Reminiscent

Tags	forensic

Reminiscent – Memory Forensics CTF Writeup

Category: Forensics

Difficulty: Intermediate

Author: Faez

Tools Used: Volatility, strings, CyberChef, base64 decoding, memory analysis

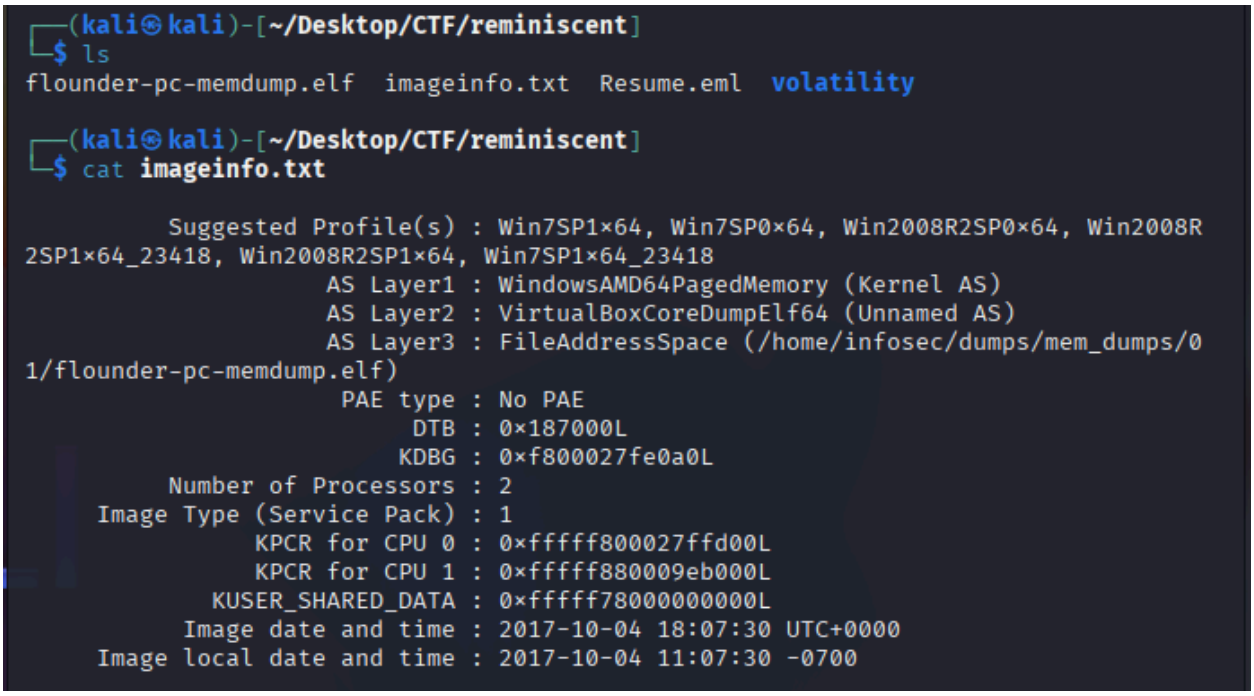
introduction

This writeup walks through the full forensic process, using tools like **Volatility**, **CyberChef**, and **strings** to reconstruct the attack and reveal the adversary's techniques. Whether you're a beginner in digital forensics or sharpening your threat-hunting skills, this challenge offers valuable insights into analyzing fileless malware and reflective PE injection.

Let's dive in.

🗂️ Files Provided:

