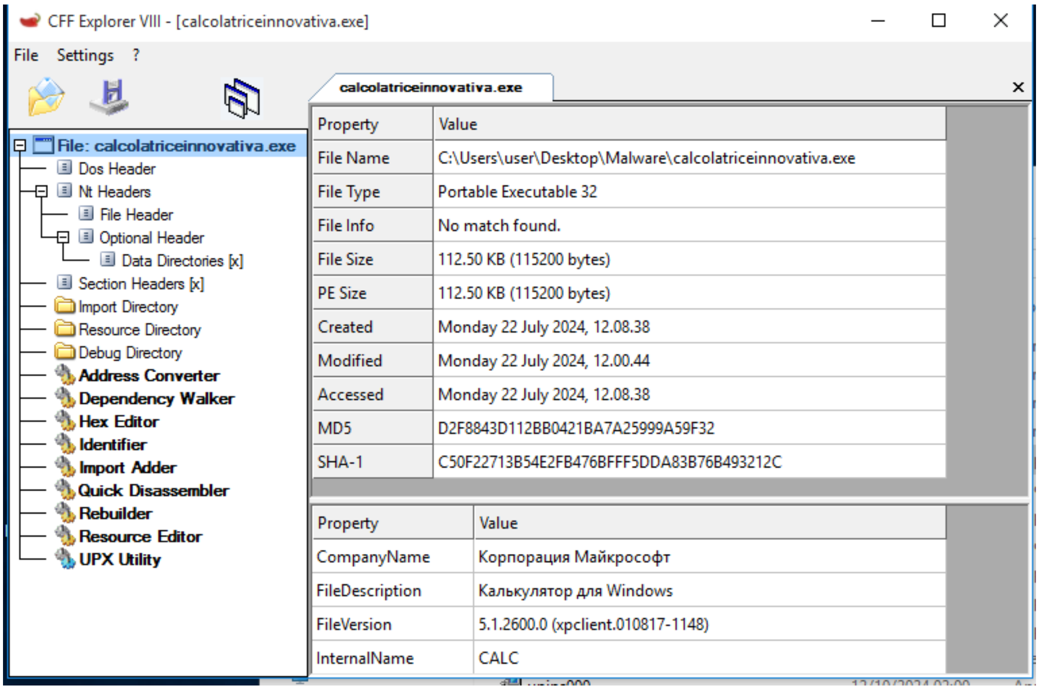
W21D4 – Analisi statica basica

Esercizio obbligatorio

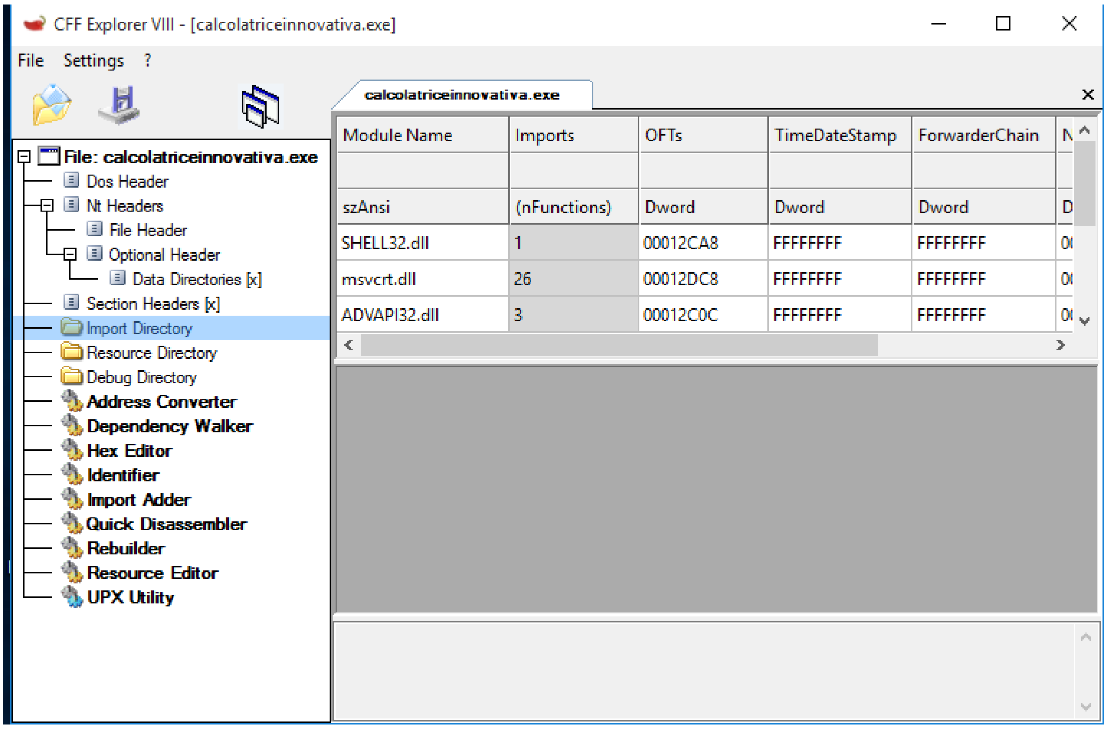
Con la flare Vm ho caricato il malware su CFF Explorer e poi ho iniziato l'analisi statica basica. Queste sono le prime informazioni che ho trovato riguardo al file del malware.



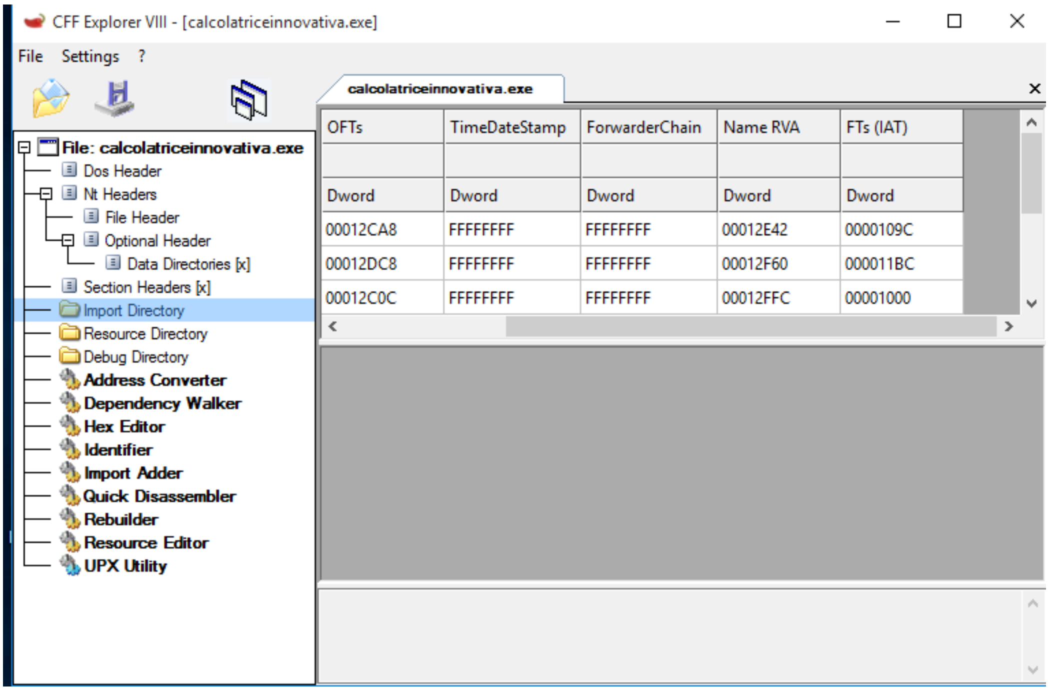
Property	Value
File Name	C:\Users\user\Desktop\Malware\calcolatriceinnovativa.exe
File Type	Portable Executable 32
File Info	No match found.
File Size	112.50 KB (115200 bytes)
PE Size	112.50 KB (115200 bytes)
Created	Monday 22 July 2024, 12.08.38
Modified	Monday 22 July 2024, 12.00.44
Accessed	Monday 22 July 2024, 12.08.38
MD5	D2F8843D112BB0421BA7A25999A59F32
SHA-1	C50F22713B54E2FB476BFFF5DDA83B76B493212C

Property	Value
CompanyName	Корпорация Майкрософт
FileDescription	Калькулятор для Windows
FileVersion	5.1.2600.0 (xpclient.010817-1148)
InternalName	CALC

Ho cercato le librerie importate guardando su "import directory", che riporta le librerie importate direttamente e poi ho controllato su "dependency walker" le dipendenze, che sono le librerie dalle quali il file eseguibile dipende e che devono essere caricate per il suo corretto funzionamento. Questa è la import directory.

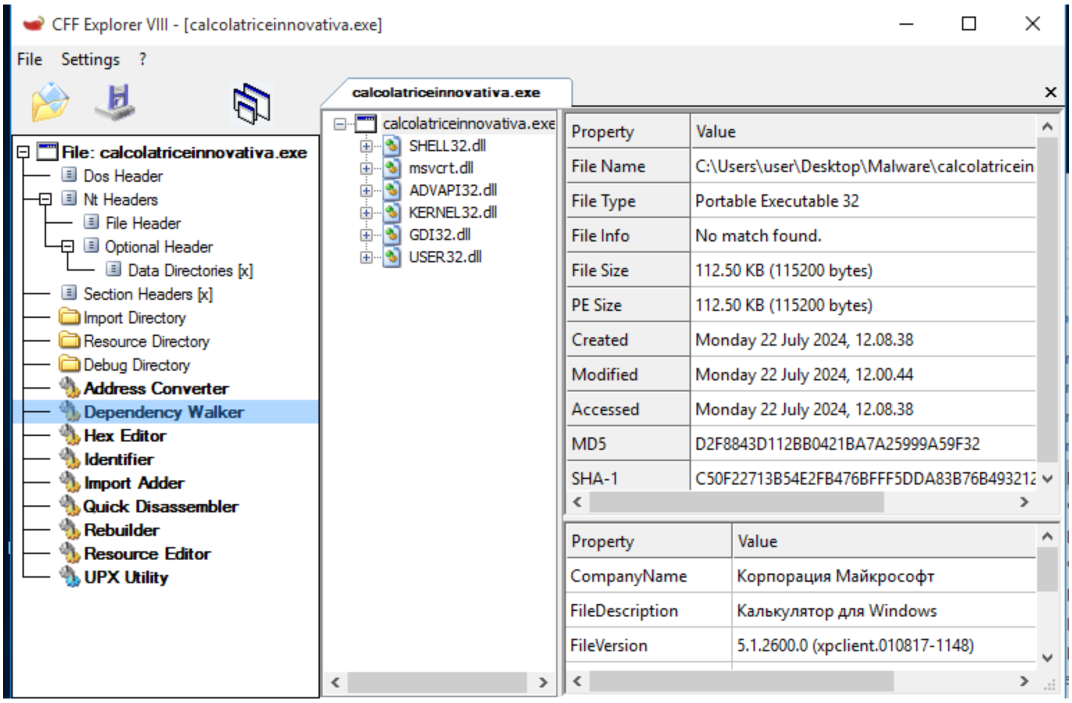


Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain
szAnsi	(nFunctions)	Dword	Dword	Dword
SHELL32.dll	1	00012CA8	FFFFFFFF	FFFFFFFF
msvcrt.dll	26	00012DC8	FFFFFFFF	FFFFFFFF
ADVAPI32.dll	3	00012C0C	FFFFFFFF	FFFFFFFF



OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
Dword	Dword	Dword	Dword	Dword
00012CA8	FFFFFFFF	FFFFFFFF	00012E42	0000109C
00012DC8	FFFFFFFF	FFFFFFFF	00012F60	000011BC
00012C0C	FFFFFFFF	FFFFFFFF	00012FFC	00001000

Queste sono invece le dipendenze trovate nel dependency walker.



