CONSEGNA U3 S10 L1

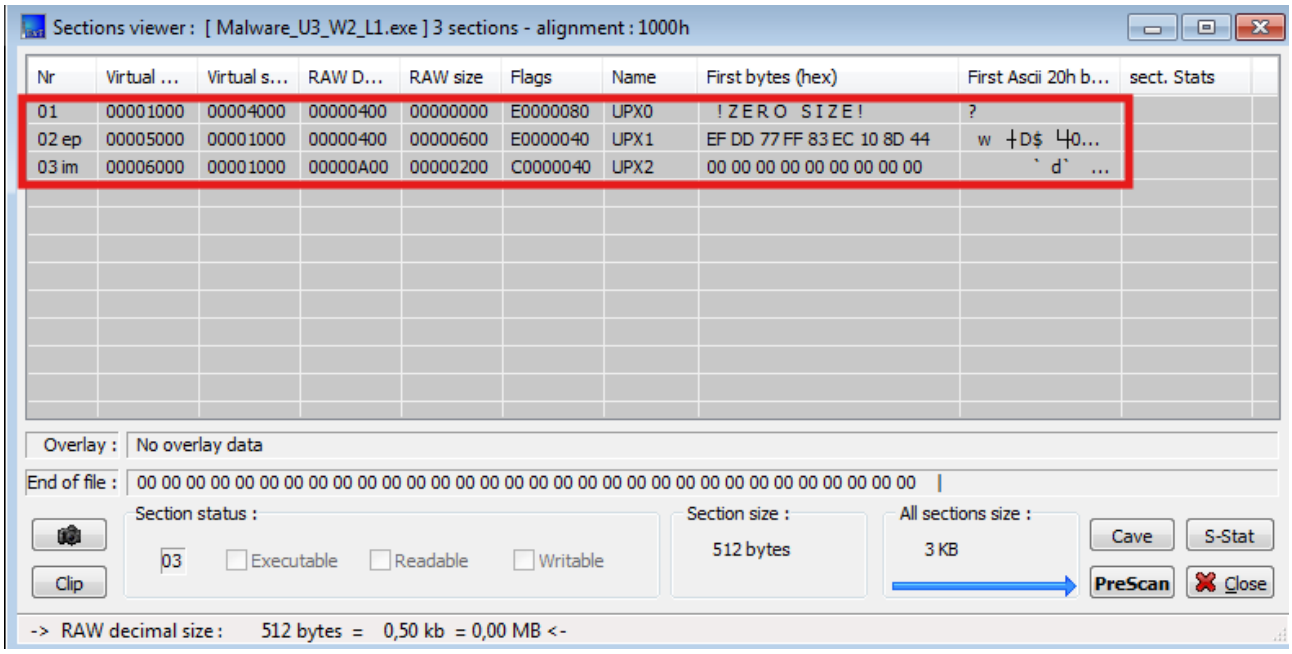
- **Librerie richiamate dal malware**

Malware_U3_W2_L1.exe

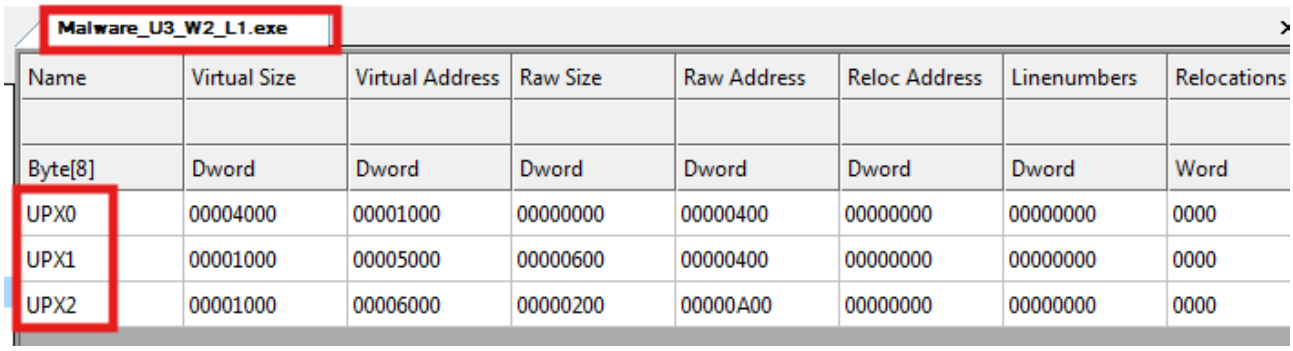
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.DLL	6	00000000	00000000	00000000	00006098	00006064
ADVAPI32.dll	1	00000000	00000000	00000000	000060A5	00006080
MSVCRT.dll	1	00000000	00000000	00000000	000060B2	00006088
WININET.dll	1	00000000	00000000	00000000	000060BD	00006090

