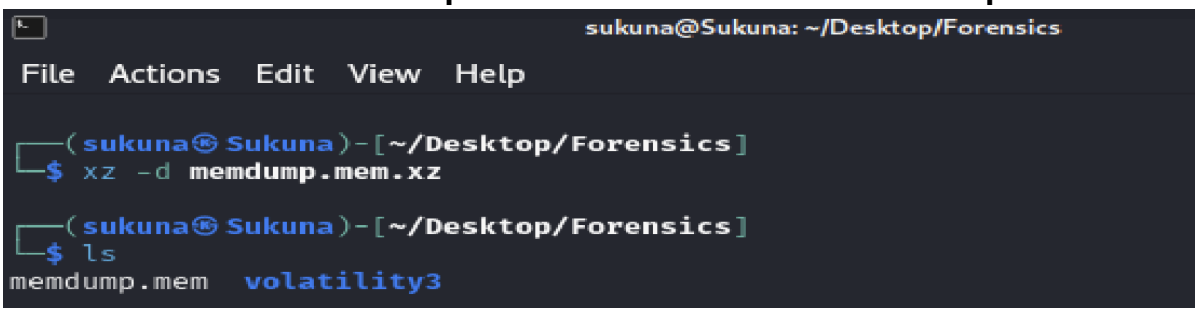
Analyzing a Memory Dump

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.

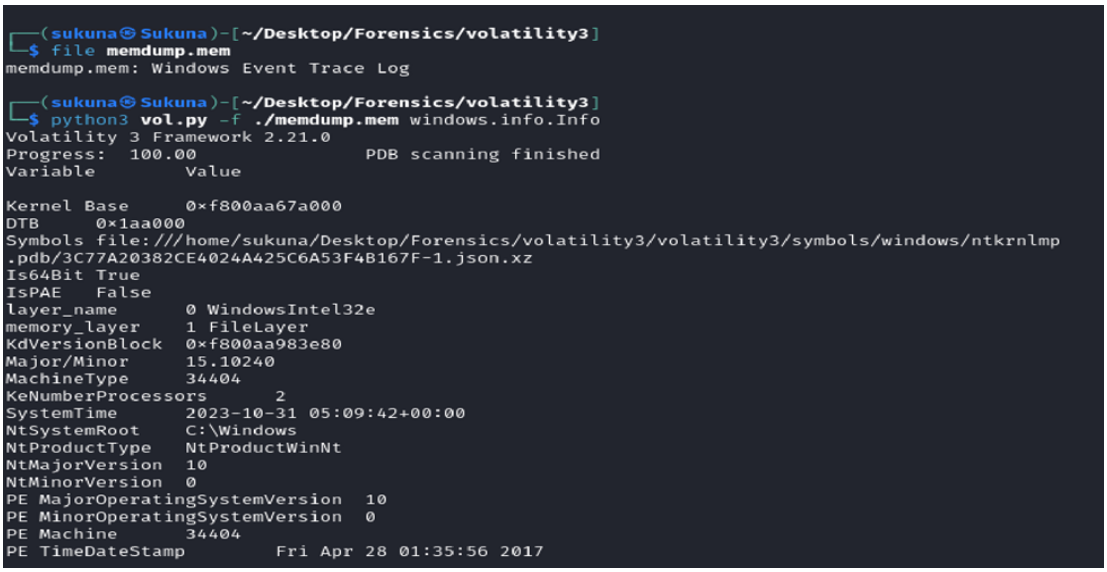
Operating System: Kali Linux

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**.



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.



