TryHackMe Volatility Investigation using Python

This project demonstrates the use of Volatility, a memory forensics framework, to analyze memory dumps and extract key forensic information as part of incident response tasks. The focus was on identifying suspicious activities, processes, and artifacts in two separate cases using specific Volatility plugins and commands.

Case 001: BOB! THIS ISN'T A HORSE!

Scenario Overview:

A banking trojan masquerading as an Adobe document compromised a quarantined endpoint.

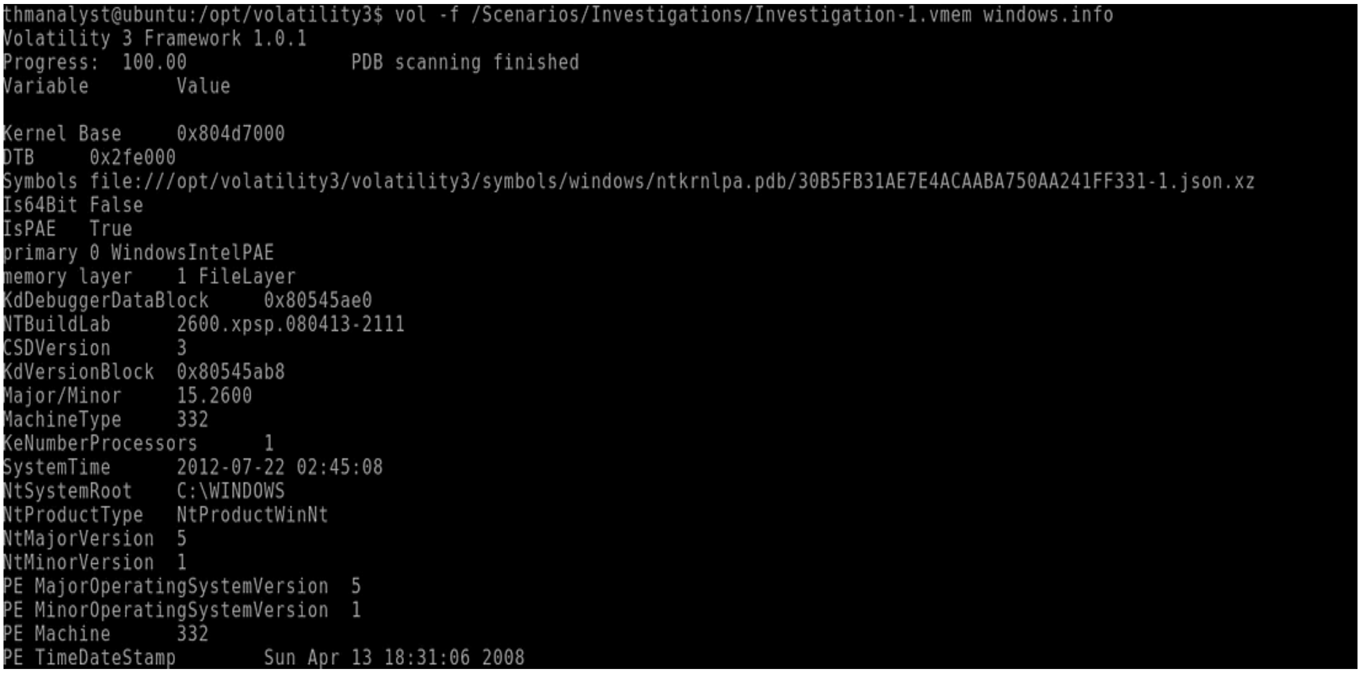
- ○ Memory dump for analysis: `/Scenarios/Investigations/Investigation-1.vmem`.
- ○ Associated suspicious IP: `41.168.5.140`

1. **Exploring Memory Information:**

Identified image information using the command:
Copy code

```
vol -f /Scenarios/Investigations/Investigation-1.vmem windows.info
```

- Extracted build version and timestamp of the memory file.



```
thmanalyst@ubuntu:/opt/volatility3$ vol -f /Scenarios/Investigations/Investigation-1.vmem windows.info
Volatility 3 Framework 1.0.1
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0x804d7000
DTB     0x2fe000
Symbols file:///opt/volatility3/volatility3/symbols/windows/ntkrnlpa.pdb/30B5FB31AE7E4ACAABA750AA241FF331-1.json.xz
Is64Bit False
IsPAE   True
primary 0 WindowsIntelPAE
memory layer    1 FileLayer
KdDebuggerDataBlock     0x80545ae0
NTBuildLab      2600.xpsp.080413-2111
CSDVersion      3
KdVersionBlock  0x80545ab8
Major/Minor     15.2600
MachineType     332
KeNumberProcessors      1
SystemTime      2012-07-22 02:45:08
NtSystemRoot    C:\WINDOWS
NtProductType   NtProductWinNt
NtMajorVersion  5
NtMinorVersion  1
PE MajorOperatingSystemVersion  5
PE MinorOperatingSystemVersion  1
PE Machine      332
PE TimeDateStamp        Sun Apr 13 18:31:06 2008
```

Host Machine Build Version: `2600.xpsp.080413-2111`.

