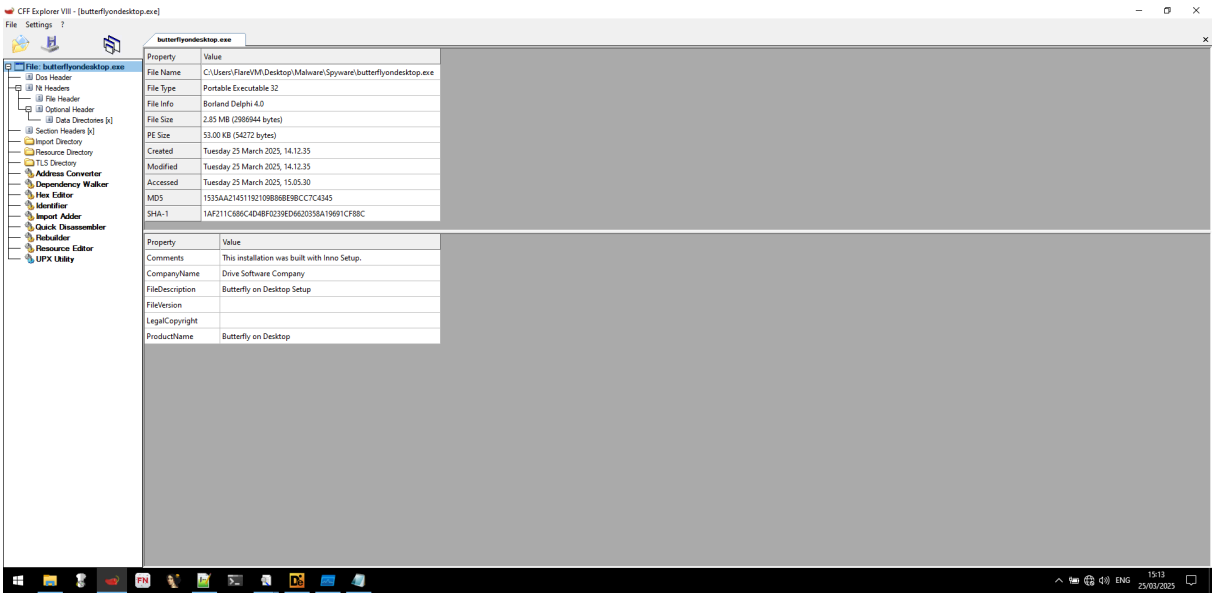
ANALISI STATICA



Per l'analisi statica ho usato **CFF Explorer** per esaminare il file eseguibile **`Butterfly Desktop.exe`**.

Ho visto che si tratta di un file di tipo **Portable Executable a 32 bit (PE32)**, compilato con **Inno Setup**, che è un programma usato per creare installatori.

Il nome del prodotto è ***Butterfly on Desktop*** e il nome dell'azienda è ***Drive Software Company***, anche se potrebbe essere falso o inserito solo per sembrare affidabile.

Il file ha un **MD5 hash** ben definito (**`1553A14211591E0806EB96FC7CC43445`**), l'ho utilizzato per cercare il file su **VirusTotal**, e verificare se era già stato riconosciuto come malware, ma non ho ottenuto informazioni.

Il punto di ingresso del codice **(`Entry Point`)** è **`0x00009C40`**, ovvero da dove inizia davvero l'esecuzione del programma. Ho scoperto che questo valore è importante se si vuole analizzare il comportamento interno del codice in modo più approfondito.

Infine, ho controllato le **librerie importate**, come **`user32.dll`** e **`kernel32.dll`**, che sono usate normalmente per gestire **finestre, processi, memoria, input da tastiera e mouse**.

