

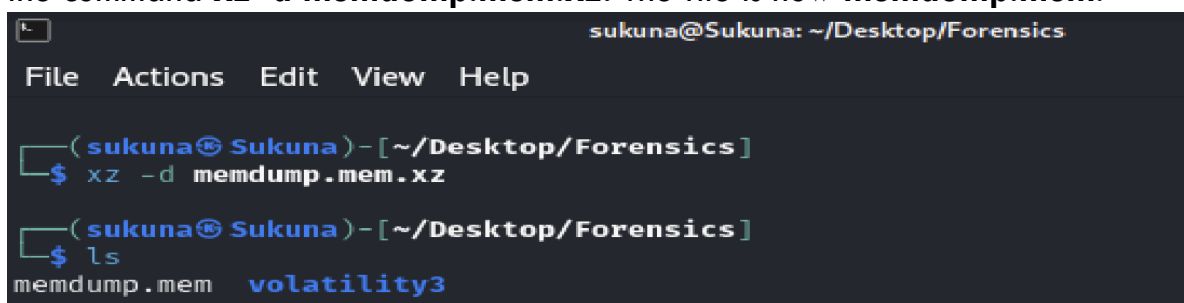
Analyzing a Memory Dump

Objective: We have obtained a live system memory dump from a hacker's computer before it fried itself. The hacker was looking at a suspicious document. Retrieve the lost information.

Operating System: Kali Linux

Software used: volatility3, sql3, hashcat.

Step 1: The memory dump that was taken is compressed using **.xz** compression. Run the command **xz -d memdump.mem.xz**. The file is now **memdump.mem**.

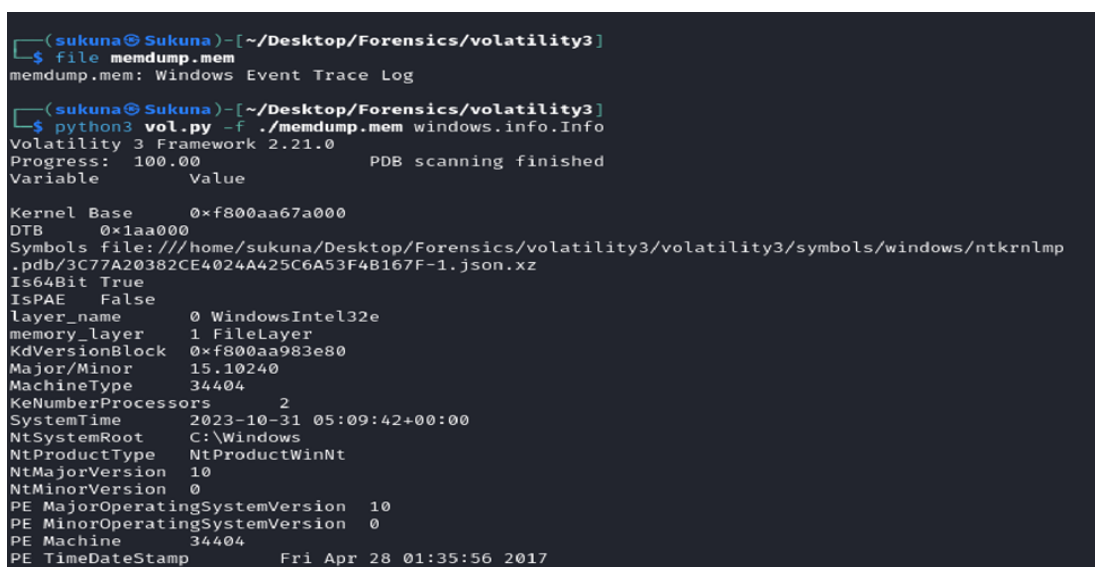


```
sukuna@Sukuna: ~/Desktop/Forensics
File Actions Edit View Help

(sukuna@Sukuna) - [~/Desktop/Forensics]
$ xz -d memdump.mem.xz

(sukuna@Sukuna) - [~/Desktop/Forensics]
$ ls
memdump.mem  volatility3
```