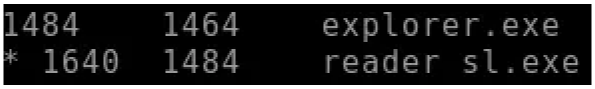
Time of Memory File Acquisition: `2012-07-22 02:45:08`.

2. **Analyzing Processes (Task 10):**

Used `windows.psscan` to list processes and identify suspicious ones:

```
vol -f /Scenarios/Investigations/Investigation-1.vmem windows.psscan
```

- Verified the parent process of the suspicious process using `windows.pstree`.



```
1484     1464     explorer.exe
* 1640   1484     reader sl.exe
```

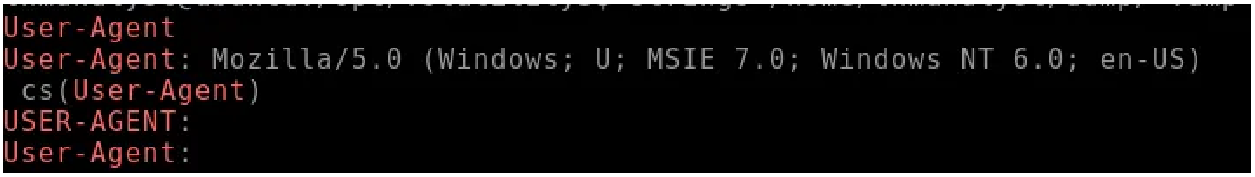
Note: After doing a quick online search, I came to the realization that reader_sl.exe was the one that was suspicious because it's an unnecessary program, and that malware can rename themselves to this. Windows also doesn't need this to function. `Windows.pstree` *tells me that the parent process was explorer.exe (The malware is evading us by hiding in the parent process).*

3. **Investigating Artifacts:**

Dumped memory into current directory using:

```
vol -f /Scenarios/Investigations/Investigation-1.vmem -o
/home/thmanalyst windows.memmap.Memmap --pid 1640 --dump
```

- Searched for specific artifacts using the `strings /home/thmanalyst/*.dmp | grep -i "user-agent"`command combined with `grep` to discover a user-agent.



```
User-Agent
User-Agent: Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 6.0; en-US)
 cs(User-Agent)
USER-AGENT:
User-Agent:
```

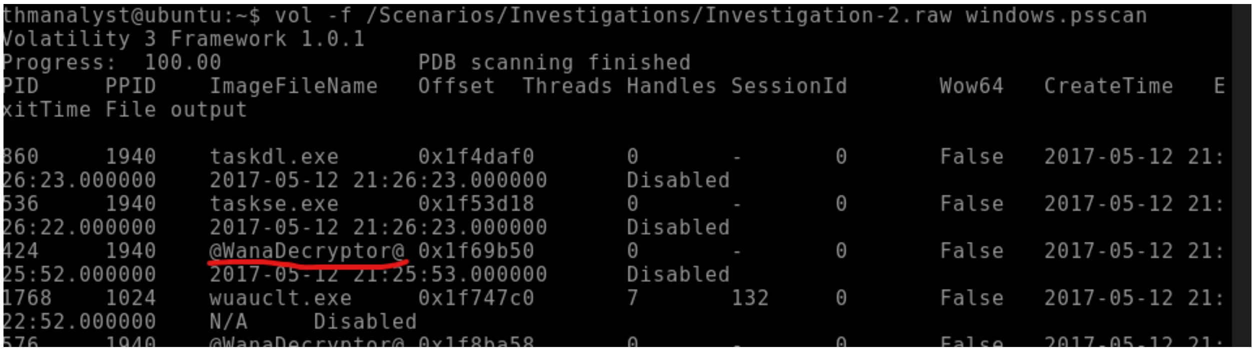
4. **Cross-Referencing Information:**
   - Used plugins like `windows.dlllist` and `windows.handles` to identify DLLs, mutexes, and other malware indicators.
   - Searched for paths and parent process details using targeted filters and lookups.

Case 002: That Kind of Hurt My Feelings

Scenario Overview:

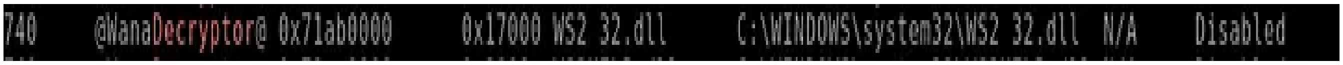
- **Corporation hit by international ransomware chain.**
- **Recovery completed; decryption key used.**
- **Post-incident analysis required to identify actors and events.**
- **Memory dump for analysis:** **`/Scenarios/Investigations/Investigation-2.raw`.**

- Conducted similar analysis using `windows.psscan`, `windows.pstree`, `windows.dlllist`, and `windows.handles` for a second memory dump file (`Investigation-2.raw`).
- Identified the malware as WannaCry, and extracted related indicators such as mutexes (MsWinZonesCacheCounterMutexA) and DLLs (WS2_32.dll).



