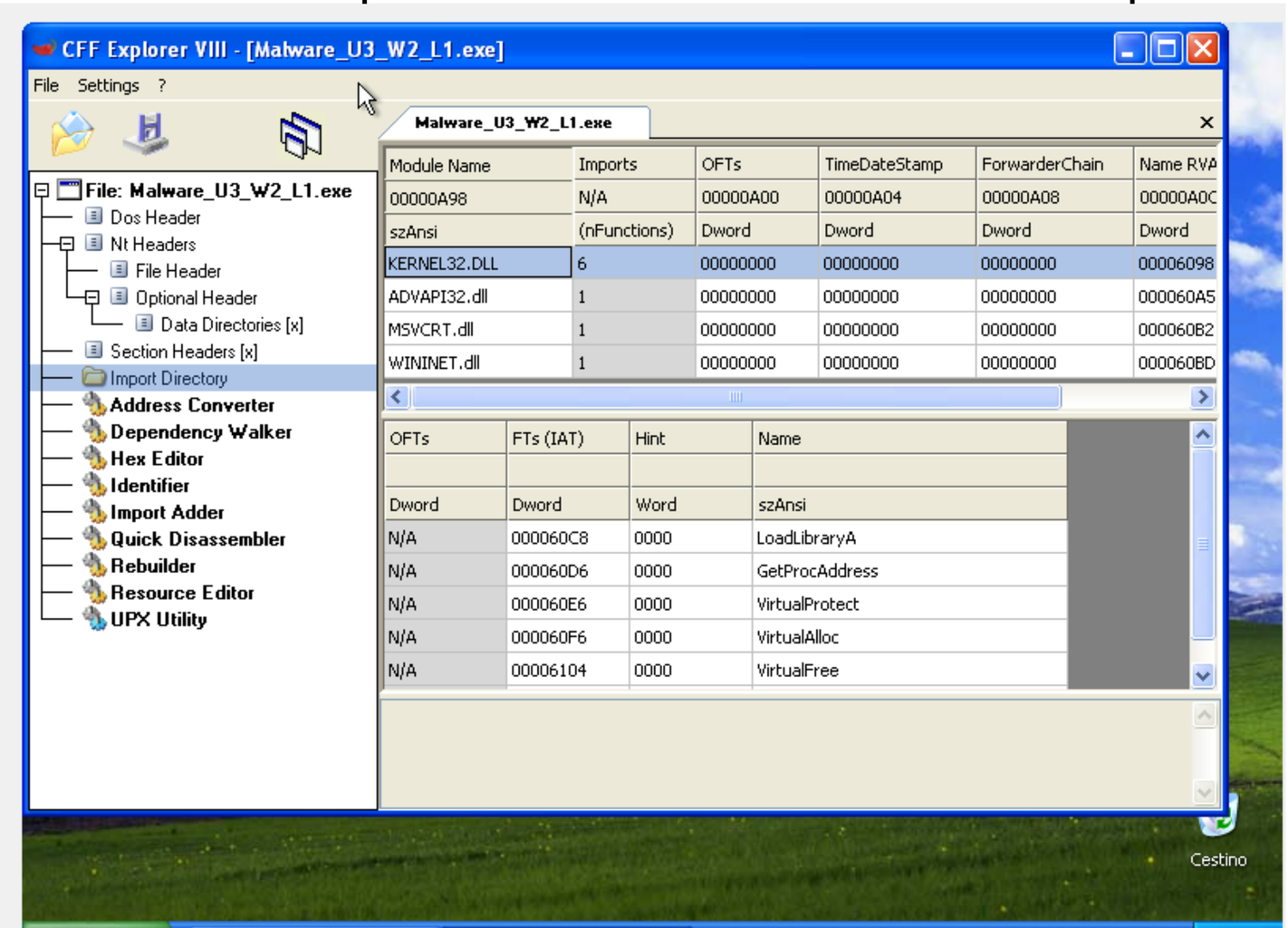
S10 L1 Malware analysis

Oggi siamo andati ad analizzare un malware che si presenta come una PE(PortableExecutable) andiamo subito ad analizzarne il contenuto con CFF Explorer, un potente tool che ci restituisce parecchie informazioni in merito.

