Search...

Navigation

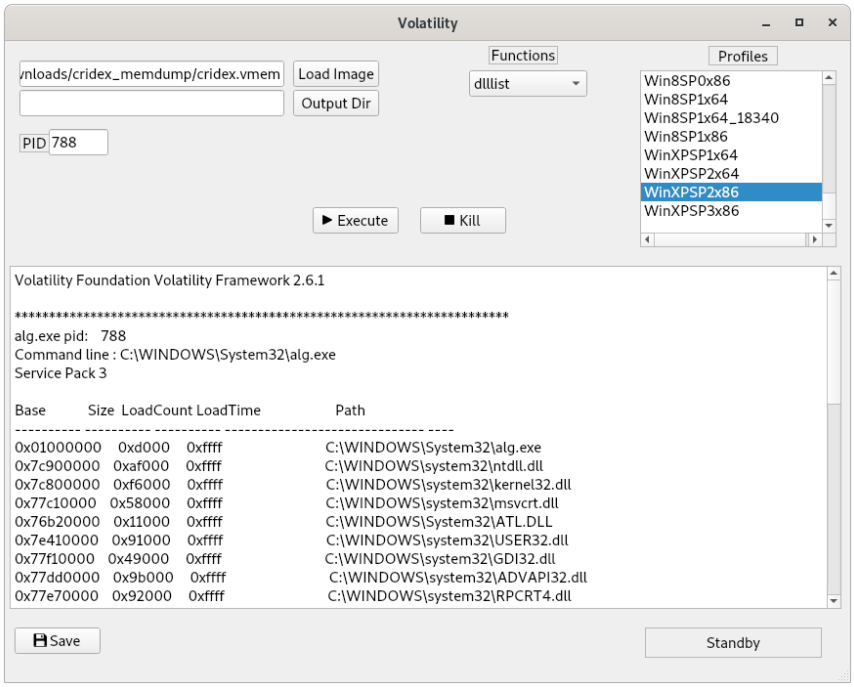
Boosting Cybersecurity with eBPF in Linux
Cuckoo 3 Installation
Volatility-2 Forensics Tool GUI and CheatSheet
Escaping firewall using SSH
Cryptography
Metasploit
Binary to shellcode
IPTables
File Integrity Check
GNU Privacy guard
SSH
TCP Wrapper
NMAP
Step zero

Hamza Megahed / Volatility-2 Forensics Tool GUI and CheatSheet

Volatility-2 Forensics Tool GUI and CheatSheet

Volatility-2 Forensics Tool GUI

This article is about a GUI for Volatility forensics tool written in PyQT5 with cheatsheet for Volatility and you can find the GUI in this URL https://github.com/Hamza-Megahed/volatility-gui



Prerequisites:

1- Installed version of Volatility.

2- Install PyQT5.

```
sudo apt-get install python3-pyqt5
```

3- Download Volatility GUI.

Configuration

From the downloaded Volatility GUI, edit `config.py` file to specify 1- Python 2 bainary name or python 2 absolute path in `python_bin`.

2- Volatility binary absolute path in `volatility_bin_loc`.

Then run `config.py` script to build the profiles list according to your configurations `python3 config.py`

After that start the gui by running `python3 vol_gui.py`.

Volatility-2 CheatSheet

ImageInfo

For a high level summary of the memory sample you're analyzing.

```
vol.py -f <Image_file> imageinfo
vol.py -f <Image_file> kdbgscan
```

pslist

To list the processes of a system.

```
vol.py -f <Image_file> --profile <profile> pslist
```

pstree

To view the process listing in tree form.

```
vol.py -f <Image_file> --profile <profile> pstree
```

psscan

To enumerate processes using pool tag scanning.

```
vol.py -f <Image_file> --profile <profile> psscan
```

dlllist

To display a process's loaded DLLs.

```
vol.py -f <Image_file> --profile <profile> dlllist
-p <PID>
```

dlldump

To extract a DLL from a process's memory space and dump it to disk.
`vol.py -f <Image_file> --profile <profile> dlldump -p <PID> -D <ouput_directory>`

handles

To display the open handles in a process.
`vol.py -f <Image_file> --profile <profile> handles -p <PID>`

cmdscan

Searches the memory of csrss.exe on XP/2003/Vista/2008 and conhost.exe on Windows 7 for commands that attackers entered through a console shell.
`vol.py -f <Image_file> --profile <profile> cmdscan`

envars

To display a process's environment variables.
`vol.py -f <Image_file> --profile <profile> envars`

procdump

To dump a process's executable.
`vol.py -f <Image_file> --profile <profile> procdump -p <PID> -D <ouput_directory>`

evtlogs

Extracts and parses binary event logs from memory.
`vol.py -f <Image_file> --profile <profile> evtlogs -D <ouput_directory>`

modscan

Finds LDR_DATA_TABLE_ENTRY structures by scanning

physical memory for pool tags.

```
vol.py -f <Image_file> --profile <profile> modscan
```

netscan

To scan for network artifacts in 32- and 64-bit Windows Vista, Windows 2008 Server and Windows 7 memory dumps.

```
vol.py -f <Image_file> --profile <profile> netscan
```

hashdump

To extract and decrypt cached domain credentials stored in the registry.

```
vol.py -f <Image_file> hashdump
```

shellbags

Parse and print Shellbag information obtained from the registry.

```
vol.py -f <Image_file> --profile <profile> shellbags
```

shimcache

Parse the Application Compatibility Shim Cache registry key.

```
vol.py -f <Image_file> --profile <profile> shimcache
```

dumpregistry

Allows you to dump a registry hive to disk.

```
vol.py -f <Image_file> --profile <profile> dumpregistry -D <ouput_directory>
```

timeliner

Creates a timeline from various artifacts in memory.

```
vol.py -f <Image_file> timeliner
```

Back to top