1. KERNEL32.dll: è una Dynamic Link Library di Windows che permette il corretto funzionamento del kernel del sistema operativo. Le funzioni al suo interno si occupano della gestione della memoria (volatile), gestione dei processi (creazione e terminazione), dell'input/ output dei file e della gestione errori.
2. ADVAPI32.dll: è una raccolta di funzioni che si occupano di vari servizi di sistema come la sicurezza, l'accesso al registro, la registrazione degli eventi, e la gestione dell'account utente.
3. MSVCRT.dll: è una libreria contenete funzioni relative all'allocazione di memoria, operazioni di input/output dei file e alla gestione delle eccezioni.
4. WININET.dll: insieme di funzioni di importanza cruciale nella gestione delle connessioni internet, relativa ai protocolli HTTP, FTP, HTTPS. È fondamentale anche nella gestione dei cookies e della cache.

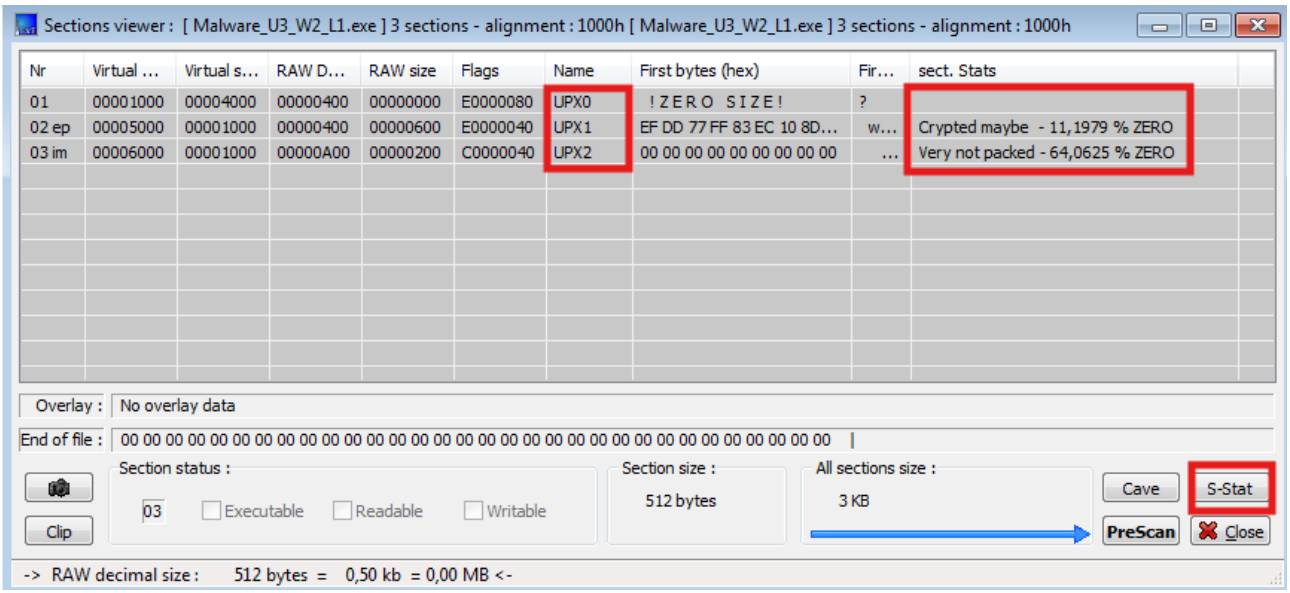
- **Sezioni del malware**



Il malware in analisi si compone di tre sezioni. Per verificarlo ho usato sia **CFFExplorer** che **Exeinfo PE**.