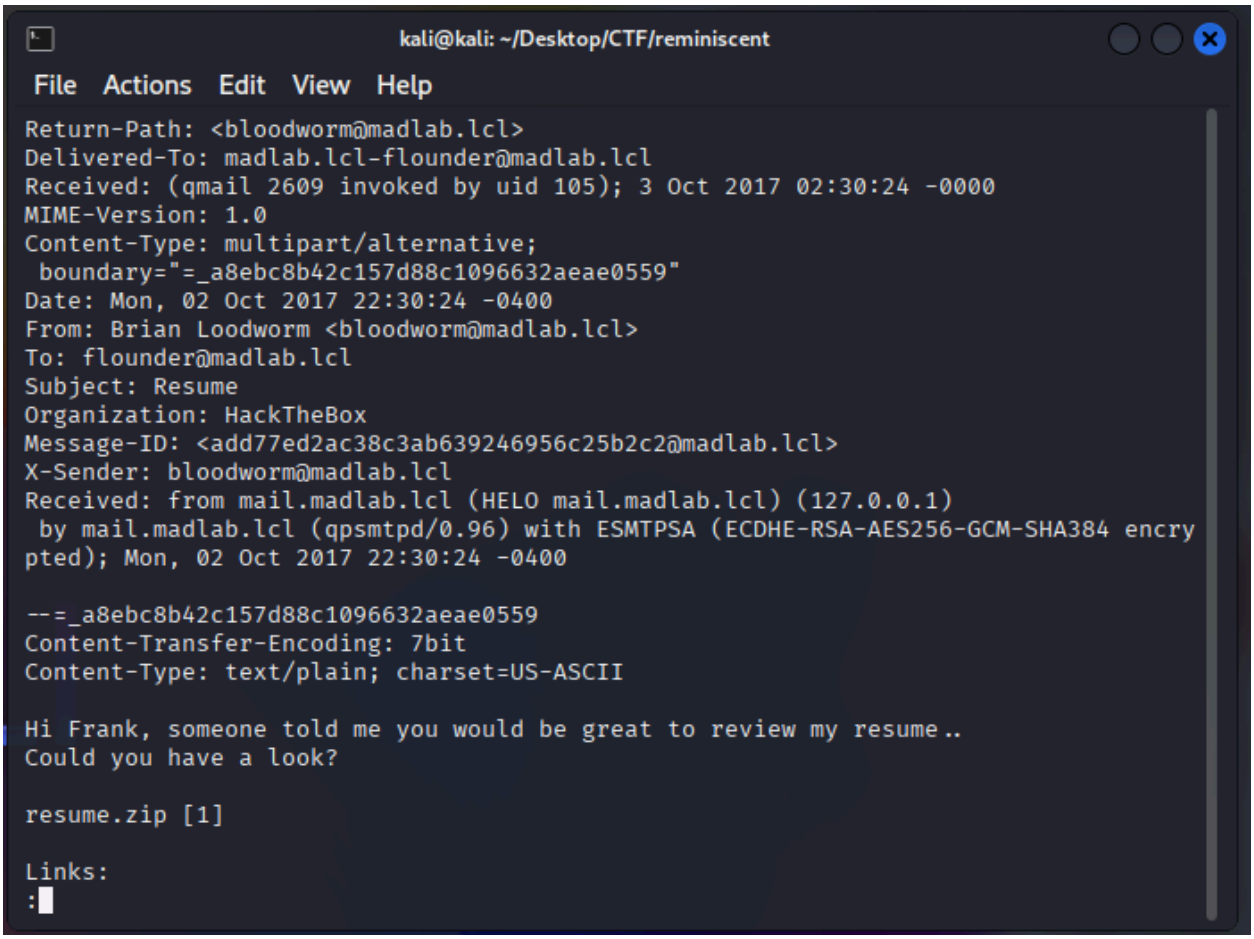
- `flounder-pc-memdump.elf` – Memory dump of the target machine
- `imageinfo.txt` – Suggested Volatility profiles
- `Resume.eml` – Email suspected to be the initial infection vector



