Challenge Name: Phantom Persistence

Challenge_Descriptions

Spectr Corp's SOC received reports of unusual behaviour on an employee workstation. A memory dump was acquired post-reboot and is provided to you for forensic investigation. You are given a memory image memory.raw from a Windows system. Your goal is to determine the persistence mechanism used by the attacker.

Submit the flag in this format:

flag{method_name}

In this challenge:

- method= type of persistence
- name= name of the registry value used by the attacker

File provided:

- memory.raw

Step 1: Validate the Memory Dump

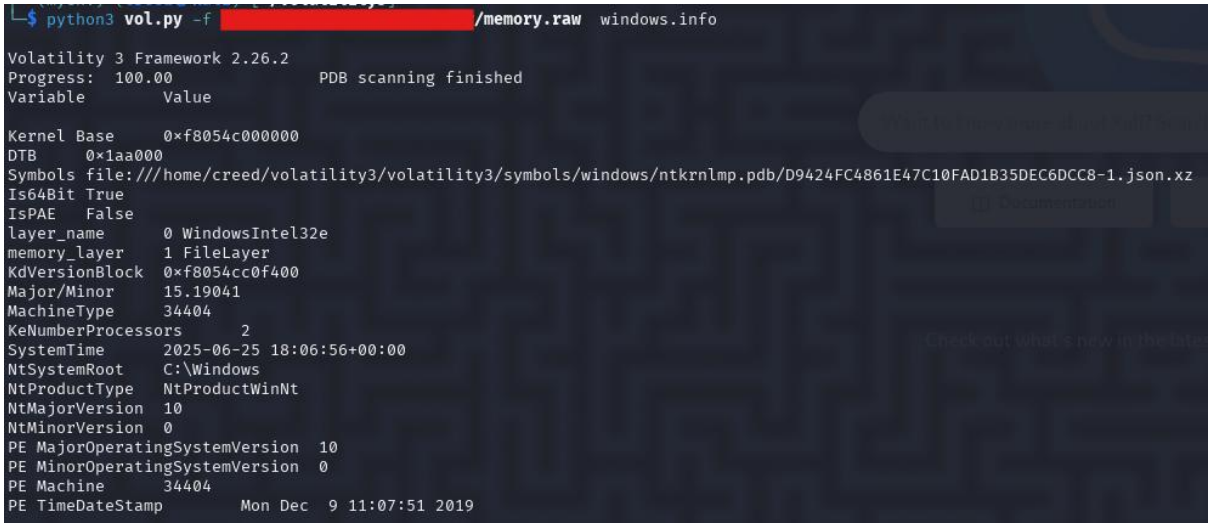
We start by verifying that the memory image can be parsed by Volatility. We use Volatility because it is a powerful memory forensics framework that allows investigators, incident responders, and malware analysts to analyse volatile memory (RAM) dumps from compromised systems.

Commands:

- *python3 vol.py -f memory.raw windows.info*

What it does:

- **python3 vol.py**: Runs the vol.py script using Python 3, this is the main entry point for Volatility 3.
- **-f memory.raw**: Specifies the input **memory dump file** (memory.raw) to analyze.
- **windows.info**: Tells Volatility to run the **windows.info plugin**, which extracts basic OS information from the Windows memory image.



This confirms the dumped memory is from a Windows 10 x64 system and allows Volatility to identify relevant plugins.

Step 2: Investigate for Persistence

In cybersecurity, persistence refers to techniques attackers use to maintain access to a compromised system even after reboots, user logoffs, or other interruptions.

Once attackers gain access to a system, they don't want to lose it, so they install backdoors or re-launch malware automatically. One common method for this is modifying the **Windows Registry Run Keys**.

What is Windows Registry Run Keys?

The Windows Registry is a central database that stores system settings, configurations, and startup programs.

Within it, there are specific keys like:

"HKCU\Software\Microsoft\Windows\CurrentVersion\Run"

These keys tell Windows to automatically launch specific programs when a user logs in. This makes it a prime target for attackers who want their malware to persist every time the machine boots or a user logs in.

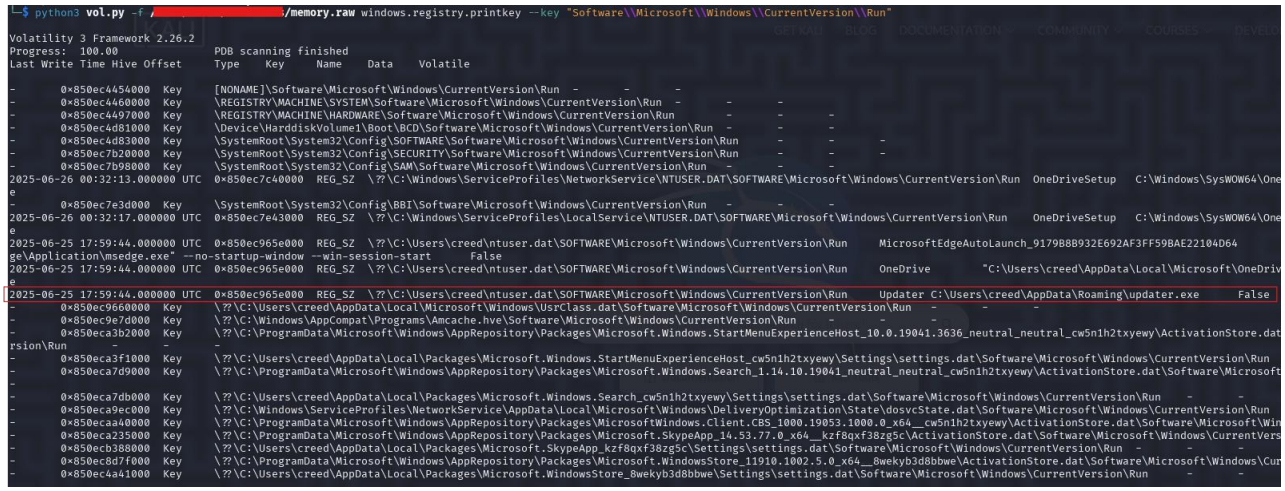
Since the attacker may have used Registry Run Keys for persistence, we check the HKCU\Software\Microsoft\Windows\CurrentVersion\Run path:

Commands:

- *python3 vol.py -f memory.raw windows.registry.printkey –key "Software\\Microsoft\\Windows\\CurrentVersion\\Run"*

What is does:

- **python3 vol.py**: Runs the **Volatility 3** framework using Python 3.
- **-f memory.raw**: Specifies the **memory dump file** to analyze (in this case, memory.raw).
- **windows.registry.printkey**: Invokes the **Volatility plugin** that prints the content of a specific Windows **Registry key**.
- **--key "Software\\Microsoft\\Windows\\CurrentVersion\\Run"**: Provides the exact **registry path** to inspect.Double backslashes (\\) are used to **escape** backslashes in the Windows registry path.



Step 3: Analyze the Output

Volatility returns the following key entry:

```
Key: \??\C:\Users\[redacted]ntuser.dat\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Name: Updater
Data: C:\Users\[redacted]\AppData\Roaming\updater.exe
```

This indicates a classic attacker technique:

- The malware was copied to a hidden path in the user profile (AppData\Roaming)
- A Run key named Updater was added to launch the malware on every login

Step 4: Extract the Flag

Based on the format and findings:

- Method = run_key
- Name = updater

Final Flag:

WARZONE{run_key_updater}