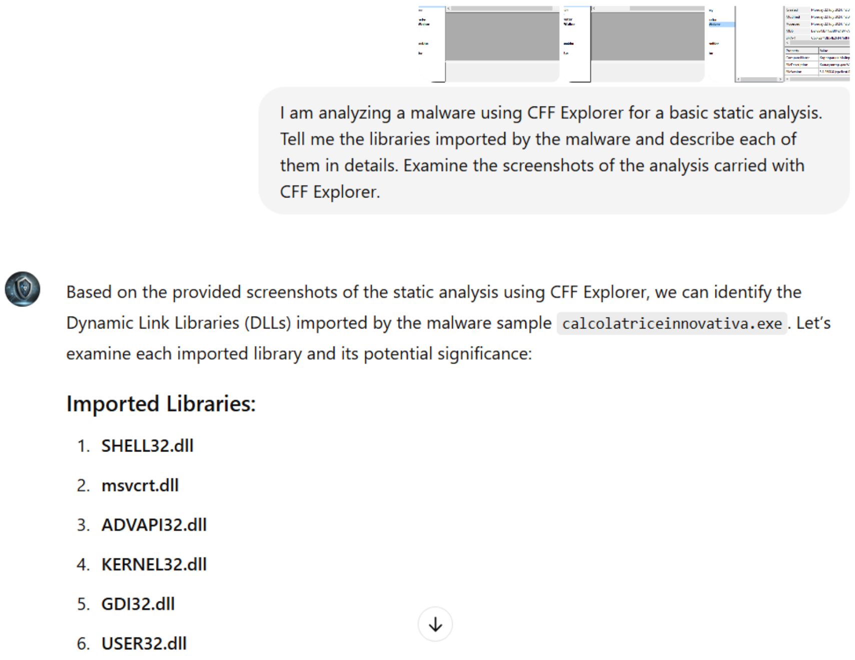
calcolatriceinnovativa.exe
- SHELL32.dll
- msvcrt.dll
- ADVAPI32.dll
- KERNEL32.dll
- GDI32.dll
- USER32.dll

Property	Value
File Name	C:\Users\user\Desktop\Malware\calcolatricein
File Type	Portable Executable 32
File Info	No match found.
File Size	112.50 KB (115200 bytes)
PE Size	112.50 KB (115200 bytes)
Created	Monday 22 July 2024, 12.08.38
Modified	Monday 22 July 2024, 12.00.44
Accessed	Monday 22 July 2024, 12.08.38
MD5	D2F8843D112BB0421BA7A25999A59F32
SHA-1	C50F22713B54E2FB476BFFF5DDA83B76B493212

Property	Value
CompanyName	Корпорация Майкрософт
FileDescription	Калькулятор для Windows
FileVersion	5.1.2600.0 (xpclient.010817-1148)

Le librerie importate direttamente sono SHELL32.dll, msvcrt.dll e ADVAPI32.dll, mentre le altre librerie necessarie per il malware sono, oltre a queste, anche KERNEL32.dll, GDI32.dll e USER32.dll. SHELL32.dll è una libreria che serve per interagire con la Windows Shell, per esempio per la gestione dei file e può essere utilizzata dal malware per interagire con il file system, ad esempio per aprire nuovi file, inserire file malevoli, modificare file esistenti, aprire nuovi processi. Interagisce anche con il sistema operativo, con la shell e con gli elementi del desktop, dando la possibilità al malware di agire sul sistema. Msvcrt.dll sta per Microsoft

Visual C++ Runtime Library e include librerie in C necessarie per la gestione della memoria o per la gestione degli input ed output I/O. Consente al malware di effettuare operazioni sulla memoria, sul buffer e di gestire le stringhe e gli input ed output dell'utente per i suoi scopi malevoli. La libreria ADVAPI32.dll fornisce servizi avanzati di Windows API legati alla sicurezza e alle operazioni di registro, per esempio per l'autenticazione e serve al malware per effettuare delle modifiche ai registri di Windows, al fine di eseguire una privilege escalation o di rimanere persistente nel sistema. KERNEL32.dll è una libreria che viene importata molto spesso nei file eseguibili Windows e serve per funzioni di sistema molto importanti come la gestione della memoria, dei threads, dei processi. Potrebbe essere sfruttata da un malware per iniettare dei threads malevoli in processi legittimi, allocare memoria aggiuntiva per alcuni payload o processi malevoli e gestire gli input ed output. GDI32.dll, ovvero Graphic Device Interface, riguarda tutti gli aspetti grafici, ad esempio riguardo alle immagini o al testo. Oltre a curare l'aspetto del malware per renderlo non sospetto, potrebbe servire per aprire delle finestre o per catturare il contenuto dello schermo tramite la manipolazione di funzioni grafiche. Infine, USER32.dll serve per gestire le interazioni con l'utente e potrebbe essere utile al malware per prendere il controllo dell'input utente, manipolare le finestre di dialogo e cercare o intercettare ciò che scrive l'utente. Sulla base delle librerie importate si capisce che il malware è in grado di interagire con il sistema infettato su più livelli, creando interazioni con il registro per mantenersi persistente, manipolando le configurazioni, i processi e la memoria per iniettare codici e thread malevoli ed interagendo con l'utente tramite la registrazione di ciò che fa e la modifica dei suoi input. Potrebbe quindi portare avanti diverse attività malevole, fungendo ad esempio da keylogger, modificando la memoria oppure installando una backdoor persistente. In seguito riporto i risultati ottenuti analizzando con ChatGPT.

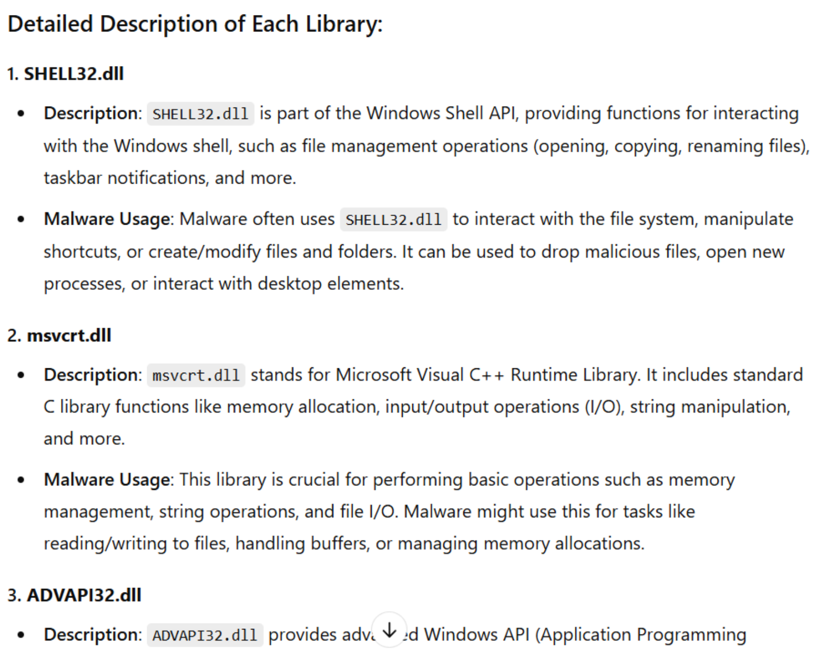


I am analyzing a malware using CFF Explorer for a basic static analysis. Tell me the libraries imported by the malware and describe each of them in details. Examine the screenshots of the analysis carried with CFF Explorer.

Based on the provided screenshots of the static analysis using CFF Explorer, we can identify the Dynamic Link Libraries (DLLs) imported by the malware sample `calcolatriceinnovativa.exe`. Let's examine each imported library and its potential significance:

Imported Libraries:

1. **SHELL32.dll**
2. **msvcrt.dll**
3. **ADVAPI32.dll**
4. **KERNEL32.dll**
5. **GDI32.dll**
6. **USER32.dll**



Detailed Description of Each Library:

1. SHELL32.dll

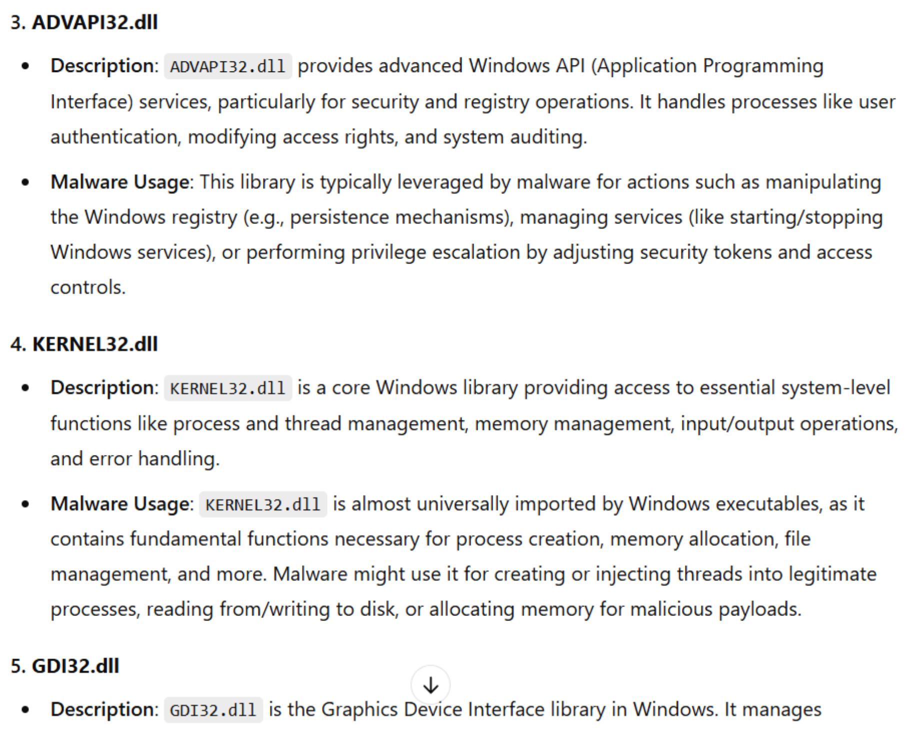
- **Description**: `SHELL32.dll` is part of the Windows Shell API, providing functions for interacting with the Windows shell, such as file management operations (opening, copying, renaming files), taskbar notifications, and more.
- **Malware Usage**: Malware often uses `SHELL32.dll` to interact with the file system, manipulate shortcuts, or create/modify files and folders. It can be used to drop malicious files, open new processes, or interact with desktop elements.