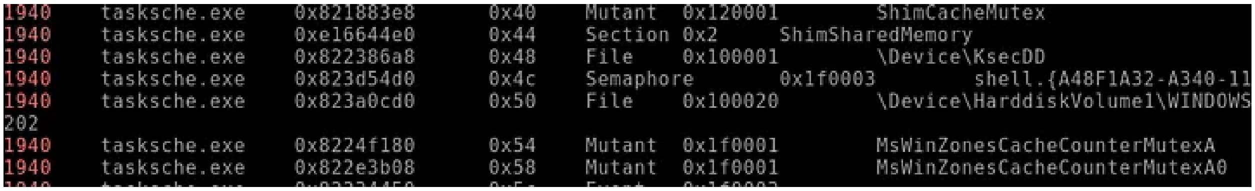
```
thmanalyst@ubuntu:~$ vol -f /Scenarios/Investigations/Investigation-2.raw windows.psscan
Volatility 3 Framework 1.0.1
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset  Threads Handles SessionId       Wow64   CreateTime   E
xitTime File output

860     1940    taskdl.exe      0x1f4daf0       0       -       0       False   2017-05-12 21:
26:23.000000    2017-05-12 21:26:23.000000      Disabled
536     1940    taskse.exe      0x1f53d18       0       -       0       False   2017-05-12 21:
26:22.000000    2017-05-12 21:26:23.000000      Disabled
424     1940    @WanaDecryptor@ 0x1f69b50       0       -       0       False   2017-05-12 21:
25:52.000000    2017-05-12 21:25:53.000000      Disabled
1768    1024    wuauclt.exe     0x1f747c0       7       132     0       False   2017-05-12 21:
22:52.000000    N/A     Disabled
576     1940    @WanaDecryptor@ 0x1f8ba58       0       -       0       False   2017-05-12 21:
```

- `vol -f /Scenarios/Investigations/Investigation-2.raw windows.dlllist | grep -i “decryptor”`



```
740     @WanaDecryptor@ 0x71ab0000      0x17000 WS2 32.dll      C:\WINDOWS\system32\WS2 32.dll  N/A     Disabled
```

- `vol -f /Scenarios/Investigations/Investigation-2.raw windows.handles | grep “1940”`



```
1940    taskschе.exe    0x821883e8      0x40    Mutant  0x120001        ShimCacheMutex
1940    tasksche.exe    0xe16644e0      0x44    Section 0x2     ShimSharedMemory
1940    tasksche.exe    0x822386a8      0x48    File    0x100001        \Device\KsecDD
1940    tasksche.exe    0x823d54d0      0x4c    Semaphore       0x1f0003        shell.{A48F1A32-A340-11
1940    tasksche.exe    0x823a0cd0      0x50    File    0x100020        \Device\HarddiskVolume1\WINDOWS
202
1940    tasksche.exe    0x8224f180      0x54    Mutant  0x1f0001        MsWinZonesCacheCounterMutexA
1940    tasksche.exe    0x822e3b08      0x58    Mutant  0x1f0001        MsWinZonesCacheCounterMutexA0
```

Utilizing Help Documentation:

- Referenced the Volatility help menu (`vol -h`) to identify appropriate plugins for specific tasks.

Final Analysis and Reporting:

- Documented findings, including suspicious processes, parent processes, PIDs, paths, DLLs, mutexes, and malware types.

Summary:

- In **Case 001**, a banking trojan disguised as an Adobe document compromised a quarantined endpoint. Memory analysis of `/Scenarios/Investigations/Investigation-1.vmem` revealed the suspicious process `reader_sl.exe` (PID 1640) spawned by `explorer.exe` (PID 1484). Indicators include a malicious user-agent (`Mozilla/5.0`), connections to suspicious domains like Chase Bank, and IP `41.168.5.140`.
- In **Case 002**, Wannacry ransomware was identified. Analysis of `/Scenarios/Investigations/Investigation-2.raw` found the decryptor process `@WanaDecryptor@` (PID 740) running from `C:\Intel\ivecuqmanpnirkt615`, spawned by `tasksche.exe` (PID 1940). It used `Ws2_32.dll` for socket creation and exhibited a known mutex, `MsWinZonesCacheCounterMutexA`. The `windows.filescan` plugin was recommended to identify malware-related files.

Takeaways: Proactive threat hunting, robust endpoint monitoring, and memory forensics are critical for detecting advanced threats. Organizations must ensure proper isolation of compromised systems and maintain detailed incident response procedures for effective containment and analysis.