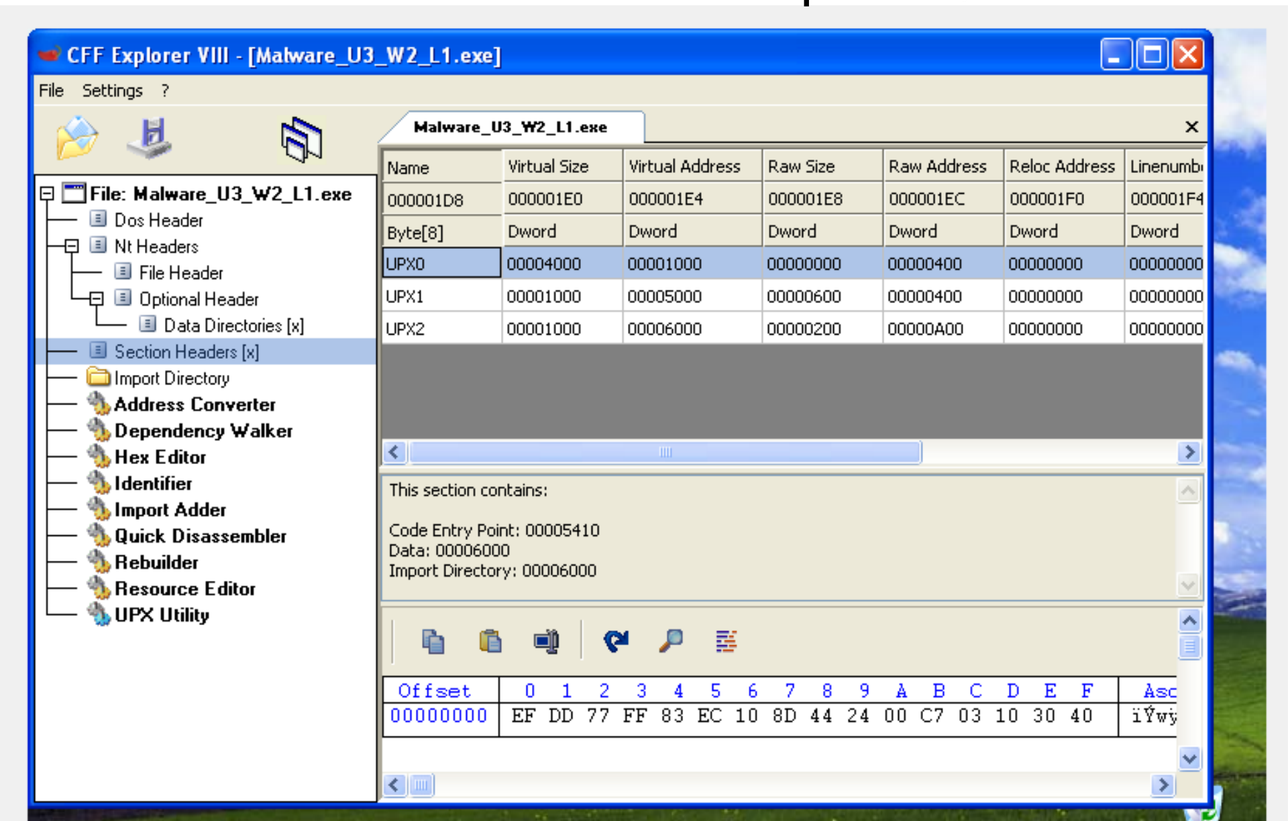


Nell’immagine nello specifico possiamo vedere le librerie richiamate dal malware. Sono presenti le seguenti librerie:

- 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
- WINNINET.dll: è una libreria di Windows che fornisce funzionalità relative alla connettività di rete. È spesso utilizzata per la gestione delle connessioni Internet nei programmi Windows.

