



25/03/2024

# MALWARE ANALYSIS - INTRO E ANALISI STATICA BASICA

Prepared by:  
Manuel Buonanno

Organized by:



# Traccia

Con riferimento al file eseguibile contenuto nella cartella «Esercizio\_Pratico\_U3\_W2\_L1» presente sul Desktop della vostra macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti:

- 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
- Aggiungere una considerazione finale sul malware in analisi in base alle informazioni raccolte

# Windows 7

Avviata la macchina virtuale Windows 7 con all'interno un malware ci ritroveremo sul desktop queste due cartelle.



Apriamo quindi la cartella <<Software malware analysis>> ed avviamo il programma <<CFF Explorer>>.

Nome	Ultima modifica	Tipo	Dimensione
apateDNS	17/01/2024 16:58	Cartella di file	
ExeinfoPe	17/01/2024 17:22	Cartella di file	
md5deep-4.3	17/01/2024 12:12	Cartella di file	
odbg110	17/01/2024 17:39	Cartella di file	
ProcessExplorer	17/01/2024 15:40	Cartella di file	
ProcessMonitor	17/01/2024 15:36	Cartella di file	
Regshot-1.9.0	17/01/2024 16:55	Cartella di file	
SysinternalsSuite	17/01/2024 17:48	Cartella di file	
CFF Explorer	17/01/2024 14:13	Collegamento	2 KB
IDA Pro Advanced (32-bit)	17/01/2024 17:21	Collegamento	1 KB
IDA Pro Advanced (64-bit)	17/01/2024 17:21	Collegamento	1 KB
OLLYDBG	20/01/2024 18:14	Collegamento	2 KB
Process Hacker 2	17/01/2024 17:54	Collegamento	2 KB

# CFF Explorer

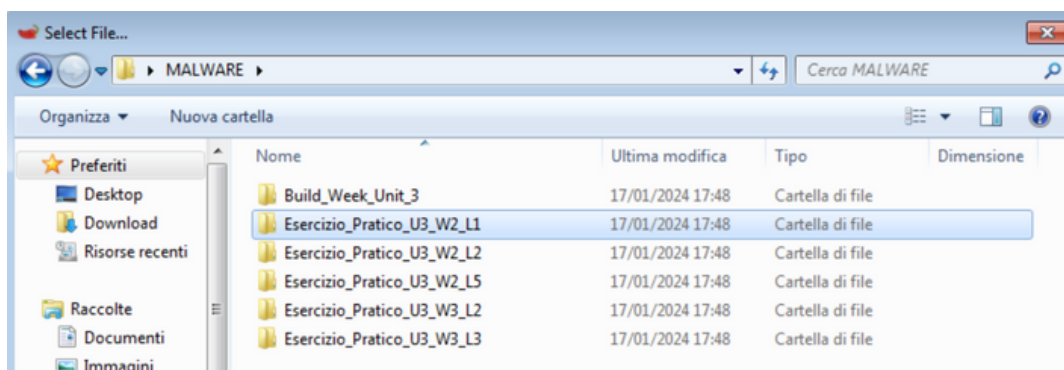
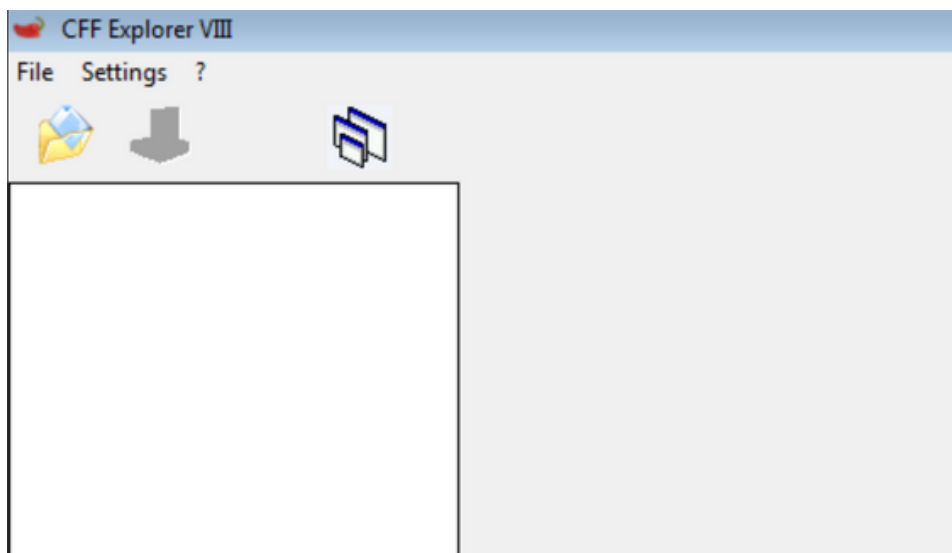
CFF Explorer è un'applicazione software progettata principalmente per esaminare e modificare i file eseguibili di Windows.