then using cat commnad to read imageinfo.txt

`Win7SP1×64_23418`

This profile was used for all subsequent Volatility commands

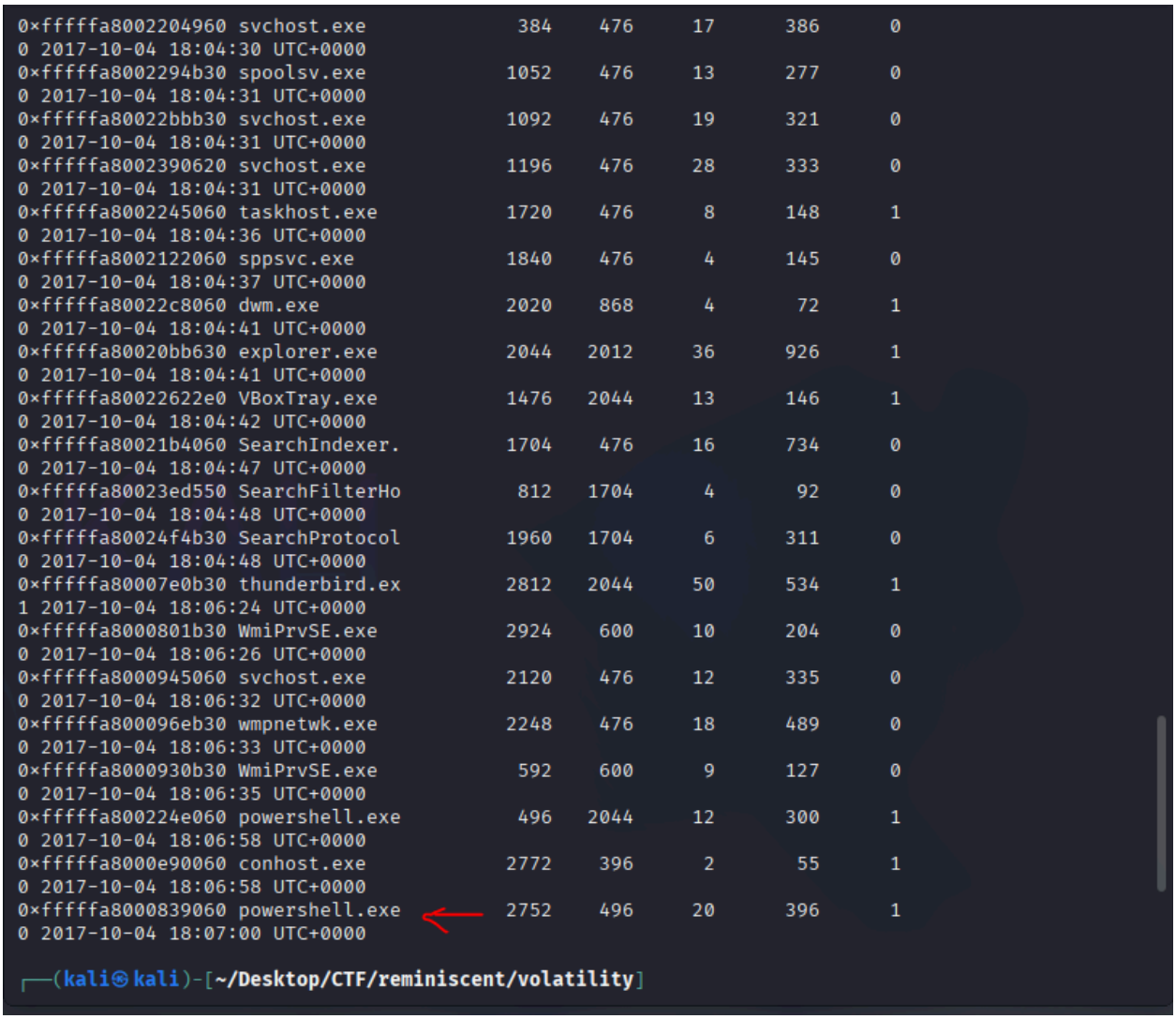


Step 2: Process Tree Analysis

We examined the process tree to identify suspicious activity:

bash

```
python2 vol.py -f flounder-pc-memdump.elf --profile=Win7SP1x64_23418 pstree
```



This chain suggests that Thunderbird (an email client) launched PowerShell — a strong indicator of malicious behavior triggered by an email

Step 4: Dumping PowerShell Memory

We dumped the memory of the second PowerShell process (PID 2752):