Step 2: Our next step to assist us in our investigation is figure out what OS the memory dump was taken from. We can run the command **file memdump.mem** or **python3 vol.py -f ./memdump.mem windows.info.Info** for a more detailed report.



```
(sukuna@Sukuna) - [~/Desktop/Forensics/volatility3]
$ file memdump.mem
memdump.mem: Windows Event Trace Log

(sukuna@Sukuna) - [~/Desktop/Forensics/volatility3]
$ python3 vol.py -f ./memdump.mem windows.info.Info
Volatility 3 Framework 2.21.0
Progress: 100.00 PDB scanning finished
Variable Value

Kernel Base 0xf800aa67a000
DTB 0x1aa000
Symbols file:///home/sukuna/Desktop/Forensics/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/3C77A20382CE4024A425C6A53F4B167F-1.json.xz
Is64Bit True
IsPAE False
layer_name 0 WindowsIntel32e
memory_layer 1 FileLayer
KdVersionBlock 0xf800aa983e80
Major/Minor 15.10240
MachineType 34404
KeNumberProcessors 2
SystemTime 2023-10-31 05:09:42+00:00
NtSystemRoot C:\Windows
NtProductType NtProductWinNt
NtMajorVersion 10
NtMinorVersion 0
PE MajorOperatingSystemVersion 10
PE MinorOperatingSystemVersion 0
PE Machine 34404
PE TimeDateStamp Fri Apr 28 01:35:56 2017
```

Step 3: Now that we identified that this is a Windows OS, we navigate into the Volatility directory and move our memory dump into that directory. We can run the command **./vol.py -f ./memdump.mem windows.envvars.Envvars | more**.

The `windows.envvars.Envvars` option tells Volatility to extract the Windows environmental variables from the memory dump which contains a the name of the machine that was being used as well as the user of the machine. After searching through report, we find...

```
2352  sihost.exe    0x79baa61300  USERDOMAIN_ROAMINGPROFILE  DESKTOP-0T97GG3
2352  sihost.exe    0x79baa61300  USERNAME                    liber8hacker
2352  sihost.exe    0x79baa61300  USERPROFILE                 C:\Users\liber8hacker
```

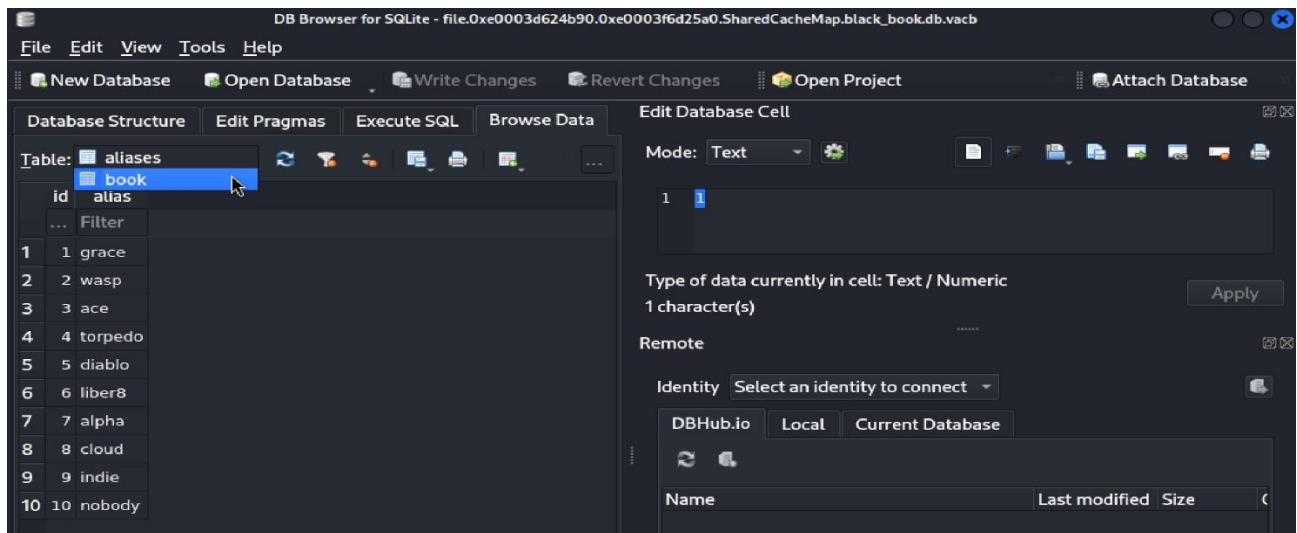
Step 4: Now that we found our user that we are interested, we will want to conduct a file scan to so we can identify any file paths that may interest us. We run the command `./vol.py -f ./memdump.mem windows.filescan.FileScan | more`. We search through and find an interesting file path for our user.

```
0xe0003d4f4440  \Directory
0xe0003d4fa5b0  \Windows\SystemApps\Microsoft.Windows.Cortana_cw5r
0xe0003d4fa860  \Windows\System32\wininit.exe
0xe0003d622f20  \Windows\appcompat\Programs\Amcache.hve.LOG2
0xe0003d623570  \ProgramData\Microsoft\Windows Defender\Definition
A71F84E6}\mpasbase.vdm
0xe0003d623c40  \Windows\appcompat\Programs\Amcache.hve.LOG1
0xe0003d623db0  \Windows\appcompat\Programs\Amcache.hve
0xe0003d624b90  \Users\liber8hacker\Desktop\black_book.db
-- More --
```

