TDX ARENA

Jordan Knill
Final Assessment Report Submission

Case: Imperial Memory

1/19/2026

Executive Summary

This investigation involved a forensic analysis of a memory dump and an encrypted archive to recover Jules' hidden "secret of success". The process required memory forensics to extract credentials, decrypt an archive, and deep-level document inspection to find a hidden file. The investigation concluded successfully with the identification of a specific MD5 hash value as the final flag.

Findings and Analysis

Finding	Finding Details	Description
User Password	**G6Vmc$Qd5cpM8ee#Ca=x&A3**	**password identified within a PowerShell process memory dump, used to unlock the protected "gift" file.**
Hidden File	**secrets.txt**	**A text file containing philosophical clues that was deliberately hidden inside the internal structure of a .docx document**

Finding	Finding Details	Description
Hash	**0f235385d25ade312a2d151a2cc43865**	**The MD5 hash of the hidden secrets.txt file. The unique "fingerprint" of the file.**

Methodology

Tools and Technologies Used

The following tools were utilized to conduct the investigation and automate the scanning process.

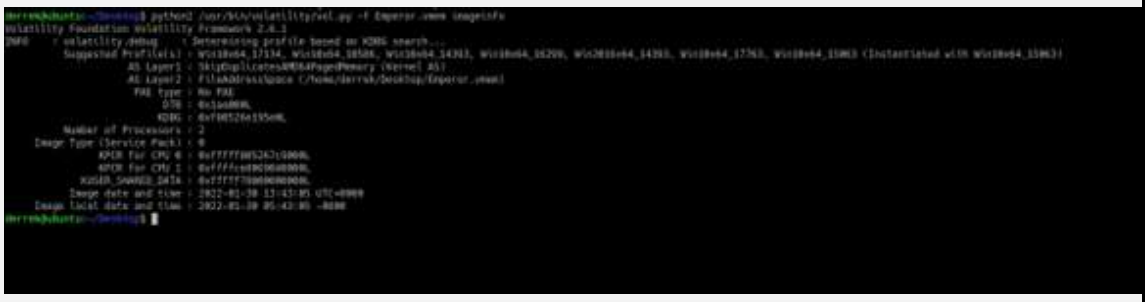
- **Volatility Framework**: Used for advanced memory forensics to analyze the Emperor.vmem file and extract process-specific data.
- **7-Zip**: Employed to decrypt and extract the contents of the password-protected gift.7z archive.
- **Terminal Utilities (unzip, md5sum)**: Used to inspect the internal XML structure of the document and generate cryptographic hashes.

Investigation Process

The investigation was conducted as follows.

1. **Reading the files:** I began by finding a way to read the memory file found on the desktop with Volatility.

```
File  Edit  View  Bookmarks  Settings  Help
derrek@ubuntu:~$ cd ~/Desktop
derrek@ubuntu:~/Desktop$ ls -l
total 1048612
-rw-r--r-- 1 root   root   1073741824 Nov 24  2022 Emperor.vmem
-rwxr-xr-x 1 derrek derrek       5416 Nov 24  2022 chromium.desktop
-rwxr-xr-x 1 derrek derrek       3826 Nov 24  2022 firefox-esr.desktop
-rw-r--r-- 1 root   root        10538 Nov 24  2022 gift.7z
-rwxr-xr-x 1 derrek derrek       9541 Nov 24  2022 org.kde.konsole.desktop
derrek@ubuntu:~/Desktop$
```



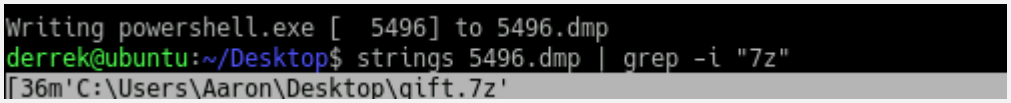
2. **Memory Analysis:** Next I analyzed the memory dump Emperor.vmem using Volatility. By looking at the running processes with python2 /usr/bin/volatility/vol.py -f Emperor.vmem --profile=Win10x64_15063 pslist which revealed the use of PowerShell.

```
0xffffbe0fa5611080 powershell.exe     5496   3180     17        0      1      0 2022-01-30 13:42:29 UTC+0000
0xffffbe0fa5554340 conhost.exe        5516   5496      7        0      1      0 2022-01-30 13:42:29 UTC+0000
```