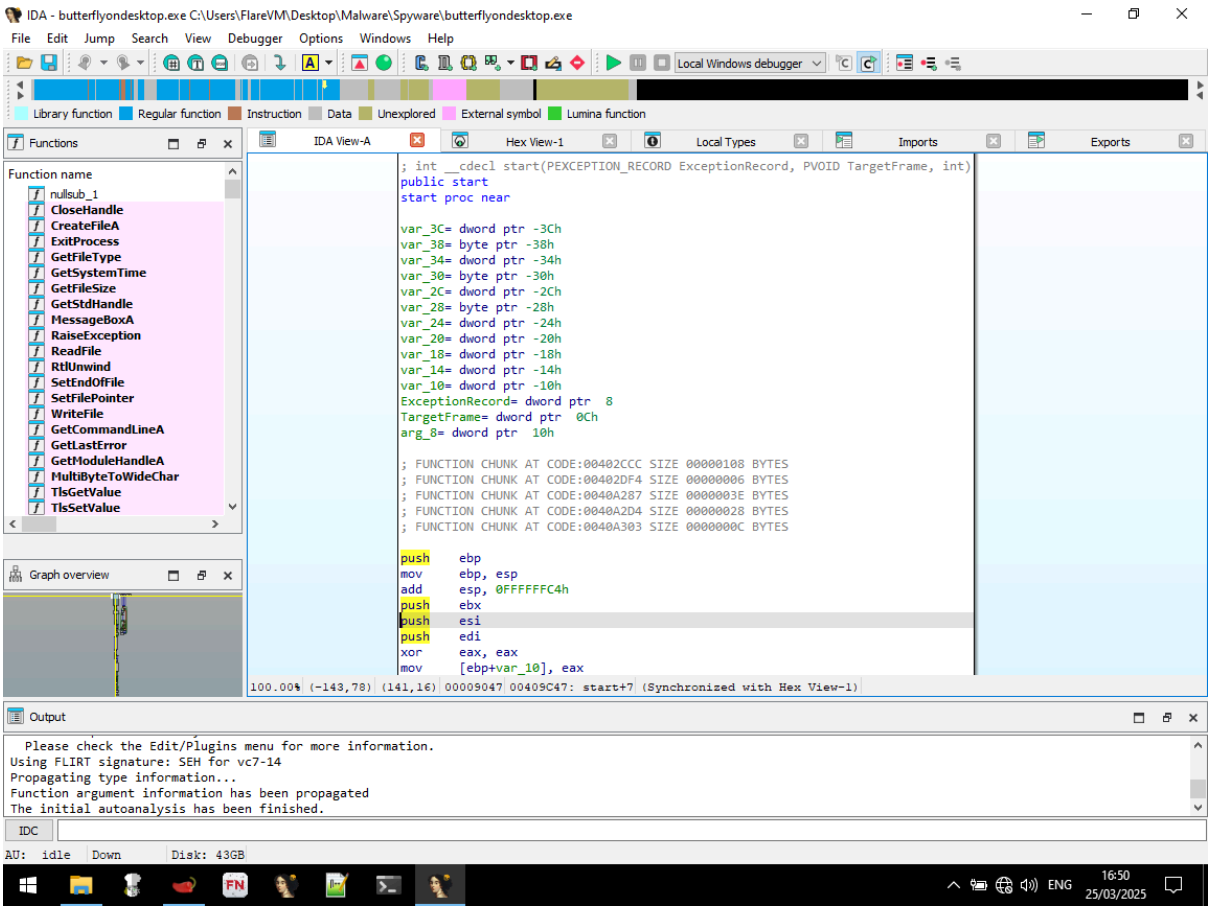
Non ho trovato nulla di particolarmente sospetto in questa fase, ma ho approfondito scoprendo che queste librerie sono spesso usate anche da malware per interagire con il sistema operativo.

Successivamente ho aperto il malware con IDA e ho analizzato il **punto di partenza del programma** (entry point). Il codice che ho visto all'inizio serve solo a **preparare la memoria e le variabili**, cioè a mettere tutto in ordine prima di iniziare a fare qualcosa.

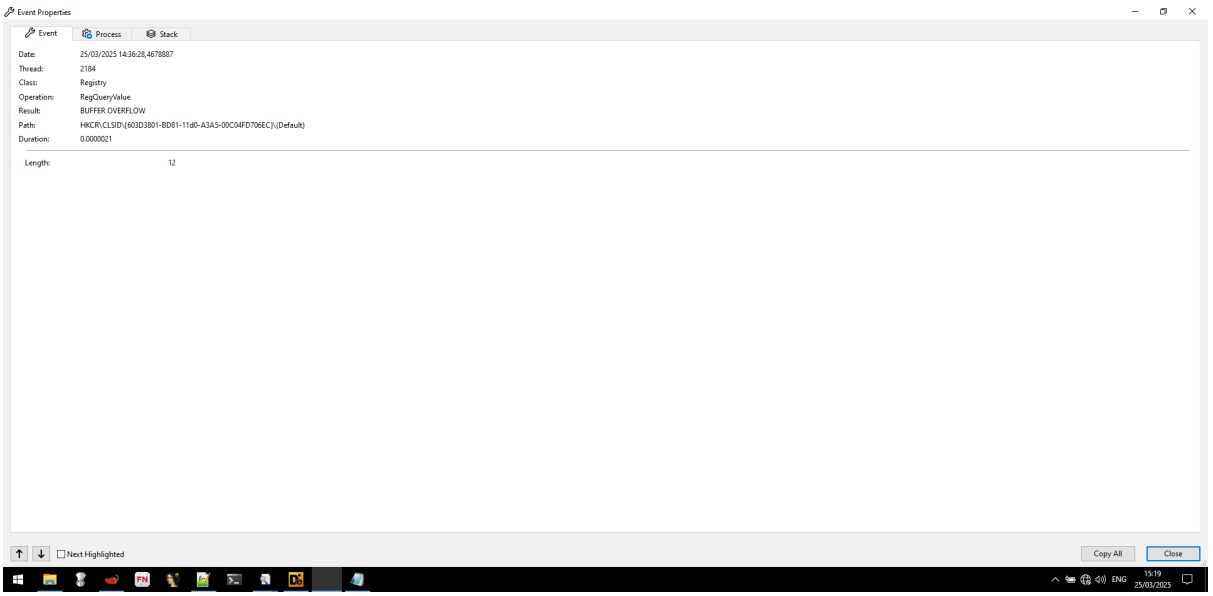
Guardando l'elenco delle funzioni usate dal programma, ho notato che ci sono alcune API di Windows come **`ReadFile`, `WriteFile` e `CreateFileA`**, che servono per **leggere, scrivere o aprire file**, e **`MessageBoxA`**, che può essere usata per **mostrare messaggi all'utente**.