bash

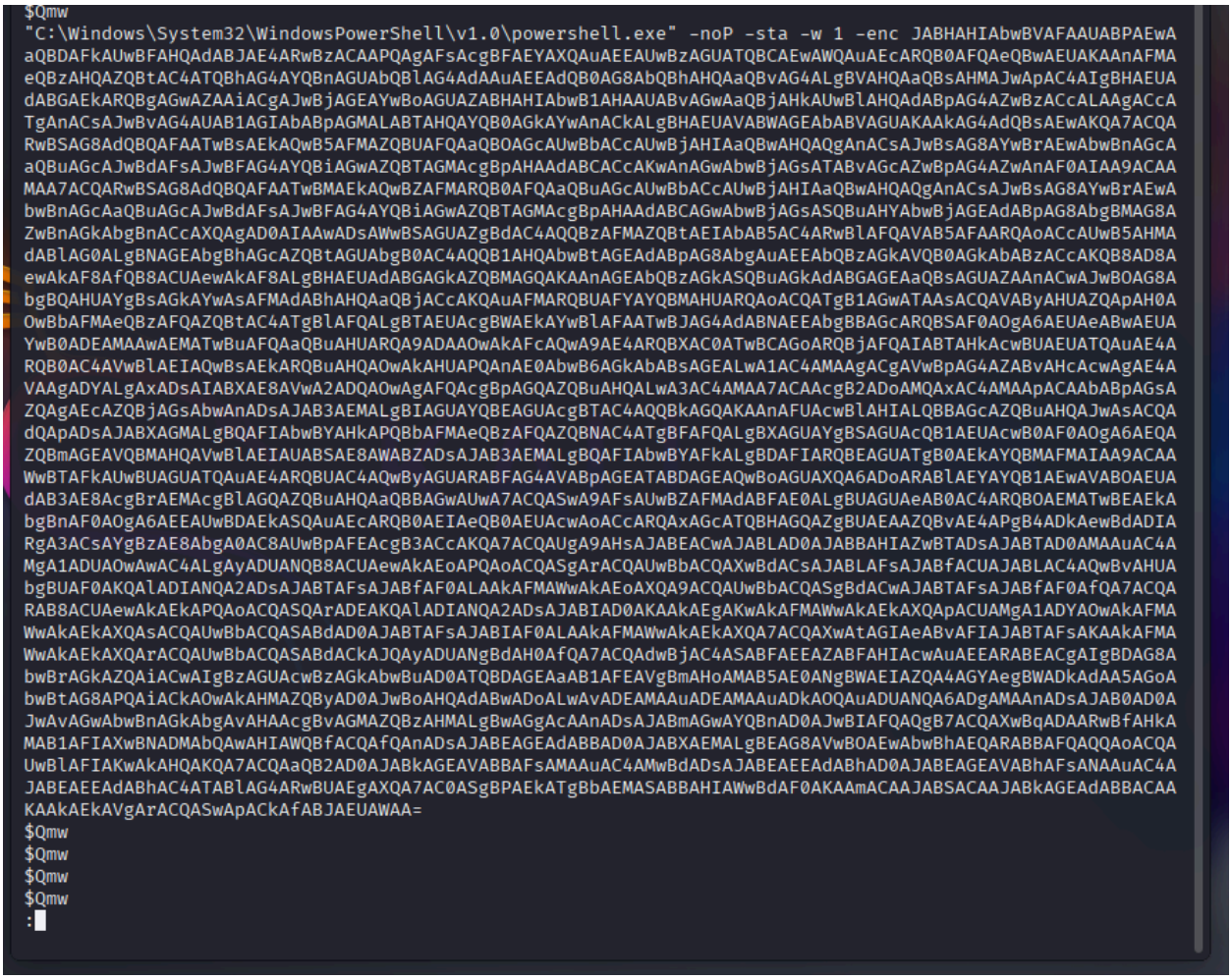
```
python 2 vol.py -f flounder-pc-memdump.elf --profile=Win7SP1×64_23418 memdump -p 2752 -D dumped/
```

```
┌──(kali㉿kali)-[~/Desktop/CTF/reminiscent/volatility]
└─$ mkdir dumped

┌──(kali㉿kali)-[~/Desktop/CTF/reminiscent/volatility]
└─$ python2 vol.py -f flounder-pc-memdump.elf --profile=Win7SP1x64 memdump -p 2752 -D dumped/
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Ha
sh)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defin
ed)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.
Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.
Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.H
ash)
*** Failed to import volatility.plugins.evtlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.H
ash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypt
o.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Has
h)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not define
d)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm
3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Has
h)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named di
storm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash
)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Ha
sh)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto
.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envars (ImportError: No module named Crypto.Hash)
**************************************************************************
Writing powershell.exe [  2752] to 2752.dmp
```

