

Analysis Hidden Malware in .exe File

Amin El Kassimi

CyberSecurity EN
Paolo Rampino
Feb 1, 2026

Executive summary

- Indicare le librerie importate dal malware, fornendo una descrizione per ognuna di esse
- Indicare le sezioni di cui si compone il malware, fornendo una descrizione per ognuna di essa

Traccia:

Rispondere ai seguenti quesiti, con riferimento al file eseguibile **notepad-classico.exe** contenuto in

questo **file compresso** (password: infected):

<https://drive.google.com/file/d/1HNnJDSY7FbD1KHfiRzA2wVNHzTJndUD/view?usp=sharing>

- Indicare le **librerie importate** dal malware, fornendo una **descrizione** per ognuna di esse tramite AI;
- Indicare le **sezioni** di cui si compone il malware, fornendo una **descrizione** per ognuna di essa tramite AI.

Suggerimento: ChatGPT (o altri LLM) possono ricevere in input degli screenshot da analizzare e cerca librerie caricate dinamicamente nei testi del codice.

Tools CFF Explorer & PEStudio

Differenza concettuale tra i due tool

🔍 CFF Explorer

È un **PE parser strutturale**.

- Mostra **i dati grezzi del formato PE**
- Zero interpretazione
- Zero giudizi
- Zero "alert"

Ti dice:

"Questo file è *fatto così*"

È come leggere la TAC di un corpo: numeri, sezioni, offset.

🔔 PEStudio

È un **analizzatore statico orientato alla sicurezza**.

- Interpreta i dati
- Evidenzia **pattern sospetti**
- Aggiunge **contesto semantico**
- Fa correlazioni

Ti dice:

"Questa cosa *potrebbe significare qualcosa*"

È come un medico che legge la TAC e dice: "questo valore è fuori scala".

Librerie Importate dal Malware

Queste librerie indicano che il malware interagisce con elementi dell'interfaccia utente, stampa, grafica e operazioni di sistema di basso livello, il che è coerente con un programma che si maschera da editor di testo come Notepad ma potrebbe eseguire azioni maligne.

library (9)	flag (0)	type	imports (201)	description
comdlg32.dll	-	Implicit	9	Common Dialogs Library
SHELL32.dll	-	Implicit	4	Windows Shell Library
WINSPOOL.DRV	-	Implicit	3	Windows Spooler Driver
COMCTL32.dll	-	Implicit	1	Common Controls Library
msvcr7.dll	-	Implicit	22	Microsoft C Runtime Library
ADVAPI32.dll	-	Implicit	7	Advanced Windows 32 Base API
KERNEL32.dll	-	Implicit	57	Windows NT BASE API Client
GDI32.dll	-	Implicit	24	GDI Client Library
USER32.dll	-	Implicit	74	Multi-User Windows USER API Client Library

Libreria	Numero di Importazioni	Descrizione
comdlg32.dll	9	Libreria per i dialoghi comuni di Windows. Fornisce funzioni per finestre di dialogo standard come "Apri file", "Salva file", "Stampa" e selezione colori/font. In un malware, potrebbe essere usata per interagire con file utente o mascherare operazioni di lettura/scrittura.
SHELL32.dll	4	Libreria della shell di Windows. Gestisce operazioni sul file system, come la creazione di collegamenti, l'accesso al desktop e l'integrazione con Explorer. Un malware potrebbe sfruttarla per persistenza (es. aggiungere shortcut) o per navigare nel sistema.

Libreria	Numero di Importazioni	Descrizione
WINSPOOL.DRV	3	Driver per il servizio di spooler di stampa di Windows. Controlla le code di stampa e le interazioni con stampanti. In contesto malware, potrebbe essere usata per inviare dati rubati via "stampa" o per sfruttare vulnerabilità in reti condivise.
COMCTL32.dll	1	Libreria per controlli comuni dell'interfaccia utente. Include elementi UI come barre di progresso, liste, alberi e tab. Utile per creare interfacce grafiche; un malware potrebbe usarla per simulare un'app legittima come Notepad.
msvcrt.dll	22	Libreria runtime del Microsoft Visual C++. Fornisce funzioni base per gestione memoria, stringhe, input/output e matematica. Essenziale per molti programmi C/C++; in malware, supporta operazioni di basso livello come manipolazione di buffer o crittografia.
ADVAPI32.dll	7	API avanzate di Windows 32-bit. Gestisce registro di sistema, servizi di sicurezza, crittografia e eventi di log. Un malware la usa spesso per elevare privilegi, modificare il registro per persistenza o accedere a credenziali.
KERNEL32.dll	57	API base del kernel NT di Windows. Copre gestione processi/thread, memoria, file I/O e sincronizzazione. È la libreria più fondamentale; in malware, abilita caricamento dinamico di codice, iniezione in processi o evasione di detection.
GDI32.dll	24	Graphics Device Interface. Gestisce disegno grafico, font, bitmap e output su schermo/stampante. Per un'app come Notepad, serve per rendering testo; un malware potrebbe usarla per screenshot o overlay grafici ingannevoli.

Libreria	Numero di Importazioni	Descrizione
USER32.dll	74	API utente per interfaccia grafica. Controlla finestre, messaggi, menu, clipboard e input (tastiera/mouse). Alta dipendenza indica un'app GUI; in malware, potrebbe catturare keystroke o manipolare finestre per phishing.