Le principali funzionalità di CFF Explorer includono:

- **Esplorazione della struttura PE:** consente agli utenti di esplorare la struttura interna dei file PE;
- **Analisi degli attributi di sicurezza:** permette di visualizzare e analizzare gli attributi di sicurezza dei file eseguibili, inclusi i certificati digitali e le firme digitali.
- **Modifica delle risorse:** consente agli utenti di modificare e manipolare le risorse all'interno dei file PE, come icone, stringhe, manifesti e altro ancora.
- **Patch e modifica binaria:** CFF Explorer permette agli utenti di apportare modifiche dirette al codice binario dei file eseguibili, ad esempio per applicare patch, cambiare valori o inserire nuove istruzioni.
- **Esplorazione di database e directory:** offre funzionalità per esplorare le directory all'interno dei file PE, come la directory di esportazione, importazione, base dei simboli, risorse e altre.
- **Plugin e scripting:** CFF Explorer supporta l'aggiunta di plugin esterni per estendere le sue funzionalità e offre anche la possibilità di automatizzare operazioni tramite scripting.

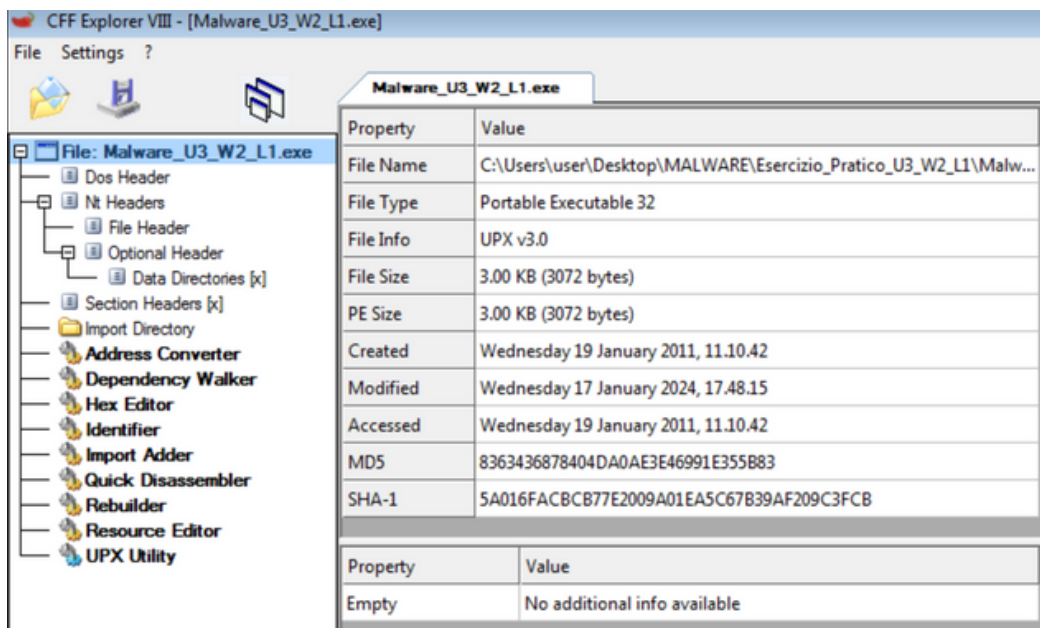
# CFF Explorer: interfaccia

Questa é l'interfaccia principale di CFF Explorer, sulla quale andremo a caricare il file «Esercizio\_Pratico\_U3\_W2\_L1» dalla cartella sul Desktop <<MALWARE>> per andare ad analizzare l'header del PE. Le informazioni circa le librerie e le funzioni richieste dall'eseguibile sono contenute nell'header del formato PE (PortableExecutable). Controllare quali sono le librerie e le funzioni importate è fondamentale per capire lo scopo del malware.