2. msvcrt.dll

- **Description**: `msvcrt.dll` stands for Microsoft Visual C++ Runtime Library. It includes standard C library functions like memory allocation, input/output operations (I/O), string manipulation, and more.
- **Malware Usage**: This library is crucial for performing basic operations such as memory management, string operations, and file I/O. Malware might use this for tasks like reading/writing to files, handling buffers, or managing memory allocations.

3. ADVAPI32.dll

- **Description**: `ADVAPI32.dll` provides adv ed Windows API (Application Programming



3. ADVAPI32.dll

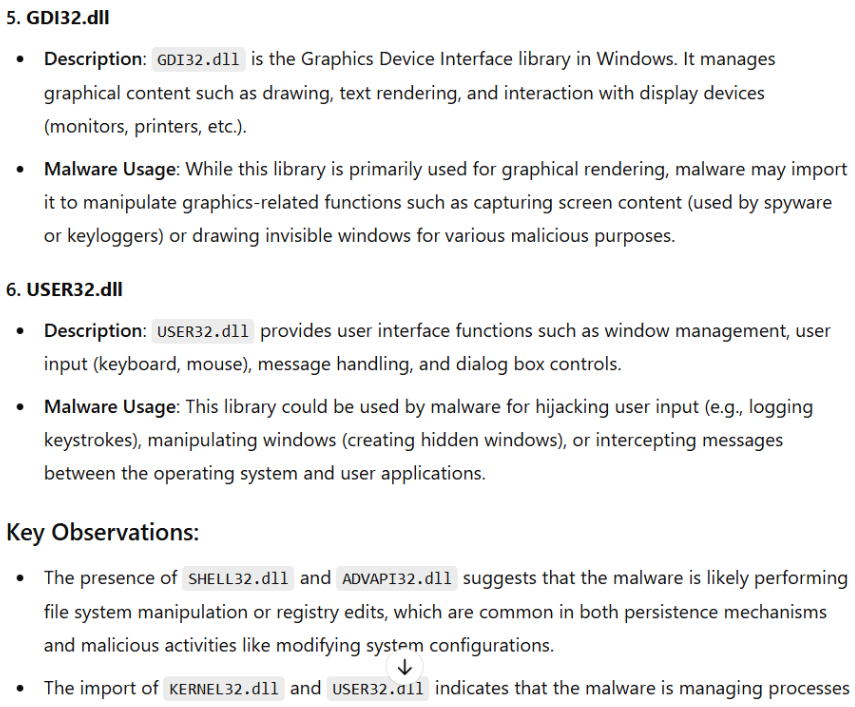
- **Description**: `ADVAPI32.dll` provides advanced Windows API (Application Programming Interface) services, particularly for security and registry operations. It handles processes like user authentication, modifying access rights, and system auditing.
- **Malware Usage**: This library is typically leveraged by malware for actions such as manipulating the Windows registry (e.g., persistence mechanisms), managing services (like starting/stopping Windows services), or performing privilege escalation by adjusting security tokens and access controls.

4. KERNEL32.dll

- **Description**: `KERNEL32.dll` is a core Windows library providing access to essential system-level functions like process and thread management, memory management, input/output operations, and error handling.
- **Malware Usage**: `KERNEL32.dll` is almost universally imported by Windows executables, as it contains fundamental functions necessary for process creation, memory allocation, file management, and more. Malware might use it for creating or injecting threads into legitimate processes, reading from/writing to disk, or allocating memory for malicious payloads.

5. GDI32.dll

- **Description**: `GDI32.dll` is the Graphics Device Interface library in Windows. It manages



5. GDI32.dll

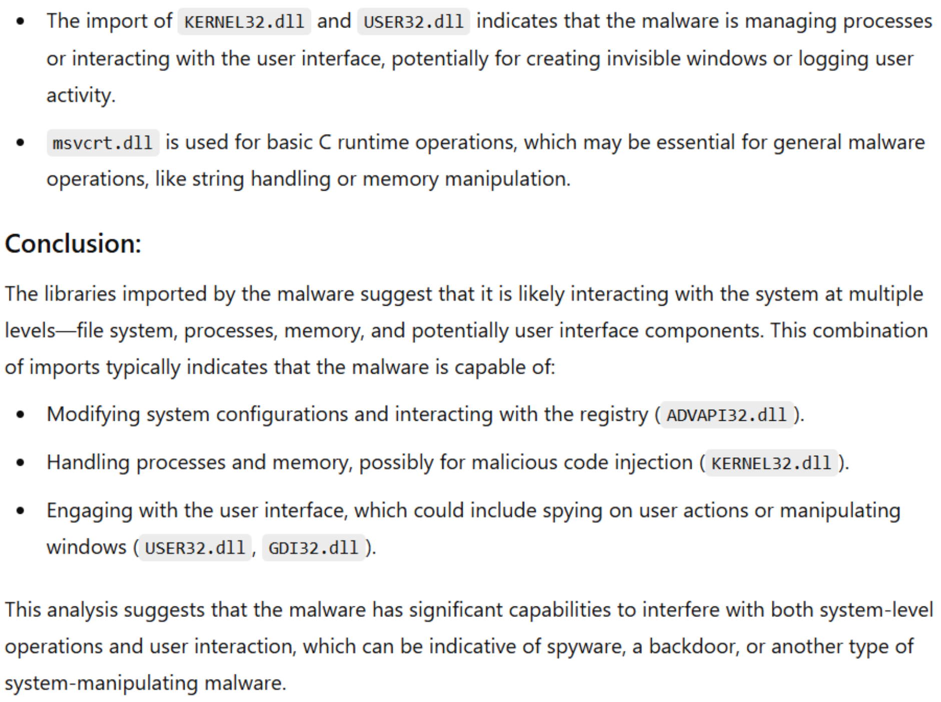
- **Description**: `GDI32.dll` is the Graphics Device Interface library in Windows. It manages graphical content such as drawing, text rendering, and interaction with display devices (monitors, printers, etc.).
- **Malware Usage**: While this library is primarily used for graphical rendering, malware may import it to manipulate graphics-related functions such as capturing screen content (used by spyware or keyloggers) or drawing invisible windows for various malicious purposes.

6. USER32.dll

- **Description**: `USER32.dll` provides user interface functions such as window management, user input (keyboard, mouse), message handling, and dialog box controls.
- **Malware Usage**: This library could be used by malware for hijacking user input (e.g., logging keystrokes), manipulating windows (creating hidden windows), or intercepting messages between the operating system and user applications.

Key Observations:

- The presence of `SHELL32.dll` and `ADVAPI32.dll` suggests that the malware is likely performing file system manipulation or registry edits, which are common in both persistence mechanisms and malicious activities like modifying system configurations.
- The import of `KERNEL32.dll` and `USER32.dll` indicates that the malware is managing processes



- The import of `KERNEL32.dll` and `USER32.dll` indicates that the malware is managing processes or interacting with the user interface, potentially for creating invisible windows or logging user activity.
- `msvcrt.dll` is used for basic C runtime operations, which may be essential for general malware operations, like string handling or memory manipulation.

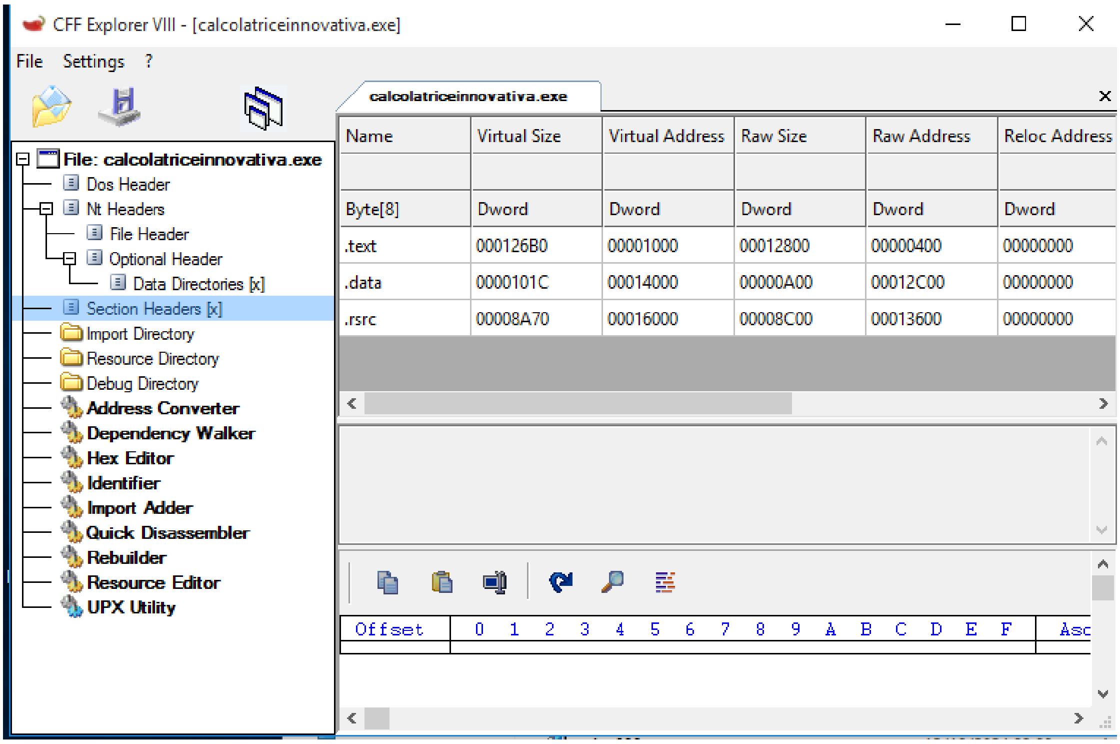
Conclusion:

The libraries imported by the malware suggest that it is likely interacting with the system at multiple levels—file system, processes, memory, and potentially user interface components. This combination of imports typically indicates that the malware is capable of:

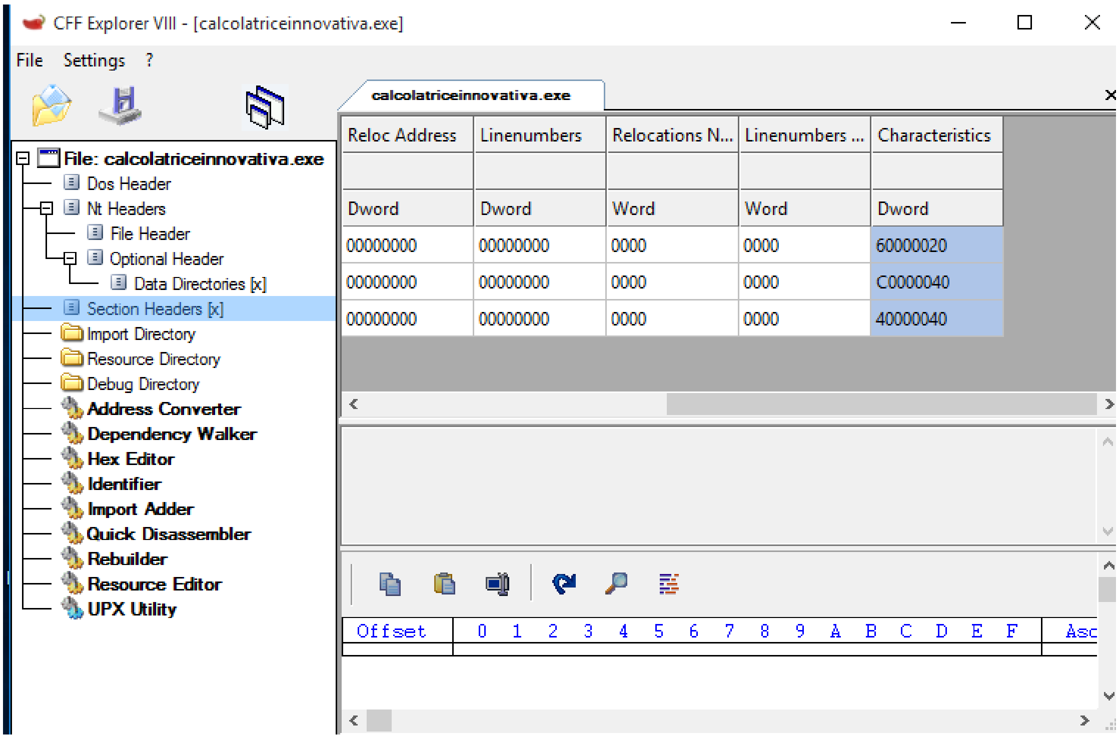
- Modifying system configurations and interacting with the registry (`ADVAPI32.dll`).
- Handling processes and memory, possibly for malicious code injection (`KERNEL32.dll`).
- Engaging with the user interface, which could include spying on user actions or manipulating windows (`USER32.dll`, `GDI32.dll`).

This analysis suggests that the malware has significant capabilities to interfere with both system-level operations and user interaction, which can be indicative of spyware, a backdoor, or another type of system-manipulating malware.

Nell’header PE si trova la sezione “section headers” che indica le varie sezioni di cui si compone il malware. Il malware in analisi si compone di 3 sezioni, ovvero .text, .data, .rsrc.

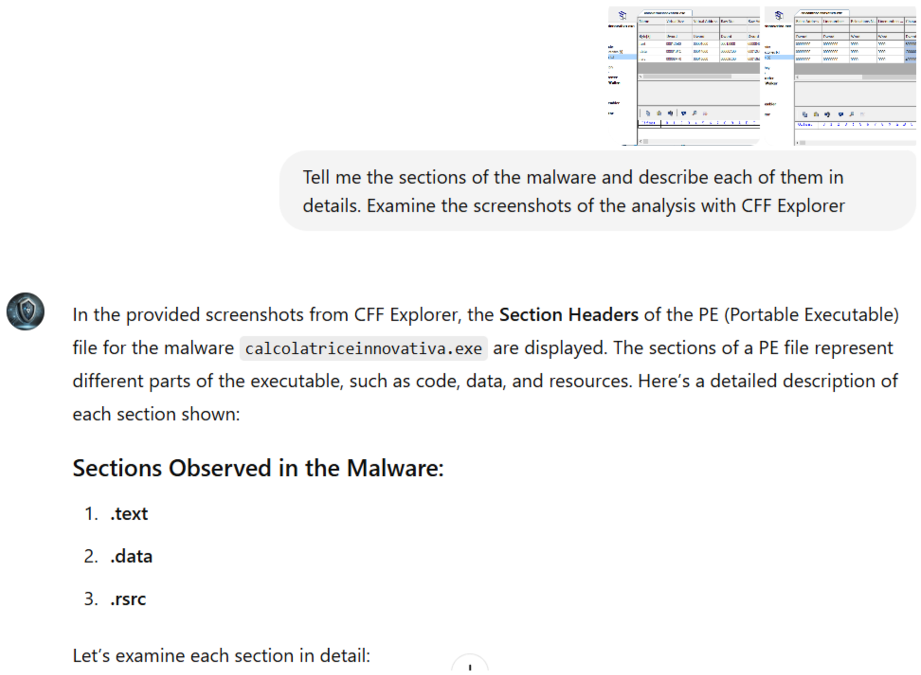


Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address
Byte[8]	Dword	Dword	Dword	Dword	Dword
.text	000126B0	00001000	00012800	00000400	00000000
.data	0000101C	00014000	00000A00	00012C00	00000000
.rsrc	00008A70	00016000	00008C00	00013600	00000000



Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Dword	Dword	Word	Word	Dword
00000000	00000000	0000	0000	60000020
00000000	00000000	0000	0000	C0000040
00000000	00000000	0000	0000	40000040

La sezione .text contiene il codice del programma, che è ovviamente eseguibile e solo leggibile per poter essere eseguito senza essere modificato ed è una parte importantissima perché consente di vedere il codice del malware e capire come potrebbe comportarsi in un sistema infetto. La sezione .data contiene dati di vario genere, ad esempio configurazioni, indirizzi IP, chiavi, icone, stringhe ed altri dati che vengono salvati dal malware durante la sua esecuzione per le sue funzioni. Contiene infatti dati inizializzati e globali utili da esaminare per capire meglio il malware e riguardo ai permessi vi sono sia permessi di lettura che di scrittura. Infine, la sezione .rsrc serve per salvare delle risorse, come informazioni sulle versioni o icone, utili per i programmi. Questa sezione ha permessi di lettura e serve al malware per salvare dei payload malevoli aggiuntivi, dei file per avere un aspetto legittimo e non essere scoperto e altri file criptati. Di seguito riporto gli screenshot dell'analisi fatta da ChatGPT riguardo alle sezioni del malware.



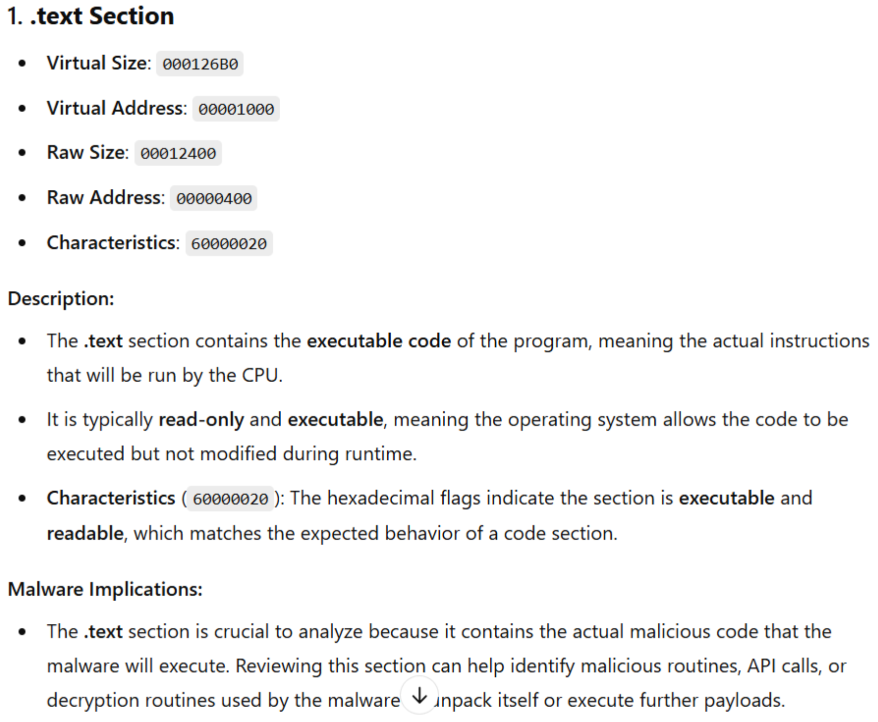
Tell me the sections of the malware and describe each of them in details. Examine the screenshots of the analysis with CFF Explorer

In the provided screenshots from CFF Explorer, the **Section Headers** of the PE (Portable Executable) file for the malware `calcolatriceinnovativa.exe` are displayed. The sections of a PE file represent different parts of the executable, such as code, data, and resources. Here's a detailed description of each section shown:

Sections Observed in the Malware:

1. **.text**
2. **.data**
3. **.rsrc**

Let's examine each section in detail:



1. .text Section

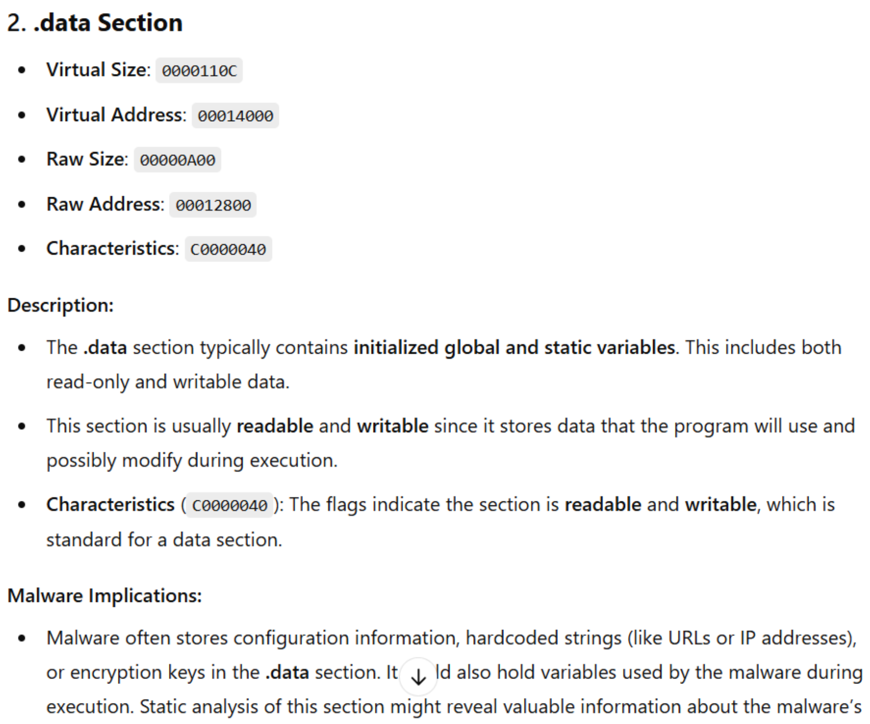
- **Virtual Size**: `000126B0`
- **Virtual Address**: `00001000`
- **Raw Size**: `00012400`
- **Raw Address**: `00000400`
- **Characteristics**: `60000020`

Description:

- The **.text** section contains the **executable code** of the program, meaning the actual instructions that will be run by the CPU.
- It is typically **read-only** and **executable**, meaning the operating system allows the code to be executed but not modified during runtime.
- **Characteristics** (`60000020`): The hexadecimal flags indicate the section is **executable** and **readable**, which matches the expected behavior of a code section.