Queste funzioni mi fanno pensare che il malware potrebbe **interagire con i file presenti nel sistema** o **mostrare popup all'utente**.

Bisogna poi seguire le istruzioni del tipo **`call`**, per vedere esattamente **quali funzioni usa e cosa fa davvero** il malware.



ANALISI DINAMICA



- **Operazione**: `RegQueryValue`
- **Risultato**: `BUFFER OVERFLOW` (!!!)
- **Registro accesso**: `HKCR\CLSID\{...}`

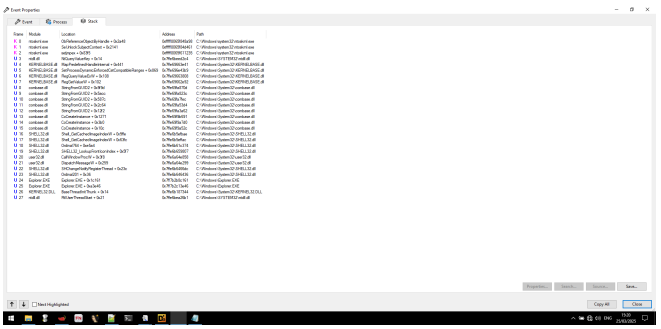
Durante l'analisi dinamica, ho osservato un evento che si è verificato mentre il malware era in esecuzione. In particolare, il programma ha cercato di leggere un valore dal registro di sistema tramite l'operazione **`RegQueryValue`**.

Questa richiesta ha avuto come risultato un **"buffer overflow"**, cioè un errore che succede quando il programma prova a scrivere o leggere più dati di quelli che può gestire.

Questo tipo di comportamento è sospetto, perché potrebbe indicare che il programma sta cercando di **usare una vulnerabilità** per eseguire operazioni non autorizzate, come **scrivere nella memoria di sistema o eseguire codice dannoso**.

Il tentativo è stato fatto su una chiave di registro molto usata **(`HKCR\CLSID\{...}`)**, che di solito è collegata ai componenti del sistema di Windows, quindi potrebbe cercare di modificare o controllare alcune parti del sistema operativo.

Successivamente ho aperto lo stack:



Ho notato che il ***Butterfly Desktop.exe*** usa alcune librerie di sistema molto comuni, come **`SHELL32.dll`**, **`USER32.dll`**, **`KERNEL32.dll`** e **`NTDLL.dll`**.

Ho fatto delle ricerche ed ho appreso che queste librerie servono per controllare parti importanti del sistema operativo, come il desktop, le finestre, il mouse e la tastiera.

In particolare, ho visto che vengono usate funzioni come **`DispatchMessageW`** e **`SHChangeNotifyRegisterThread`**, che (sempre secondo le mie ricerche) permettono al programma di inviare messaggi alle finestre e di accorgersi se qualcosa cambia sul desktop, come ad esempio un'icona o un file.

Questo mi fa concludere che il malware voglia mostrare qualcosa sullo schermo, come animazioni o effetti grafici (le farfalle in questo caso, avendolo visto in esecuzione), e quindi interagisce direttamente con l'interfaccia di Windows. Il suo comportamento sembra progettato per attirare l'attenzione dell'utente o per disturbare, rimanendo visibile sul desktop.