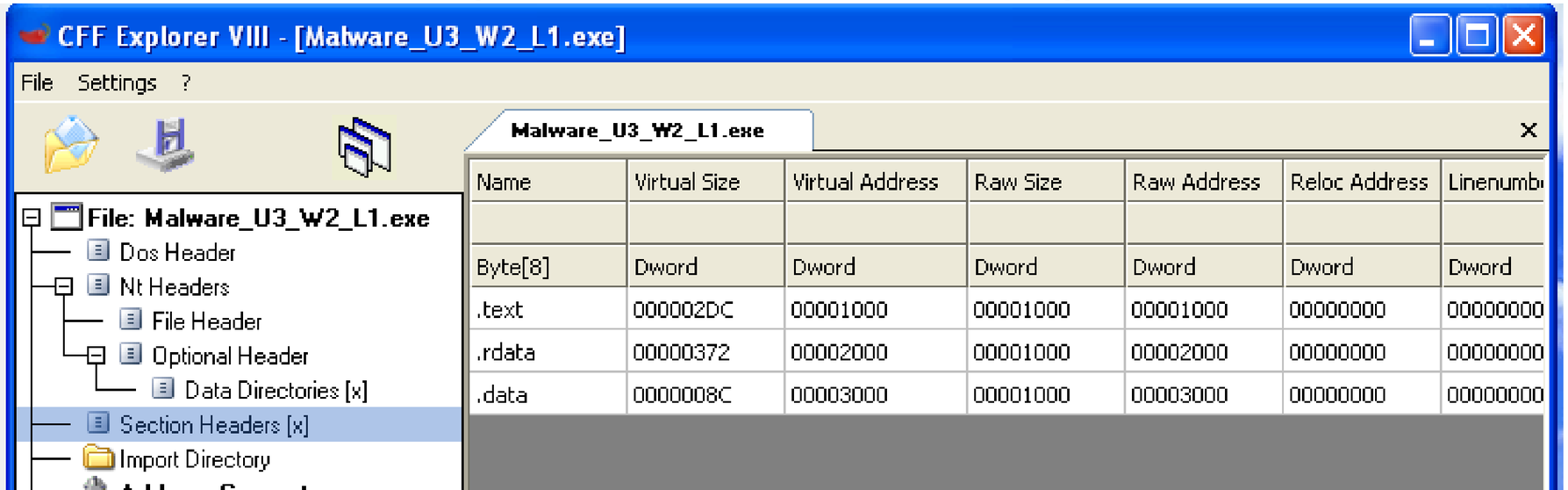
Oltre le librerie presenti nel malware possiamo controllare anche le sezioni da cui è composto, nella sezione «section headers».



Quando UPX viene utilizzato per comprimere un file eseguibile, crea sezioni aggiuntive nel file risultante, etichettate spesso come upx0, upx1, upx2, ecc. Ogni sezione ha un ruolo specifico nel processo di decompressione e nel caricamento del programma nel sistema.

- upx0: Questa sezione contiene il codice UPX decompressione. Durante l'esecuzione del file compresso, questa sezione viene caricata in memoria e il codice qui presente si occupa di decomprimere le sezioni successive.
- upx1, upx2, ecc.: Queste sezioni contengono i dati compressi del programma originale. Una volta che la sezione upx0 ha eseguito la decompressione, i dati vengono ripristinati nella loro forma originale.

Con il nostro tool possiamo anche decomprimerli e andiamo a vedere cosa contengono :



• .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.

Alcuni malware utilizzano ad esempio 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 addizionali durante l’esecuzione.

Ed è proprio il caso del nostro malware, come possiamo vedere dalla prima immagine ha richiamato la funzione “GetProcAddress”.
Quindi abbiamo a che fare con un malware che potrebbe non essere rilevato da un analisi statica semplice, rendendolo pericoloso.