Malware Implications:

- The **.text** section is crucial to analyze because it contains the actual malicious code that the malware will execute. Reviewing this section can help identify malicious routines, API calls, or decryption routines used by the malware [illegible] unpack itself or execute further payloads.



2. .data Section

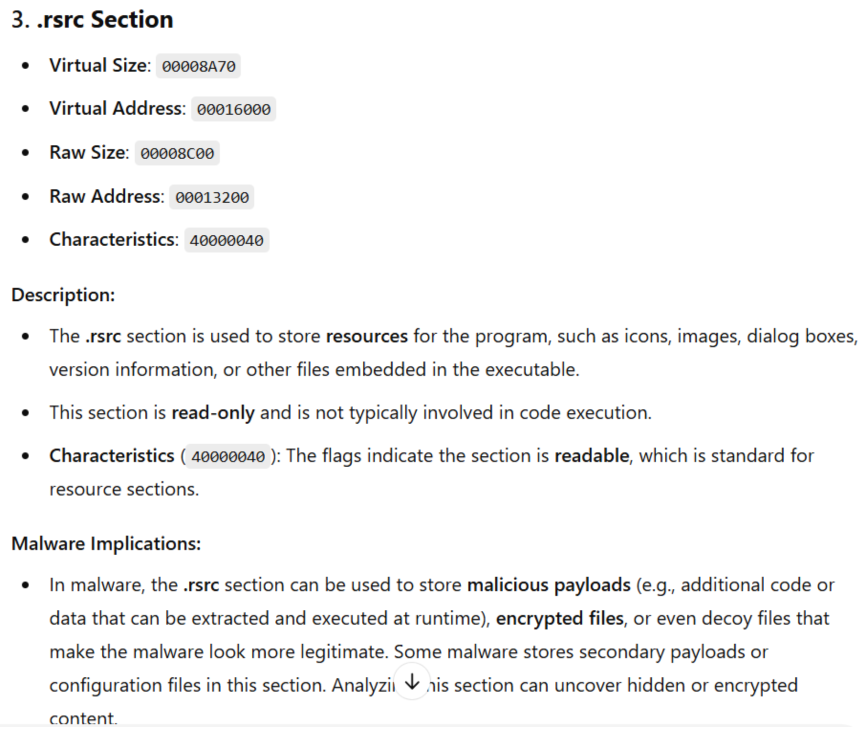
- **Virtual Size**: `0000110C`
- **Virtual Address**: `00014000`
- **Raw Size**: `00000A00`
- **Raw Address**: `00012800`
- **Characteristics**: `C0000040`

Description:

- The **.data** section typically contains **initialized global and static variables**. This includes both read-only and writable data.
- This section is usually **readable** and **writable** since it stores data that the program will use and possibly modify during execution.
- **Characteristics** (`C0000040`): The flags indicate the section is **readable** and **writable**, which is standard for a data section.

Malware Implications:

- Malware often stores configuration information, hardcoded strings (like URLs or IP addresses), or encryption keys in the **.data** section. It ↓ ld also hold variables used by the malware during execution. Static analysis of this section might reveal valuable information about the malware's



3. .rsrc Section

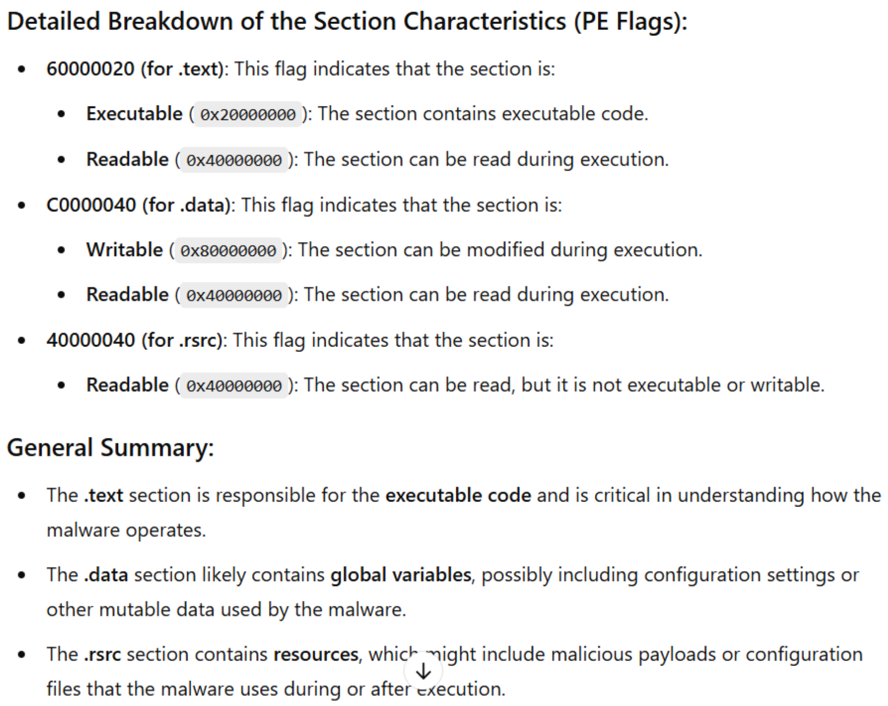
- **Virtual Size**: `00008A70`
- **Virtual Address**: `00016000`
- **Raw Size**: `00008C00`
- **Raw Address**: `00013200`
- **Characteristics**: `40000040`

Description:

- The **.rsrc** section is used to store **resources** for the program, such as icons, images, dialog boxes, version information, or other files embedded in the executable.
- This section is **read-only** and is not typically involved in code execution.
- **Characteristics** (`40000040`): The flags indicate the section is **readable**, which is standard for resource sections.

Malware Implications:

- In malware, the **.rsrc** section can be used to store **malicious payloads** (e.g., additional code or data that can be extracted and executed at runtime), **encrypted files**, or even decoy files that make the malware look more legitimate. Some malware stores secondary payloads or configuration files in this section. Analyzi ↓ his section can uncover hidden or encrypted content.

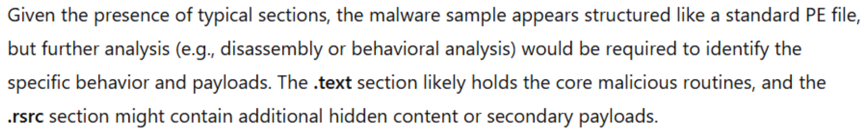


Detailed Breakdown of the Section Characteristics (PE Flags):

- **60000020 (for .text)**: This flag indicates that the section is:
  - **Executable** (`0x20000000`): The section contains executable code.
  - **Readable** (`0x40000000`): The section can be read during execution.
- **C0000040 (for .data)**: This flag indicates that the section is:
  - **Writable** (`0x80000000`): The section can be modified during execution.
  - **Readable** (`0x40000000`): The section can be read during execution.
- **40000040 (for .rsrc)**: This flag indicates that the section is:
  - **Readable** (`0x40000000`): The section can be read, but it is not executable or writable.

General Summary:

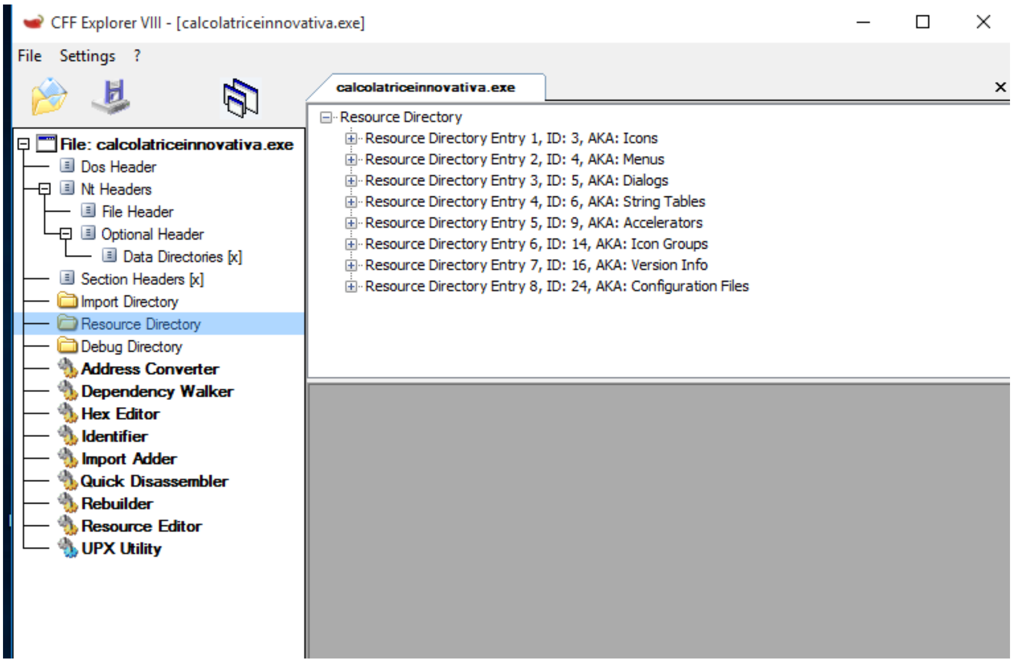
- The **.text** section is responsible for the **executable code** and is critical in understanding how the malware operates.
- The **.data** section likely contains **global variables**, possibly including configuration settings or other mutable data used by the malware.
- The **.rsrc** section contains **resources**, which might include malicious payloads or configuration files that the malware uses during or after execution.