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.envars.Envars | more.**

The windows.envars.Envars 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
```

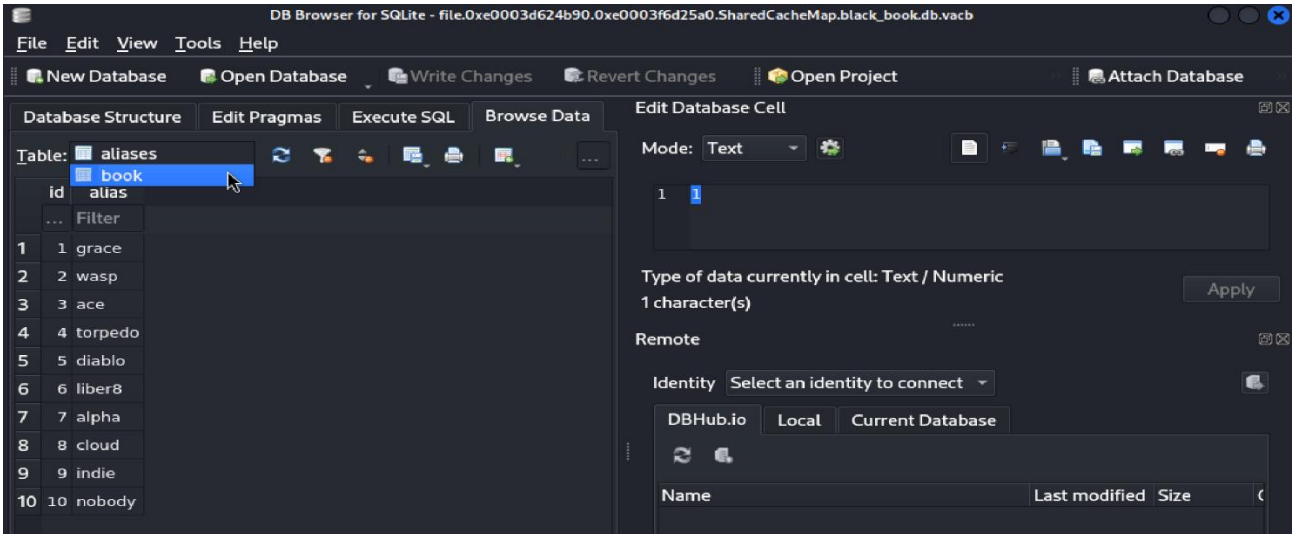
Step 4: Now that we found our user that we are interested, we will want to conduct a file scan to so we can identify any file paths that may interest us. We run the command **./vol.py -f ./memdump.mem windows.filescan.FileScan | more**. We search through and find an interesting file path for our user.

```
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\Definitior
A71F84E6}\mpasbase.vdm
0xe0003d623c40  \Windows\appcompat\Programs\Amcache.hve.LOG1
0xe0003d623db0  \Windows\appcompat\Programs\Amcache.hve
0xe0003d624b90  \Users\liber8hacker\Desktop\black_book.db
--More--
```

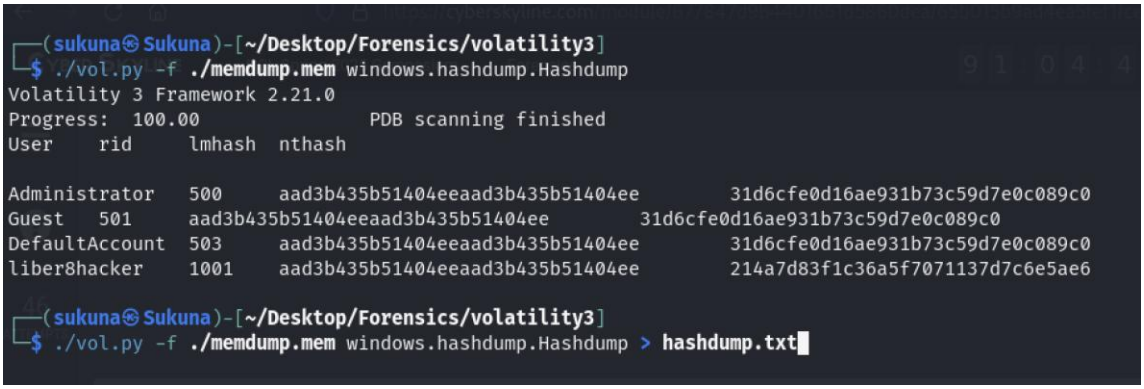
Step 5: We can extract the files and contents of the files by running **./vol.py -f ./memdump.mem -o ./dump windows.dumpfiles.DumpFiles --virtaddr 0xe0003d624b90**.

```
┌──(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.



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.