:\\Windows\\System32\\cmd.exe
13 7,4456,4356,java.exe,0×b68cb2f21800,16,-,1,False,2021-08-06 15:30:00.000000 UTC,N/A,\\Device\\HarddiskVolume4\\PROGRA~1\\Java\\JDK18~1.0_2\\bin\\java.exe,""C:\\PROGRA~1\\Java\\JDK18~1.0_2\\bin\\java.exe""  -server -
  Xms32m -Xmx200m -Djdk.tls.ephemeralDHKeySize=2048 -Dcoherence.home=C:\\Users\\ADMINI~1\\Desktop\\wls1411\\coherence -Dbea.home=C:\\Users\\ADMINI~1\\Desktop\\wls1411  -Dweblogic.RootDirectory=C:\\Users\\ADMINI~1\
  \Desktop\\wls1411\\USER_P~1\\domains\\BASE_D~1  -Djava.system.class.loader=com.oracle.classloader.weblogic.LaunchClassLoader ""-Djava.security.policy=C:\\Users\\ADMINI~1\\Desktop\\wls1411\\wlserver\\server\\lib\
```

Process Dumping

We can provide a process ID, and the memap plugin will dump the raw contents of the process space (this may take awhile)

```
└─$ python3 ./volatility3/vol.py -f ./hw6/Lab1/memory.mem -o ./dumps/ windows.memmap --dump --pid 4200
```

There will be a lot of data (in hexadecimal) that is dumped. There are several different tools