Given the presence of typical sections, the malware sample appears structured like a standard PE file, but further analysis (e.g., disassembly or behavioral analysis) would be required to identify the specific behavior and payloads. The **.text** section likely holds the core malicious routines, and the **.rsrc** section might contain additional hidden content or secondary payloads.

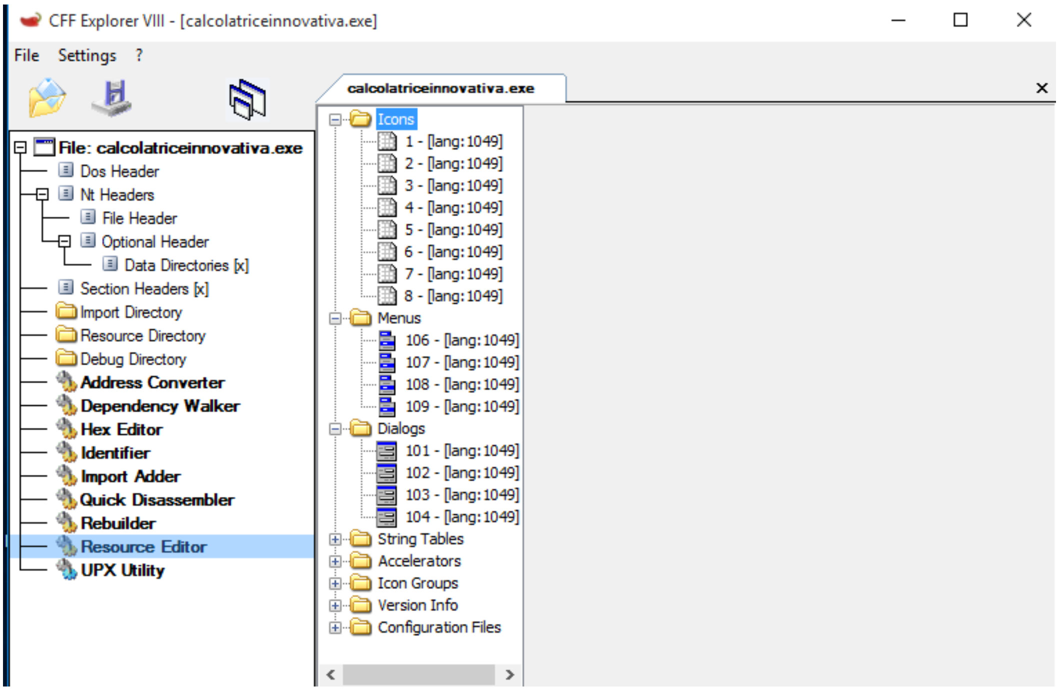
Esercizio facoltativo

Ho continuato ad analizzare il malware, esaminandolo ulteriormente con CFF Explorer. Ho esaminato tutto il DOS Header, trovando la firma MZ che riconosce il file come eseguibile e valido e poi ho visto la resource directory e il resource editor che contengono le risorse presenti nel file del malware e che possono servire ad apparire come un software legittimo e insospettabile. Hex dump contiene la rappresentazione esadecimale dei dati binari contenuti nel file. Riporto gli screenshot dell'ulteriore analisi del malware.


CFF Explorer VIII - [calcolatriceinnovativa.exe]
File
Settings
?
calcolatriceinnovativa.exe

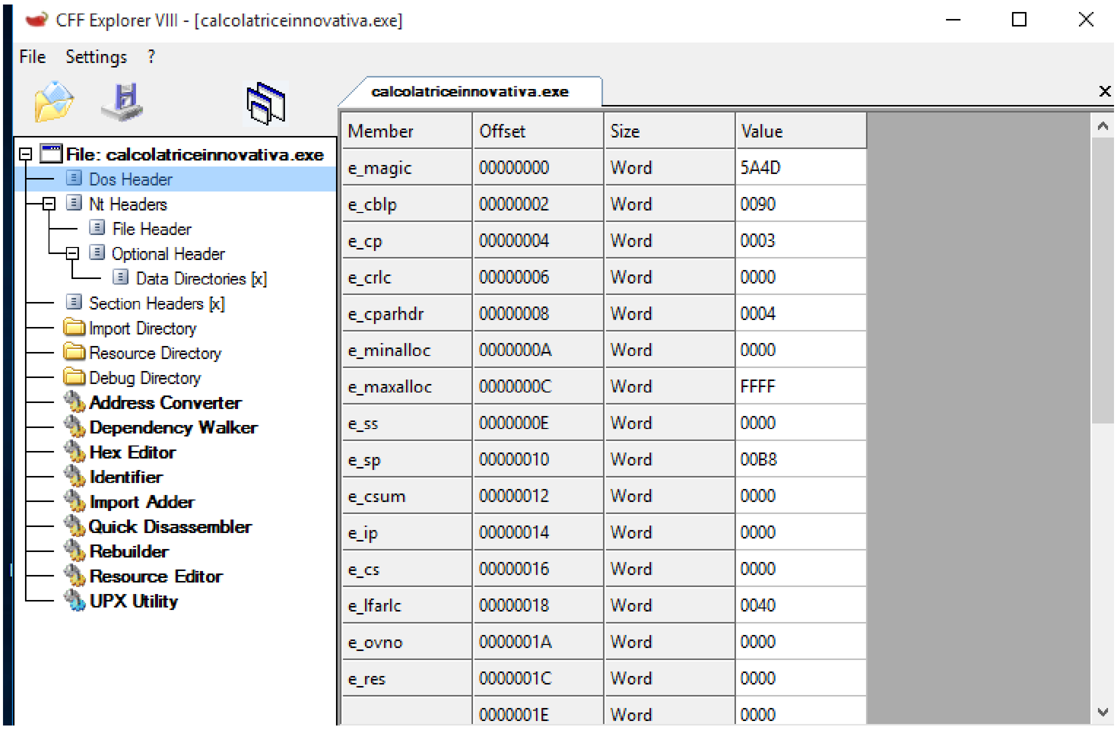
```
File: calcolatriceinnovativa.exe
    Dos Header
    Nt Headers
        File Header
        Optional Header
            Data Directories [x]
    Section Headers [x]
    Import Directory
    Resource Directory
    Debug Directory
    Address Converter
    Dependency Walker
    Hex Editor
    Identifier
    Import Adder
    Quick Disassembler
    Rebuilder
    Resource Editor
    UPX Utility
```

```
Resource Directory
    Resource Directory Entry 1, ID: 3, AKA: Icons
    Resource Directory Entry 2, ID: 4, AKA: Menus
    Resource Directory Entry 3, ID: 5, AKA: Dialogs
    Resource Directory Entry 4, ID: 6, AKA: String Tables
    Resource Directory Entry 5, ID: 9, AKA: Accelerators
    Resource Directory Entry 6, ID: 14, AKA: Icon Groups
    Resource Directory Entry 7, ID: 16, AKA: Version Info
    Resource Directory Entry 8, ID: 24, AKA: Configuration Files
```

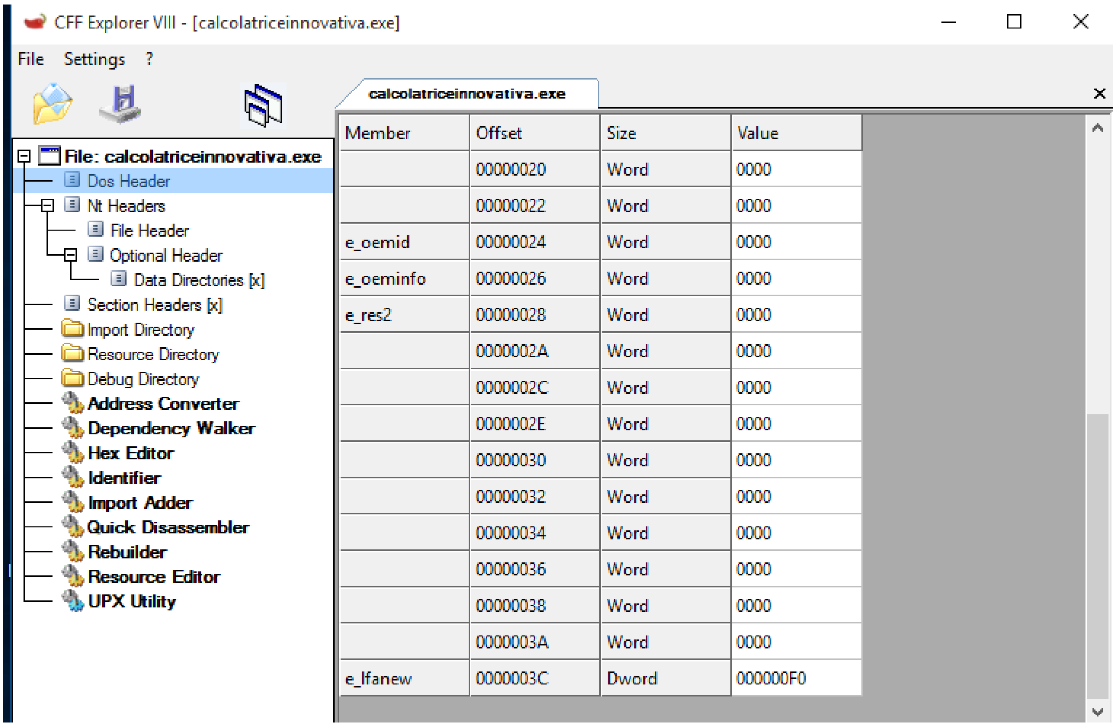

CFF Explorer VIII - [calcolatriceinnovativa.exe]
File
Settings
?
calcolatriceinnovativa.exe

```
File: calcolatriceinnovativa.exe
    Dos Header
    Nt Headers
        File Header
        Optional Header
            Data Directories [x]
    Section Headers [x]
    Import Directory
    Resource Directory
    Debug Directory
    Address Converter
    Dependency Walker
    Hex Editor
    Identifier
    Import Adder
    Quick Disassembler
    Rebuilder
    Resource Editor
    UPX Utility
```