Step 5: We can extract the files and contents of the files by running `./vol.py -f ./memdump.mem -o ./dump windows.dumpfiles.DumpFiles --virtaddr 0xe0003d624b90`.

```
(sukuna@Sukuna)-[~/Desktop/Forensics/volatility3]
$ ./vol.py -f ./memdump.mem -o ./dump windows.dumpfiles.DumpFiles --virtaddr 0xe0003d624b90
Volatility 3 Framework 2.21.0
Progress: 100.00 PDB scanning finished
Cache  FileObject  FileName  Result
DataSectionObject  0xe0003d624b90  black_book.db  file.0xe0003d624b90.0xe0003f47b990.DataSectionOb
ject.black_book.db.dat
SharedCacheMap  0xe0003d624b90  black_book.db  file.0xe0003d624b90.0xe0003f6d25a0.SharedCacheMap.black_
book.db.vacb
(sukuna@Sukuna)-[~/Desktop/Forensics/volatility3]
$
```

Step 6: Our file of interest here is a SQL database file. We run the command `sql3 [file path]` to view the database. Searching through our database we can see there are users listed as well as there aliases. This is useful to identify our perp.



Step 7: We can go back to volatility and also extract the users passwords. We run the command `./vol.py -f ./memdump.mem windows.hashdump.Hashdump > hashdump.txt`. This tells Volatility to extract all the NTLM hashes for the users present in the memory dump and `>hashdump.txt` will save it to a text file for easier reference.

```
(sukuna@Sukuna)-[~/Desktop/Forensics/volatility3]
$ ./vol.py -f ./memdump.mem windows.hashdump.Hashdump
Volatility 3 Framework 2.21.0
Progress: 100.00          PDB scanning finished
User    rid    lmhash  nthash
Administrator  500    aad3b435b51404eeaad3b435b51404ee  31d6cfe0d16ae931b73c59d7e0c089c0
Guest         501    aad3b435b51404eeaad3b435b51404ee  31d6cfe0d16ae931b73c59d7e0c089c0
DefaultAccount 503    aad3b435b51404eeaad3b435b51404ee  31d6cfe0d16ae931b73c59d7e0c089c0
liber8hacker   1001   aad3b435b51404eeaad3b435b51404ee  214a7d83f1c36a5f7071137d7c6e5ae6

(sukuna@Sukuna)-[~/Desktop/Forensics/volatility3]
$ ./vol.py -f ./memdump.mem windows.hashdump.Hashdump > hashdump.txt
```

Step 8: Now that we have identified our password hash, we can use password cracking software like **hashcat** to reveal the password. We save the hash in a text file [liber8.txt] and run the command `hashcat -m 1000 -a 0 liber8.txt usr/share/wordlist/rockyou.txt -o lm_hash.txt`. The cracked password is listed in `lm_hash.txt`.

```
(sukuna@Sukuna)-[~/Desktop/Forensics/volatility3]
$ cat lm_hash.txt
214a7d83f1c36a5f7071137d7c6e5ae6:avatar2
```