```
┌──(kali㉿kali)-[~]
└─$ strings dumps/pid.4200.dmp > strings.txt
```

The **`strings`** command can be used to identify possible strings that existed in the process space

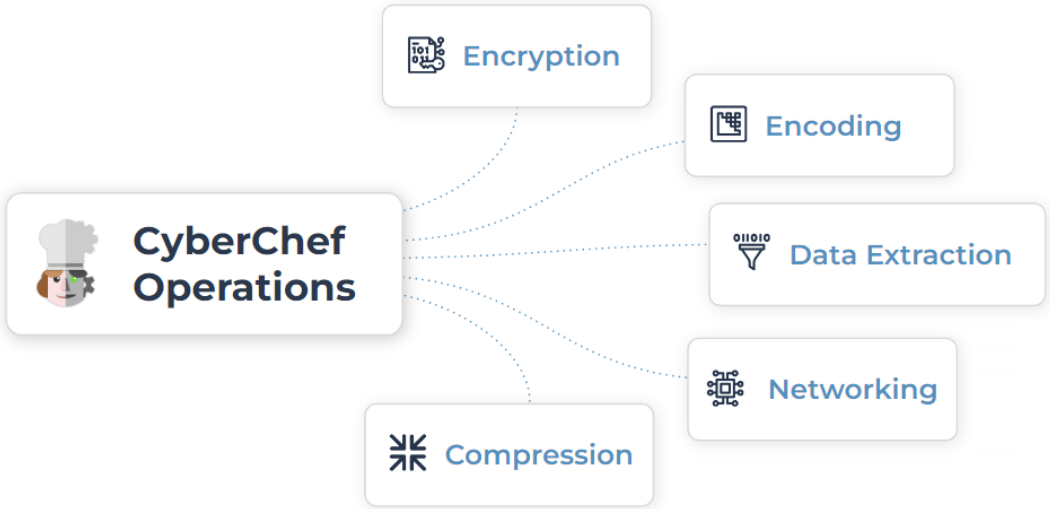
90% of the strings generated will likely be irrelevant, but some might provide some insight!

If a malicious payload is executed, that string should be located somewhere as a String

Code obfuscation

It is common behavior for a threat actor to obfuscate their payload by encoding it in a certain way

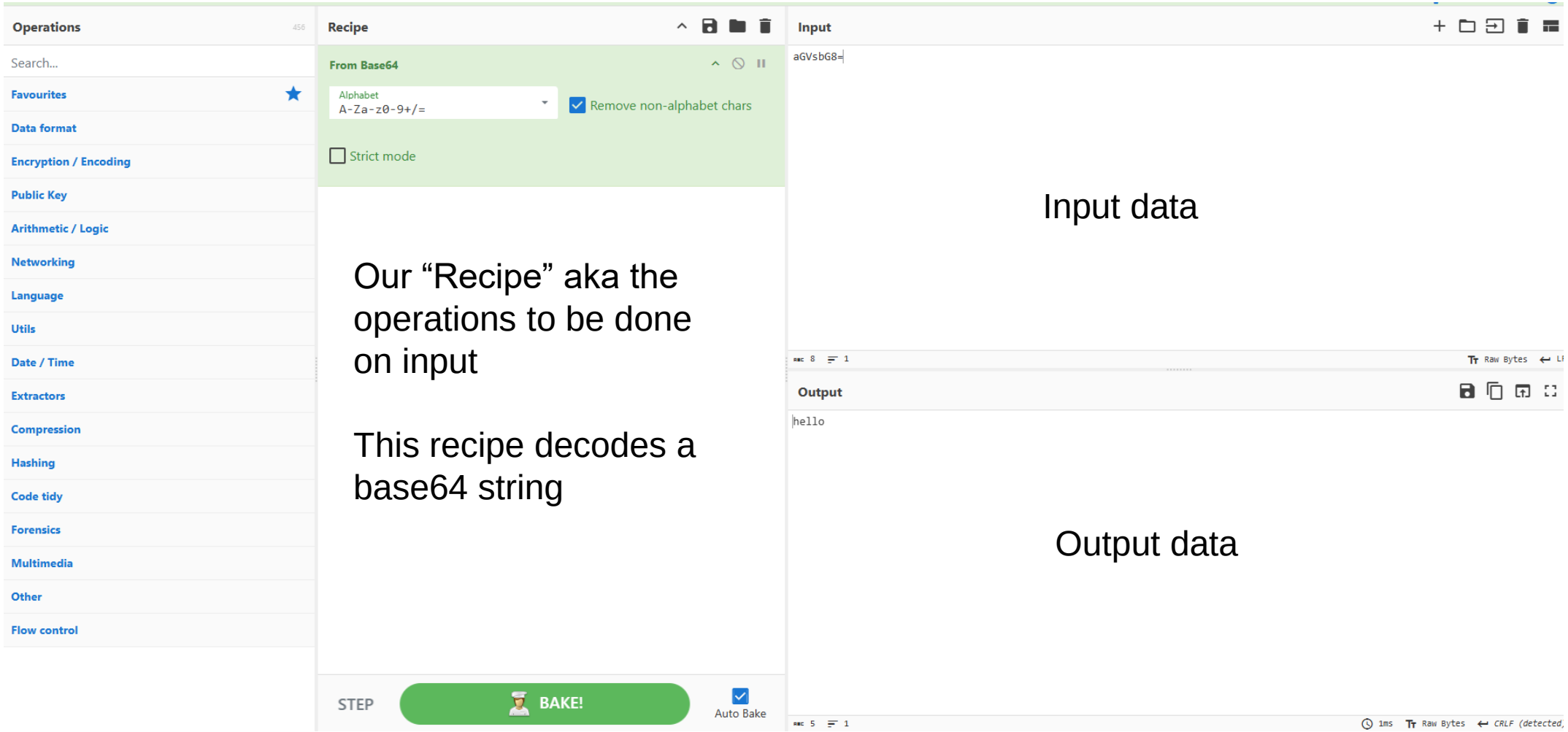
Cyberchef is a “cyber swiss army knife” web app for encryption, encoding, compression, and data analysis



We can plug Strings into cyberchef, apply some “recipes” and see if we can find the original payload

Cyberchef

Cyberchef is a “cyber swiss army knife” web app for encryption, encoding, compression, and data analysis

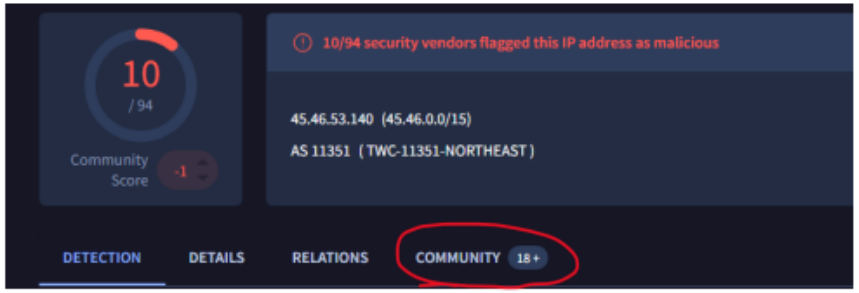


VirusTotal

VirusTotal is a massive database of known malware signatures and malicious fingerprints

We can provide signatures from our investigation to see if they have been flagged as malicious in the past

- File Hashes
- IP Addresses
- Domain Names



The community tab will provide more context around the malicious signatures

Printing Windows Registry Values

```
┌──(kali㉿kali)-[~]
└─$ python ./volatility3/vol.py -f ./hw6/Lab1/memory.mem windows.registry.printkey.PrintKey --key "Microsoft\Windows\CurrentVersion\Run"
```

Software\Microsoft\Windows\CurrentVersion\Run has items that execute when the user logs in

Software\Microsoft\Internet Explorer\TypedURLs has a list of typed URLs

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs shows recently opened documents per file extension

SYSTEM\CurrentControlSet\Control\DeviceClasses shows detailed USB device information

…and so much more. Some keys may not have a value yet