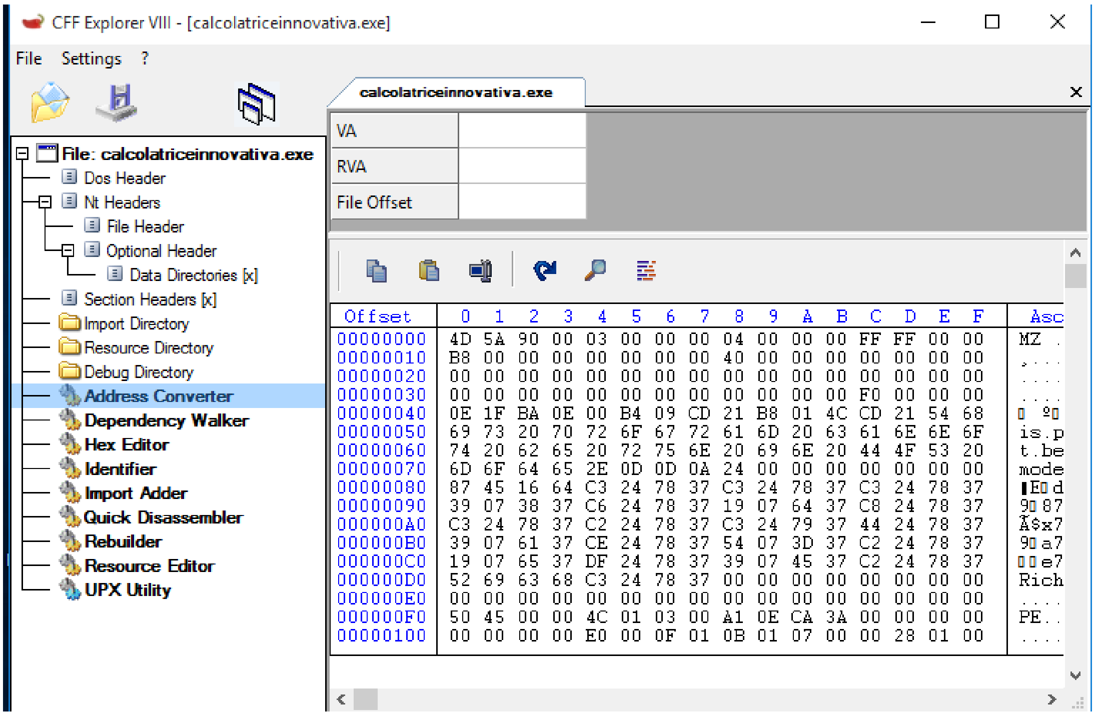
```
Icons
    1 - [lang:1049]
    2 - [lang:1049]
    3 - [lang:1049]
    4 - [lang:1049]
    5 - [lang:1049]
    6 - [lang:1049]
    7 - [lang:1049]
    8 - [lang:1049]
Menus
    106 - [lang:1049]
    107 - [lang:1049]
    108 - [lang:1049]
    109 - [lang:1049]
Dialogs
    101 - [lang:1049]
    102 - [lang:1049]
    103 - [lang:1049]
    104 - [lang:1049]
String Tables
Accelerators
Icon Groups
Version Info
Configuration Files
```



Member	Offset	Size	Value
e_magic	00000000	Word	5A4D
e_cblp	00000002	Word	0090
e_cp	00000004	Word	0003
e_crlc	00000006	Word	0000
e_cparhdr	00000008	Word	0004
e_minalloc	0000000A	Word	0000
e_maxalloc	0000000C	Word	FFFF
e_ss	0000000E	Word	0000
e_sp	00000010	Word	00B8
e_csum	00000012	Word	0000
e_ip	00000014	Word	0000
e_cs	00000016	Word	0000
e_lfarlc	00000018	Word	0040
e_ovno	0000001A	Word	0000
e_res	0000001C	Word	0000
	0000001E	Word	0000

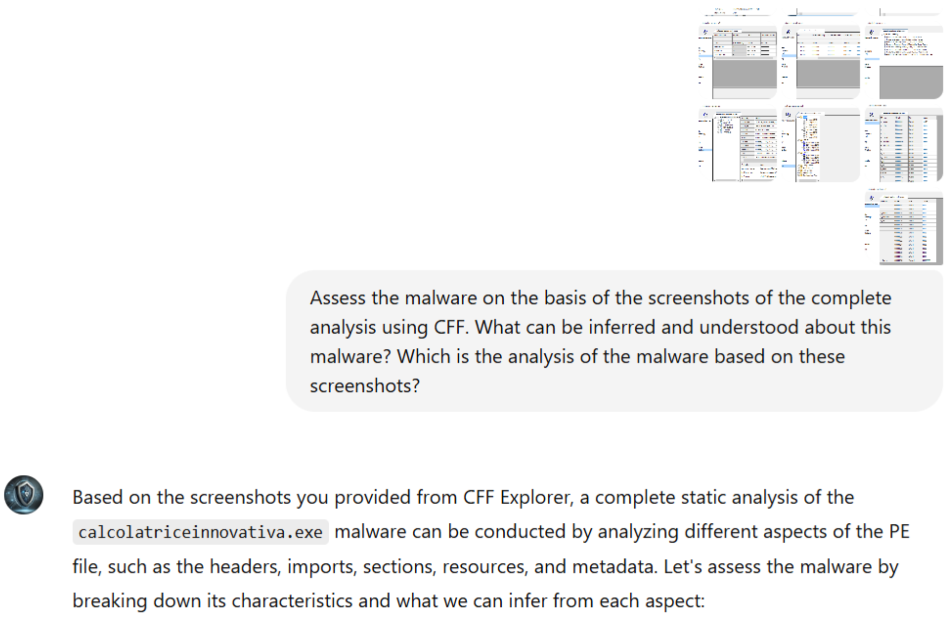


Member	Offset	Size	Value
	00000020	Word	0000
	00000022	Word	0000
e_oemid	00000024	Word	0000
e_oeminfo	00000026	Word	0000
e_res2	00000028	Word	0000
	0000002A	Word	0000
	0000002C	Word	0000
	0000002E	Word	0000
	00000030	Word	0000
	00000032	Word	0000
	00000034	Word	0000
	00000036	Word	0000
	00000038	Word	0000
	0000003A	Word	0000
e_lfanew	0000003C	Dword	000000F0



```
Offset    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   Asc
00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ .
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ,...
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....
00000030  00 00 00 00 00 00 00 00 00 00 00 00 F0 00 00 00  ....
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ▯ º▯
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is.p
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t.be
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode
00000080  87 45 16 64 C3 24 78 37 C3 24 78 37 C3 24 78 37  ▌E▯d
00000090  39 07 38 37 C6 24 78 37 19 07 64 37 C8 24 78 37  9▯87
000000A0  C3 24 78 37 C2 24 78 37 C3 24 79 37 44 24 78 37  Ã$x7
000000B0  39 07 61 37 CE 24 78 37 54 07 3D 37 C2 24 78 37  9▯a7
000000C0  19 07 65 37 DF 24 78 37 39 07 45 37 C2 24 78 37  ▯▯e7
000000D0  52 69 63 68 C3 24 78 37 00 00 00 00 00 00 00 00  Rich
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....
000000F0  50 45 00 00 4C 01 03 00 A1 0E CA 3A 00 00 00 00  PE..
00000100  00 00 00 00 E0 00 0F 01 0B 01 07 00 00 28 01 00  ....
```

Ho dato in input a ChatGPT tutti gli screenshot dell'analisi di CFF ed ha prodotto un report preciso di analisi del malware in tutte le sue parti. Riporto questo report ed anche alcune considerazioni aggiuntive riguardo al malware, elaborate da ChatGPT.



Assess the malware on the basis of the screenshots of the complete analysis using CFF. What can be inferred and understood about this malware? Which is the analysis of the malware based on these screenshots?

Based on the screenshots you provided from CFF Explorer, a complete static analysis of the `calcolatriceinnovativa.exe` malware can be conducted by analyzing different aspects of the PE file, such as the headers, imports, sections, resources, and metadata. Let's assess the malware by breaking down its characteristics and what we can infer from each aspect:

1. File Properties and Metadata

- **File Type**: Portable Executable (PE32), typical for Windows executables.
- **File Size**: 112.50 KB (115200 bytes), small enough to potentially avoid detection by some security software but still capable of containing significant malicious functionality.
- **Company Name and File Description**:
  - **Company Name**: "Корпорация Майкрософт" (Microsoft Corporation).
  - **File Description**: "Калькулятор для Windows" (Calculator for Windows).
  - **FileVersion**: 5.1.2600.0 (common for older Windows XP executables).
  - **Internal Name**: `CALC`.

Inference:

- The metadata suggests the malware is **masquerading as a legitimate Microsoft file**, specifically a calculator application. This tactic of **imitation** or **file forgery** is a common approach in malware to reduce suspicion.
- However, the absence of matching information in the **File Info** field (which reads "No match found") suggests the file might be **manually altered**, and its signature is not recognized by standard databases.

2. Sections of the Malware

- The malware contains three standard PE sections:
  - **.text**: Contains the executable code of the malware.
  - **.data**: Stores the initialized data (variables, configurations) used during execution.
  - **.rsrc**: Stores resources such as icons, menus, dialogs, and other embedded files.

Inference:

- The structure is typical of legitimate applications, which helps disguise its malicious nature.
- The **.rsrc** section containing menus, icons, and dialogs implies that the malware is **designed to mimic a legitimate Windows application**, possibly presenting a user interface to the victim.
- Further inspection of the **.rsrc** section could reveal if any **additional payloads or malicious scripts** are hidden here (e.g., secondary malware, encrypted data, or configuration files).

3. **Import Directory (DLL Imports)**

The malware imports several common Windows libraries:

- **SHELL32.dll**: Used for file manipulation and system interaction.
- **msvcrt.dll**: Provides standard C runtime functions (memory management, file handling).
- **ADVAPI32.dll**: Handles advanced API calls, including registry manipulation and security-related functions.

Inference:

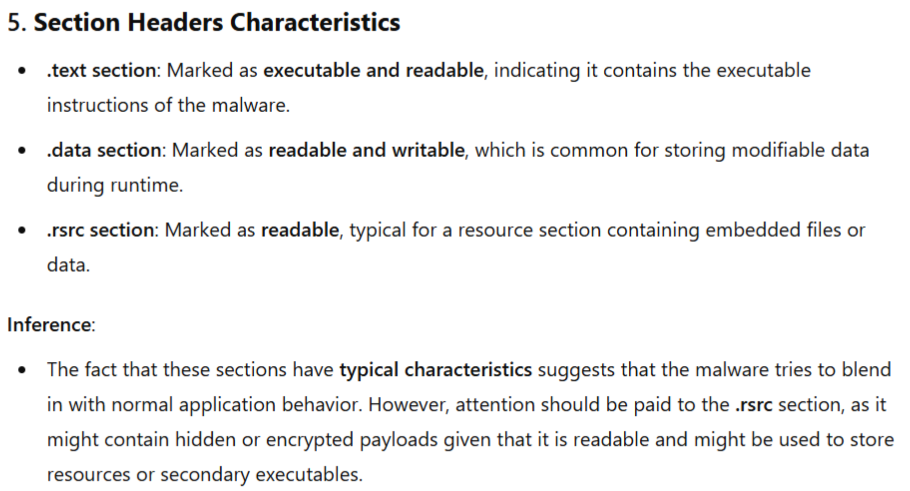
- The import of **SHELL32.dll** suggests that the malware may be **interacting with the file system**, such as creating, modifying, or deleting files.
- The presence of **ADVAPI32.dll** indicates potential **interaction with the Windows registry**, which could be used for **persistence** (e.g., ensuring the malware runs at startup) or for manipulating system permissions or security tokens.
- The combination of these imports implies that the malware has **capabilities to modify system configurations** and **persist** across reboots.