Dds

property	value	value	value	value	value	value	value
section	section[0]	section[1]	section[2]	section[3]	section[4]	section[5]	section[6]
name	.text	.data	.rsrc	.text	.idata	.rsrc	n/a
section > unknown	n/a	n/a	n/a	n/a	n/a	n/a	n/a
entropy	6.214	1.149	5.421	6.428	5.439	5.407	
file > ratio (99.65%)	10.62 %	0.71 %	12.57 %	61.59 %	1.59 %	12.57 %	
raw-address (begin)	0x00000400	0x00007C00	0x00008400	0x00011200	0x0003CA00	0x0003DC00	
raw-address (end)	0x00007C00	0x00008400	0x00011200	0x0003CA00	0x0003DC00	0x00046A00	
raw-size (288256 bytes)	0x00007800 (30720 bytes)	0x00000800 (2048 bytes)	0x00008E00 (36352 bytes)	0x0002B800 (178176 bytes)	0x0001200 (4608 bytes)	0x00008E00 (36352 bytes)	
virtual-address (begin)	0x00001000	0x00009000	0x0000B000	0x00140000	0x00400000	0x00042000	
virtual-address (end)	0x00008748	0x0000AAB8	0x00013DB4	0x0003F6AC	0x004113E	0x0004ADBD	
virtual-size (292414 bytes)	0x00007748 (30536 bytes)	0x00001BA8 (7080 bytes)	0x00008DB4 (36276 bytes)	0x0002B6AC (177836 bytes)	0x0000113E (4414 bytes)	0x00008DB0 (36272 bytes)	

Cosa mostra PEStudio (livello corretto di lettura)

Questa tabella incrocia **struttura + permessi + comportamento potenziale**.

Le colonne importanti sono:

- **entropy**
- **raw size / virtual size**
- **characteristics**
- **write / execute**
- **self-modifying**

Queste non sono decorative. Sono **indizi comportamentali statici**.

Analisi delle sezioni

◊ .text (section[0] e section[3])

Osservazioni:

- **Execute = sì**
- **Write = no**
- **Entropy ~6.2 – 6.4**
- Entry point nella .text

Interpretazione:

- Codice eseguibile normale
- Entropia media → **non cifrato / non compresso**
- Nessuna scrittura → **codice statico**

☞ Questo è il comportamento che *ti aspetti* da software legittimo.

Descrizione da report

La sezione .text contiene il codice eseguibile del programma.

I permessi sono coerenti (execute senza write) e l'entropia moderata suggerisce l'assenza di packing o cifratura del codice.

◊ .data

Osservazioni:

- **Write = sì**
- **Execute = no**
- Entropy **molto bassa (1.149)**

Interpretazione:

- Dati inizializzati
- Nessuna offuscazione
- Uso standard

Descrizione

La sezione .data contiene dati inizializzati e scrivibili utilizzati a runtime.

La bassa entropia indica dati in chiaro e un utilizzo conforme agli standard di un eseguibile Windows legittimo.

❖ .rdata

Osservazioni:

- Read-only
- Entropy ~5.4
- Nessun permesso di scrittura o esecuzione

Interpretazione:

- Stringhe, costanti, tavole
- Nulla di sospetto

Descrizione

La sezione .rdata contiene dati in sola lettura, come stringhe e tavole.

Le caratteristiche risultano coerenti con un utilizzo standard.

❖ .idata

Osservazioni:

- Dati legati alle importazioni
- Write = sì (normale)
- Entropy bassa (~1.59)

Interpretazione:

- Import Address Table
- Fondamentale per capire **le capacità**, non sospetta di per sé

Descrizione

La sezione .idata contiene le informazioni relative alle librerie e alle funzioni importate dall'eseguibile, utilizzate dal loader di Windows durante l'inizializzazione del processo.

◊ .rsrc

Osservazioni:

- Entropy ~5.4
- Read-only
- Nessun execute

Interpretazione:

- Risorse standard (icone, manifest, version info)

Descrizione

La sezione .rsrc contiene le risorse dell'eseguibile, come informazioni di versione e manifest.

Non presenta caratteristiche anomale.

The screenshot shows the CFF Explorer VIII interface. On the left is a tree view of the file structure for 'notepad-classico.exe', including sections like Dos Header, Nt Headers, File Header, Optional Header, Data Directories, and Section Headers. A blue selection bar highlights the 'Section Headers [x]' section. On the right is a table titled 'notepad-classico.exe' showing memory dump details for various sections. The columns include Name, Virtual Size, Virtual Address, Raw Size, Raw Address, Reloc Address, Linenumbers, Relocations N..., Linenumbers ..., and Characteristics. The table lists sections such as .text, .data, .rsrc, and .idata, with their respective memory addresses and sizes.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Word	Word	Word	Dword
.text	00007748	00001000	00007800	00000400	00000000	00000000	0000	0000	60000020
.data	00001BA8	00009000	00000800	00007C00	00000000	00000000	0000	0000	C0000040
.rsrc	00008DB4	0000B000	00008E00	00008400	00000000	00000000	0000	0000	40000040
.text	0002B6AC	00014000	0002B800	00011200	00000000	00000000	0000	0000	E0000020
.idata	0000113E	00040000	00001200	0003CA00	00000000	00000000	0000	0000	C2000040
.rsrc	00008DB0	00042000	00008E00	0003DC00	00000000	00000000	0000	0000	40000040