# CFF Explorer: analisi

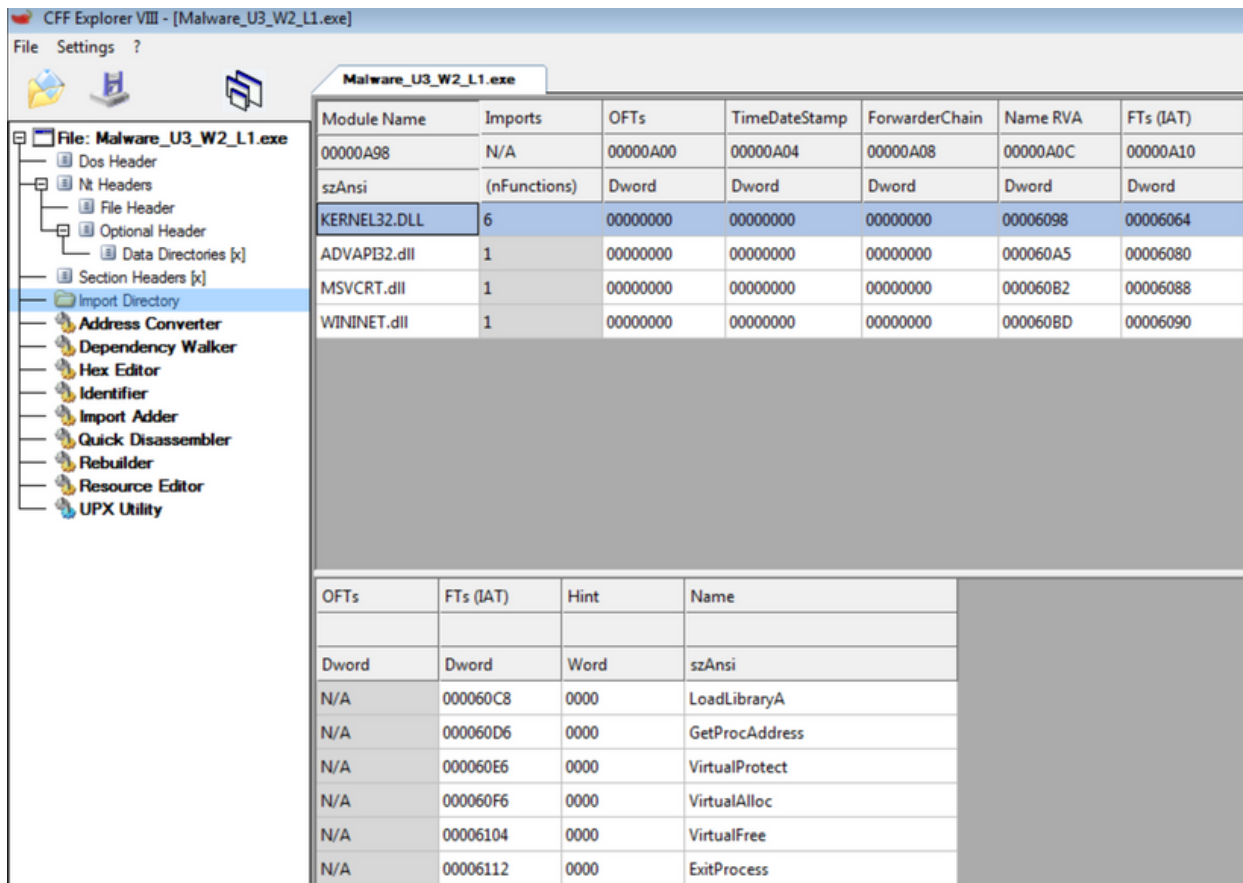
Una volta scelto il file eseguibile per il quale vogliamo esaminare l'header del formato PE, si aprirà una schermata come questa.



Possiamo notare come CFF Explorer analizza tutto il file che abbiamo caricato restituendoci molte informazioni su esso.

# CFF Explorer: import directory

Per controllare le librerie e le funzioni importate, ci spostiamo su «import directory» nel menù a sinistra.