4. **Resource Directory**

- The **Resource Directory** contains various entries including:
  - Icons, menus, dialogs, string tables, version information, and configuration files.

Inference:

- The presence of icons, menus, and dialogs suggests the malware is designed to **mimic a legitimate application** like a calculator (as seen in the metadata). This could be used to deceive users into thinking it is harmless software.
- However, the presence of **version info and configuration files** might indicate **additional hidden functionality**. Malware often uses the resource section to store **configuration data, encrypted payloads**, or even **executable code** to unpack and run dynamically.
- This could suggest the malware is **hiding its core payload or instructions in the resource section** and will deploy them after certain conditions are met.

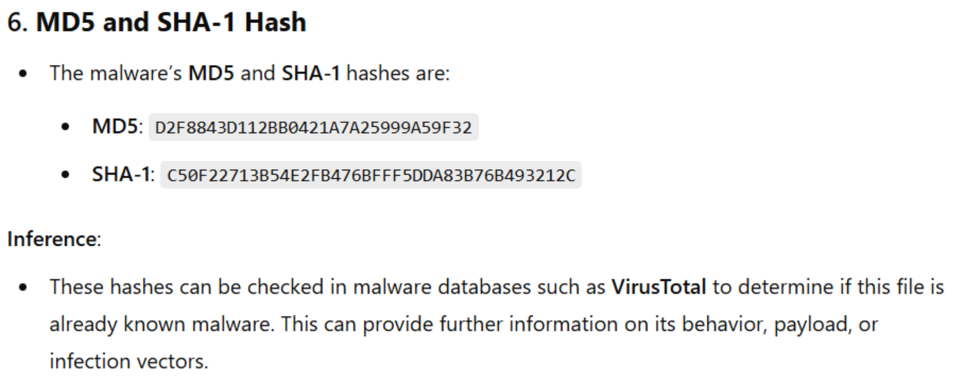


5. Section Headers Characteristics

- **.text section**: Marked as **executable and readable**, indicating it contains the executable instructions of the malware.
- **.data section**: Marked as **readable and writable**, which is common for storing modifiable data during runtime.
- **.rsrc section**: Marked as **readable**, typical for a resource section containing embedded files or data.

Inference:

- The fact that these sections have **typical characteristics** suggests that the malware tries to blend in with normal application behavior. However, attention should be paid to the **.rsrc** section, as it might contain hidden or encrypted payloads given that it is readable and might be used to store resources or secondary executables.

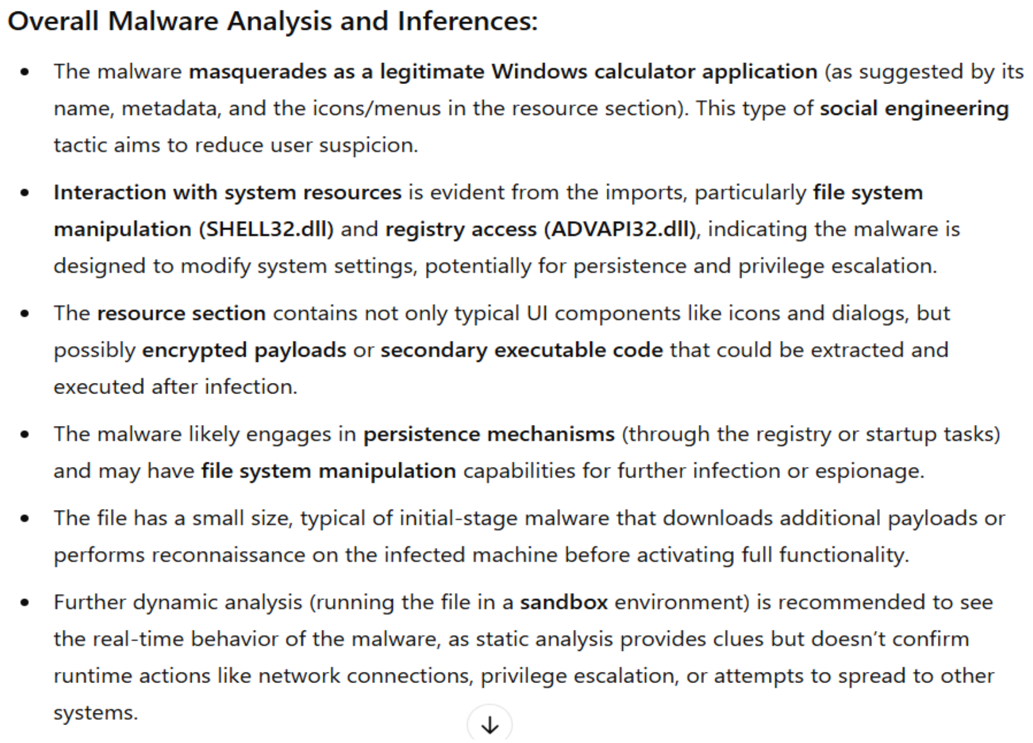


6. MD5 and SHA-1 Hash

- The malware's **MD5** and **SHA-1** hashes are:
  - **MD5**: `D2F8843D112BB0421A7A25999A59F32`
  - **SHA-1**: `C50F22713B54E2FB476BFFF5DDA83B76B493212C`

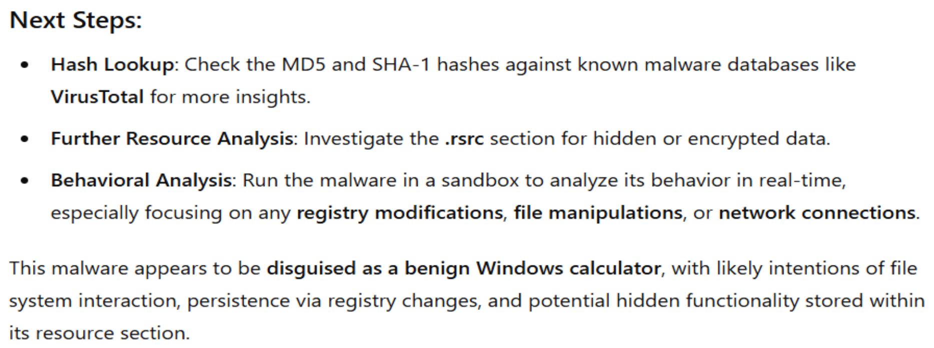
Inference:

- These hashes can be checked in malware databases such as **VirusTotal** to determine if this file is already known malware. This can provide further information on its behavior, payload, or infection vectors.



Overall Malware Analysis and Inferences:

- The malware **masquerades as a legitimate Windows calculator application** (as suggested by its name, metadata, and the icons/menus in the resource section). This type of **social engineering** tactic aims to reduce user suspicion.
- **Interaction with system resources** is evident from the imports, particularly **file system manipulation (SHELL32.dll)** and **registry access (ADVAPI32.dll)**, indicating the malware is designed to modify system settings, potentially for persistence and privilege escalation.
- The **resource section** contains not only typical UI components like icons and dialogs, but possibly **encrypted payloads** or **secondary executable code** that could be extracted and executed after infection.
- The malware likely engages in **persistence mechanisms** (through the registry or startup tasks) and may have **file system manipulation** capabilities for further infection or espionage.
- The file has a small size, typical of initial-stage malware that downloads additional payloads or performs reconnaissance on the infected machine before activating full functionality.
- Further dynamic analysis (running the file in a **sandbox** environment) is recommended to see the real-time behavior of the malware, as static analysis provides clues but doesn't confirm runtime actions like network connections, privilege escalation, or attempts to spread to other systems.



Next Steps:

- **Hash Lookup**: Check the MD5 and SHA-1 hashes against known malware databases like **VirusTotal** for more insights.
- **Further Resource Analysis**: Investigate the **.rsrc** section for hidden or encrypted data.
- **Behavioral Analysis**: Run the malware in a sandbox to analyze its behavior in real-time, especially focusing on any **registry modifications**, **file manipulations**, or **network connections**.

This malware appears to be **disguised as a benign Windows calculator**, with likely intentions of file system interaction, persistence via registry changes, and potential hidden functionality stored within its resource section.

Da un'attenta analisi del malware emerge la sua capacità di mascherarsi da software legittimo per il calcolo di Microsoft, simulando una calcolatrice grazie alle icone e al menu e quindi agendo in modo da trarre in inganno l'utente per fargli eseguire il file. È un malware potente perché interagisce con il sistema, riesce a modificare i registri per essere persistente, potrebbe manipolare i file e contiene altre risorse, come altri payload che potrebbero essere eseguiti in un secondo momento. La sezione .text potrebbe contenere anche del codice compresso e zippato e il codice stesso potrebbe essere offuscato per essere meno riconoscibile e rendere più difficile l'analisi. Il file è piccolo e potrebbe rendere difficile il rilevamento da parte di un software anti malware e potrebbe anche essere in grado di rilevare di essere eseguito su una macchina virtuale per implementare misure di offuscamento del codice. Ha anche funzionalità di accesso remoto. Il malware, come già esaminato, è in grado in interagire con l'utente e potrebbe anche catturare l'input da tastiera con DirectInput. Il malware comunica con un indirizzo IP segnalato come malevolo ed utilizza tecniche MITRE ATT&CK per l'offuscamento delle informazioni e dei file, la cattura di input, il rilevamento di software di sicurezza e la cattura di informazioni riguardo al sistema infettato. Il malware è quindi pericoloso, perché ha accesso al sistema da remoto, raccoglie dati dagli input dell'utente e riguardo al sistema ed ha ottime tecniche di offuscamento per nascondersi ai sistemi di sicurezza e per non far insospettire l'utente finale. La connessione con Metasploit potrebbe far pensare che questo malware sia parte di un attacco più ampio e le scritte in russo fanno ipotizzare che il malware sia realizzato in Russia.