I specifically targeted the PowerShell process history, as follows

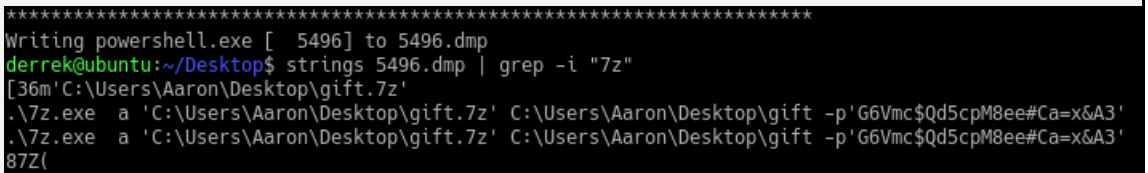
```
derrek@ubuntu:~/Desktop$ python2 /usr/bin/volatility/vol.py -f Emperor.vmem --profile=Win10x64_15063 memdump -p 5496 -D ./
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
Writing powershell.exe [  5496] to 5496.dmp
derrek@ubuntu:~/Desktop$
```

3. **Searching the process dump:** I searched the process dump for powershell for something related to a 7zip file.



```
Writing powershell.exe [  5496] to 5496.dmp
derrek@ubuntu:~/Desktop$ strings 5496.dmp | grep -i "7z"
[36m'C:\Users\Aaron\Desktop\gift.7z'
```

which revealed a clear-text password that unlocked the file archive.



```
************************************************************************
Writing powershell.exe [  5496] to 5496.dmp
derrek@ubuntu:~/Desktop$ strings 5496.dmp | grep -i "7z"
[36m'C:\Users\Aaron\Desktop\gift.7z'
.\7z.exe  a 'C:\Users\Aaron\Desktop\gift.7z' C:\Users\Aaron\Desktop\gift -p'G6Vmc$Qd5cpM8ee#Ca=x&A3'
.\7z.exe  a 'C:\Users\Aaron\Desktop\gift.7z' C:\Users\Aaron\Desktop\gift -p'G6Vmc$Qd5cpM8ee#Ca=x&A3'
87Z(
```

4. **Archive Extraction:** Using the discovered password, I extracted the gift.7z archive found on the user's desktop, which yielded a file named suspicious.docx.

```
derrek@ubuntu:~/Desktop$ ls
5496.dmp  Emperor.vmem  chromium.desktop  firefox-esr.desktop  gift.7z  org.kde.konsole.desktop  suspicious.docx
derrek@ubuntu:~/Desktop$
```

5. **Document Inspection:** The document appeared empty when opened normally. I used the unzip command to treat the .docx file as a container, revealing a hidden secrets.txt file in the root directory.

```
derrek@ubuntu:~/Desktop$ unzip suspicious.docx -d doc_internals
Archive:  suspicious.docx
  inflating: doc_internals/[Content_Types].xml
  inflating: doc_internals/_rels/.rels
  inflating: doc_internals/word/document.xml
  inflating: doc_internals/word/_rels/document.xml.rels
  inflating: doc_internals/word/theme/theme1.xml
  inflating: doc_internals/word/settings.xml
  inflating: doc_internals/word/styles.xml
  inflating: doc_internals/word/webSettings.xml
  inflating: doc_internals/word/fontTable.xml
  inflating: doc_internals/docProps/core.xml
  inflating: doc_internals/docProps/app.xml
  inflating: doc_internals/secrets.txt
derrek@ubuntu:~/Desktop$
```

6. **Final Flag Discovery:** I read secrets.txt, which provided a clue regarding the "Law of Individuality" and "fingerprints." I then generated the MD5 hash of this file to produce the final required flag.

Recommendations

Based on the findings, I am proposing the following recommendations to mitigate the identified risks, secure the systems, and prevent future incidents.

1. **Credential Hygiene:** Ensure that sensitive passwords are not entered into command-line environments where they can be captured in memory dumps or process logs.

2. **Advanced File Scanning:** Implement security tools capable of inspecting compressed files (like .docx or .zip) for hidden or non-standard "out-of-place" files.