The screenshot shows the CFF Explorer interface for the file 'Malware\_U3\_W2\_L1.exe'. The left sidebar displays the file's structure, with 'Import Directory' selected. The main window displays a table of imported modules and functions.

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
00000A98	N/A	00000A00	00000A04	00000A08	00000A0C	00000A10
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.DLL	6	00000000	00000000	00000000	00006098	00006064
ADVAPI32.dll	1	00000000	00000000	00000000	000060A5	00006080
MSVCRT.dll	1	00000000	00000000	00000000	000060B2	00006088
WININET.dll	1	00000000	00000000	00000000	000060BD	00006090

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
N/A	000060C8	0000	LoadLibraryA
N/A	000060D6	0000	GetProcAddress
N/A	000060E6	0000	VirtualProtect
N/A	000060F6	0000	VirtualAlloc
N/A	00006104	0000	VirtualFree
N/A	00006112	0000	ExitProcess

Il pannello darà informazioni sulle librerie importate dall'eseguibile.

# CFF Explorer: funzioni


Il pannello inferiore mostrerà una lista delle funzioni richieste all'interno della libreria selezionata.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
N/A	000060C8	0000	LoadLibraryA
N/A	000060D6	0000	GetProcAddress
N/A	000060E6	0000	VirtualProtect
N/A	000060F6	0000	VirtualAlloc
N/A	00006104	0000	VirtualFree
N/A	00006112	0000	ExitProcess

Queste sono comunemente utilizzate in malware per eseguire per ottenere il controllo del sistema. Ecco cosa si può dedurre da ciascuna di esse:

- **loadLibraryA:** utilizzata per caricare una libreria dinamicamente nel processo corrente. I malware spesso caricano librerie per eseguire codice aggiuntivo o per evitare la rilevazione da parte degli antivirus.
- **GetProcAddress:** utilizzata per ottenere il puntatore di una funzione esportata da una libreria. I malware possono utilizzare questa funzione per ottenere l'indirizzo di altre funzioni API del sistema o per evitare la rilevazione statica.
- **VirtualProtect:** utilizzata per modificare i permessi di accesso della memoria virtuale. I malware possono utilizzare questa funzione per rendere la memoria eseguibile o scrivibile, per eseguire il proprio codice o per modificare il codice esistente nel processo.



- **VirtualAlloc:** Utilizzata per allocare memoria virtuale all'interno del processo. I malware possono utilizzare questa funzione per allocare spazio per il proprio codice eseguibile o per lo storage di dati.
  - **VirtualFree:** Serve per liberare la memoria virtuale precedentemente allocata. I malware possono utilizzare questa funzione per nascondere le proprie tracce o per liberare la memoria utilizzata dopo aver completato il loro scopo.
  - **ExitProcess:** Utilizzata per terminare il processo corrente. I malware possono utilizzare questa funzione per terminare il proprio processo in modo pulito dopo aver completato le operazioni dannose o per evitare la rilevazione da parte degli strumenti di sicurezza.
- 

# CFF Explorer: librerie

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
00000A98	N/A	00000A00	00000A04	00000A08	00000A0C	00000A10
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.DLL	6	00000000	00000000	00000000	00006098	00006064
ADVAPI32.dll	1	00000000	00000000	00000000	000060A5	00006080
MSVCRT.dll	1	00000000	00000000	00000000	000060B2	00006088
WININET.dll	1	00000000	00000000	00000000	000060BD	00006090

In questo caso il malware che abbiamo analizzato importa 4 librerie:

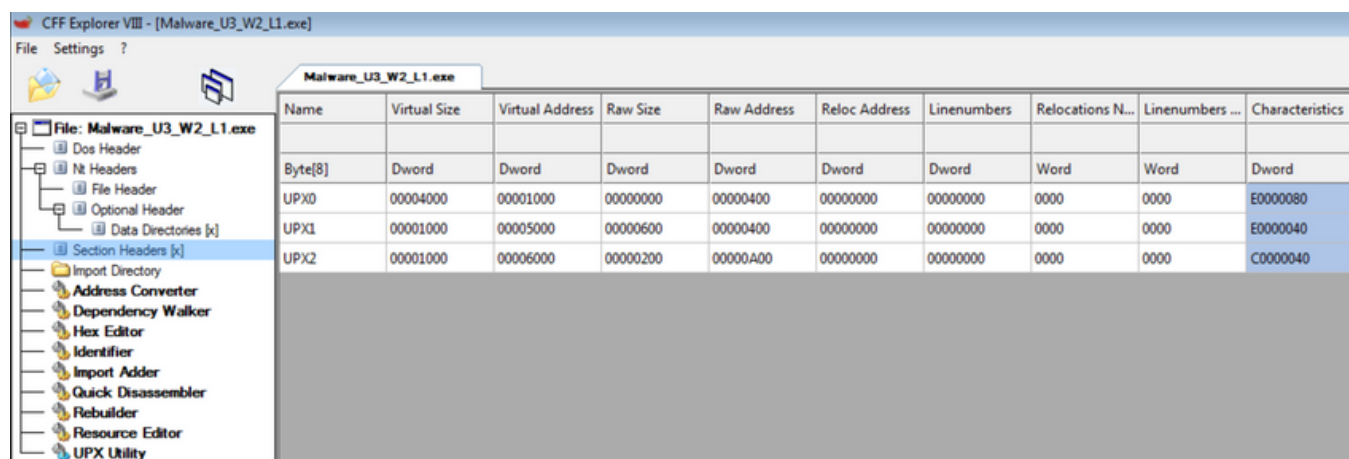
- **Kernel32.dll**, contiene le funzioni principali per interagire con il sistema operativo, ad esempio: manipolazione dei file, la gestione della memoria;
- **Advapi32.dll**, contiene le funzioni per interagire con i servizi ed i registri del sistema operativo;
- **MSVCRT.dll**, contiene funzioni per la manipolazione stringhe, allocazione memoria e altro come chiamate per input/output, come nel linguaggio C;
- **Wininet.dll**, contiene le funzioni per l'implementazione di alcuni protocolli di rete come HTTP, FTP, NTP.

# CFF Explorer: Section headers

L'header del formato PE fornisce molte altre informazioni importanti oltre alle funzioni/librerie importate ed esportate, come ad esempio le sezioni di cui si compone il software. Ogni sezione ha un preciso scopo, e conoscerle è una preziosa informazione per le analisi. Le più comuni ed interessanti sezioni in un file PE sono:

- **.text:** contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato. Generalmente questa è l'unica sezione di un file eseguibile che viene eseguita dalla CPU, in quanto tutte le altre sezioni contengono dati o informazioni a supporto.
- **.rdata:** include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile, informazione che come abbiamo visto possiamo ricavare con CFF Explorer.
- **.data:** contiene tipicamente i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma. Una variabile si dice globale quando non è definita all'interno di un contesto di una funzione, ma bensì è globalmente dichiarata ed è di conseguenza accessibile da qualsiasi funzione all'interno dell'eseguibile.
- **.rsrc:** include le risorse utilizzate dall'eseguibile come ad esempio icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso.

Per controllare le sezioni di un file eseguibile spostiamoci nel pannello a sinistra nella sezione «section headers». Il pannello principale a destra mostrerà le informazioni circa le sezioni di cui si compone l'eseguibile.



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
UPX0	00004000	00001000	00000000	00000400	00000000	00000000	0000	0000	E0000080
UPX1	00001000	00005000	00000600	00000400	00000000	00000000	0000	0000	E0000040
UPX2	00001000	00006000	00000200	00000A00	00000000	00000000	0000	0000	C0000040

La figura riporta non solo il nome delle sezioni ma anche altre importanti informazioni, come ad esempio:

- **Virtual size:** indica lo spazio allocato per la sezione durante il processo di caricamento dell'eseguibile in memoria
- **Rawsize:** indica lo spazio occupato dalla sezione quando è sul disco

# Conclusioni

In questo caso sembra che il malware abbia nascosto il vero nome delle sezioni e non siamo in grado **al momento** di capire che tipo di sezioni sono.

Alcuni malware utilizzano il caricamento delle librerie durante l'esecuzione (*runtimeimport*) nascondendo di fatto all'analisi statica le funzioni e le librerie importate. Questi malware sono riconoscibili in quanto hanno generalmente poche entry nella sezione import, e tra esse figurano le funzioni «LoadLibrary» e «GetProcAddress» che vengono appunto utilizzate per caricare funzioni aggiuntive durante l'esecuzione, nascondendo di fatto le librerie importate a monte.