

We and our [partners](#) store and/or access information on a device, such as cookies and process personal data, such as unique identifiers and standard information sent by a device for personalised advertising and content, advertising and content measurement, audience research and services development. With your permission we and our partners may use precise geolocation data and identification through device scanning. You may click to consent to our and our 1444 partners' processing as described above. Alternatively you may click to refuse to consent or access more detailed information and change your preferences before consenting. Please note that some processing of your personal data may not require your consent, but you have a right to object to such processing. Your preferences will apply to this website only. You can change your preferences or withdraw your consent at any time by returning to this site and clicking the "Privacy" button at the bottom of the webpage.

[MORE OPTIONS](#) [DISAGREE](#) [AGREE](#)



[Home](#) > [Software](#) > [Windows Cheat Sheets](#)

Volatility 3.0 Windows Cheat Sheet (DRAFT) by BpDZone

The Volatility Framework is a completely open collection of tools, implemented in Python under the GNU General Public License, for the extraction of digital artifacts from volatile memory (RAM) samples. The extraction techniques are performed completely independent of the system being investigated but offer visibility into the runtime state of the system.

This is a **draft** cheat sheet. It is a work in progress and is not finished yet.

Installation

- 1) Install [Visual Studio C++ build tools](#) (both 64 and 32 bit)
- 2) Clone the latest Volatility version

```
git clone https://github.com/volatilityfoundation/volatility3.git
```
- 3) As of 02.2024 the plugin yara-python is not yet updated so make sure to delete it from requirements.txt before installing.

```
py -m pip install -r requirements.txt
```
- 4) Download symbol tables and put and extract inside "volatility3\symbols":
[Windows](#)
[Mac](#)
[Linux](#)
- 5) Start the installation by entering the following commands in this order.

```
py setup.py build
```

```
py setup.py install
```

 Once the last commands finishes work Volatility will be ready for use.

OS Information

#Show OS & kernel details of the memory sample being analyzed.

```
py vol.py -f "filename" windows.info
```

Hashes

#Dumps user hashes from memory

```
py vol.py -f "filename" windows.hashdump
```

Cache

#Dumps Isa secrets from memory

```
py vol.py -f "filename" windows.cachedump
```

Environment Variables

#Display process environment variables

```
py vol.py -f "filename" windows.envvars.Envars
```

Symlinks

#Scans for links present in a particular windows memory image.

```
py vol.py -f "filename" windows.symlinkscan.SymlinkScan
```

Network

#Scans for network objects present in a particular windows memory image.

```
py vol.py -f "filename" windows.netscan
```

#Traverses network tracking structures present in a particular windows memory image.

```
py vol.py -f "filename" windows.netstat
```

Registry

#Lists the registry hives present in a particular memory image.

```
py vol.py -f "filename" windows.registry.hivelist
```

#Scans for registry hives present in a particular windows memory image.

```
py vol.py -f "filename" windows.registry.hivescan
```

#Lists the registry keys under a hive or specific key value.

```
py vol.py -f "filename" windows.registry.pprintkey.PrintKey --key <KEY>
```

Command line arguments

#Lists process command line arguments.

```
py vol.py -f "filename" windows.cmdline.CmdLine
```

Services

#Lists process token sids.

```
py vol.py -f "filename" windows.getservices.GetServiceSIDs
```

Drivers

#List IRPs for drivers in a particular windows memory image.

```
py vol.py -f "filename" windows.driverirp.DriverIrp
```

#Scans for drivers present in a particular windows memory image.

```
py vol.py -f "filename" windows.driverscan.DriverScan
```

Processes

#Get process list (EPROCESS)

```
py vol.py -f "filename" windows.pslist
```

#Get hidden process list(malware)

```
py vol.py -f "filename" windows.psscanner
```

#Get processes tree (not hidden)

```
py vol.py -f "filename" windows.pstree
```

#Dumps cached file contents from memory samples

```
py vol.py -f "filename" -o "output/dir" windows.dumpfiles --pid <PID>
```

#Prints the memory map

```
py vol.py -f "filename" -o "output/dir" windows.memmap --dump --pid <PID>
```

#Lists process open handles.

```
py vol.py -f "filename" windows.handles --pid <PID>
```

#Lists the loaded modules in a particular windows memory image.

```
py vol.py -f "filename" windows.dlllist --pid <PID>
```

#Lists process token privileges

```
py vol.py -f "filename" windows.privileges.Privs
```

Files

#Scans for file objects present in a particular windows memory image.

```
py vol.py -f "filename" windows.filescan
```

#Dumps cached file contents from Windows memory samples.

```
py vol.py -f -o "output/dir" "filename" windows.dumpfiles
```

Malware General

#Lists process memory ranges that potentially contain injected code.

```
py vol.py -f "filename" windows.malfind.Malfind
```

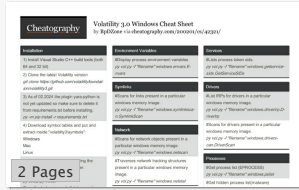
#Lists the system call table.

```
py vol.py -f "filename" windows.ssdt.SSDT
```



Download the Volatility 3.0 Windows

Cheat Sheet



PDF (recommended)

[PDF \(2 pages\)](#)

Alternative Downloads

[PDF \(black and white\)](#)

[LaTeX](#)

Latest Cheat Sheet

DNA as the Genetic Material Cheat Sheet

This cheat sheet summarizes key experiments proving DNA as genetic material, highlighting objectives, methods, results, and conclusions.

2 Pages

★★★★☆ (0)

by [Umeshjagtap](#)

📅 27 Jul 24

🔖 biology, and, and, dna, experiment and 6 more ...

Random Cheat Sheet

C Cheat Sheet

c

📄 pmg

📅 17 Feb 12, updated 1 Mar 20

🔖 development, programming, c, BSD

2 Pages

★★★★★ (11)

About Cheatography

Cheatography is a collection of **6522 cheat sheets** and quick references in **25 languages** for everything from **language** to **business!**

Behind the Scenes

If you have any problems, or just want to say hi, you can find us right here:

- [DaveChild](#)
- [SpaceDuck](#)
- [Cheatography](#)

Recent Cheat Sheet Activity

- [1080000000kmph](#) updated [SEO Tools collection \[Technical SEO edition\]](#).
2 hours 55 mins ago
- [Umeshjagtap](#) published [DNA as the Genetic Material](#).
6 hours 46 mins ago
- [joemefford](#) published [Previs Pro Keyboard Shortcuts](#).
1 day 21 hours ago