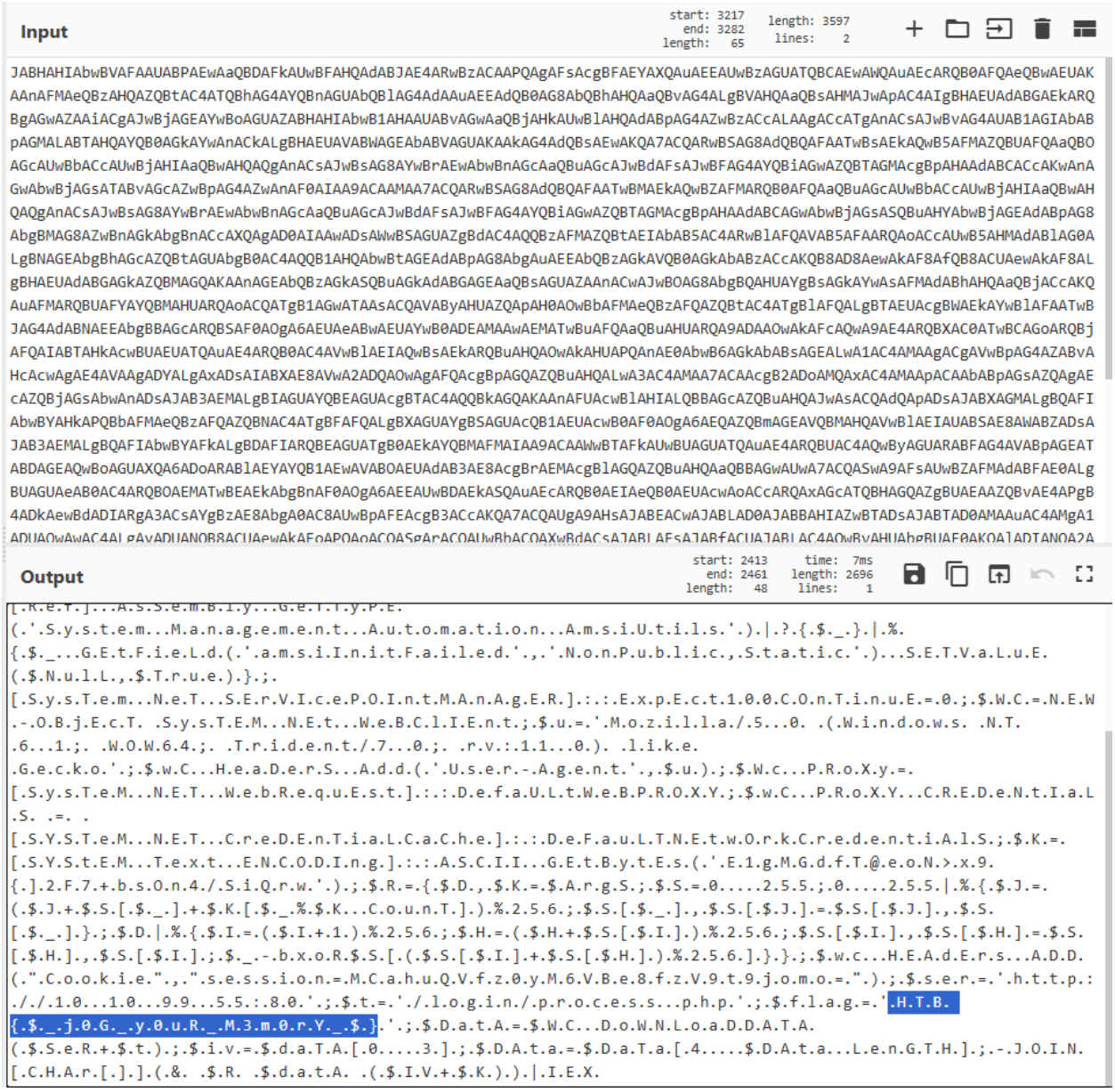
```
┌──(kali㉿kali)-[~/Desktop/CTF/reminiscent/volatility]
└─$
```

```
strings -a dumped/2752.dmp | tee dumped/2752-strings.txt | less
```



Step 5: Decoding the Payload

Inside the strings output, we found a long Base64-encoded PowerShell payload. We decoded it using CyberChef and finally found the flag



HTB{$_j0G_y0uR_M3m0rY_$}