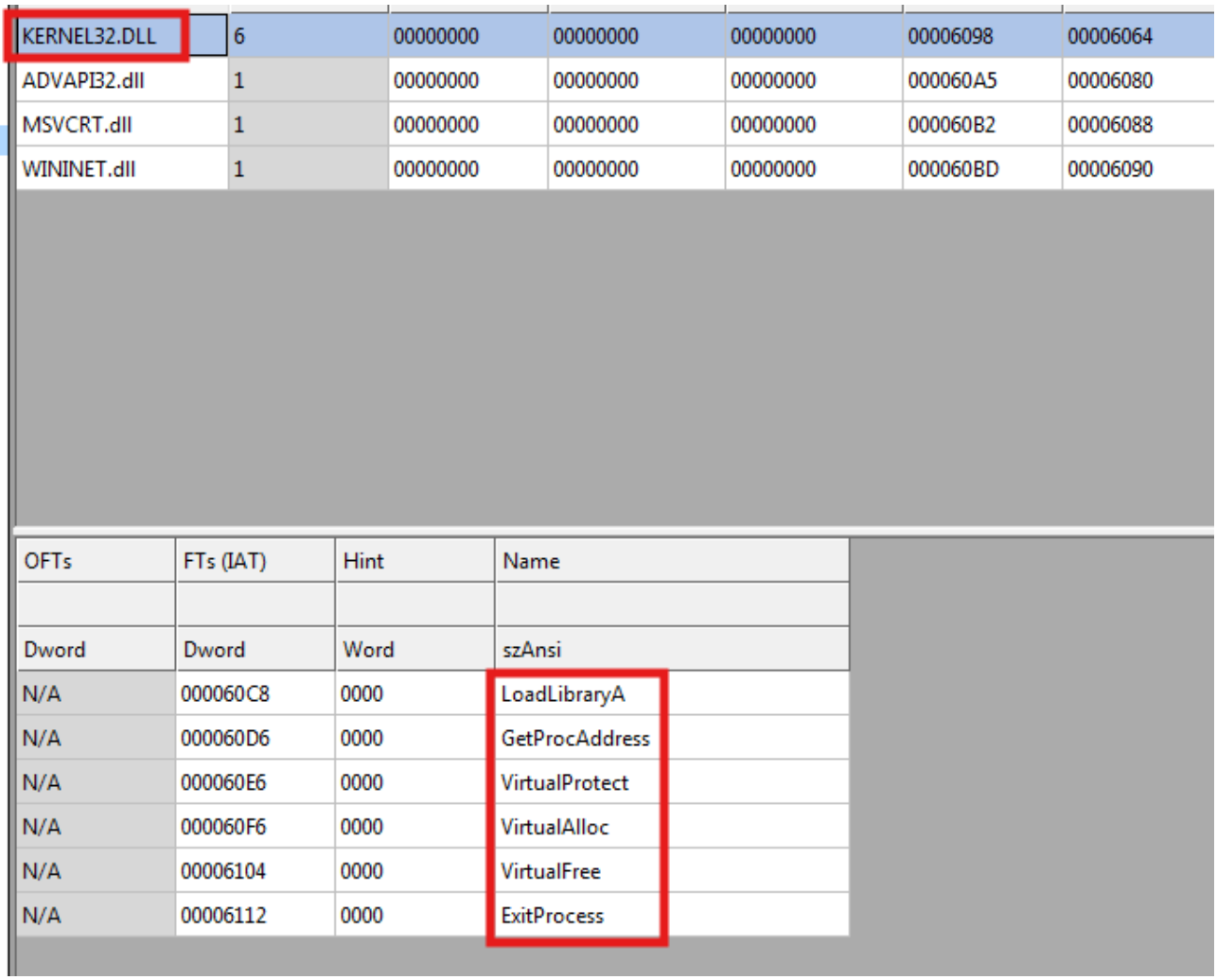


Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word
UPX0	00004000	00001000	00000000	00000400	00000000	00000000	0000
UPX1	00001000	00005000	00000600	00000400	00000000	00000000	0000
UPX2	00001000	00006000	00000200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



Purtroppo il malware ha al suo interno una protezione per nascondere il funzionamento delle varie sezioni, detto **Anti-UPX Unpacking**. Ogni sezione viene quindi nominata con la sigla **UPX** seguita da un numero (ad esempio 0,1,2…). Si tratta di un comune software per la compressione di tipo legittimo per ridurre la dimensione di un file. Spesso però capita che questi tools si occupino di offuscare e criptare il contenuto del malware per rendere più difficile l'analisi statica e il reverse enginering dello stesso.

Ci si accorge di ciò, oltre che dalla colonna **Name,** dove le varie sezioni vengono nominate a partire dalla nomenclatura **UPX0** a salire **(UPX1, UPX2…)** anche da ciò che compare alla colonna **sect. Stat** dopo aver premuto il pulsante **S-Stat**. Alcuni dei più popolari sono **UPX**, **ASPack**, **Themida**, **Exe Packer, MPRESS**, **Morphine**.

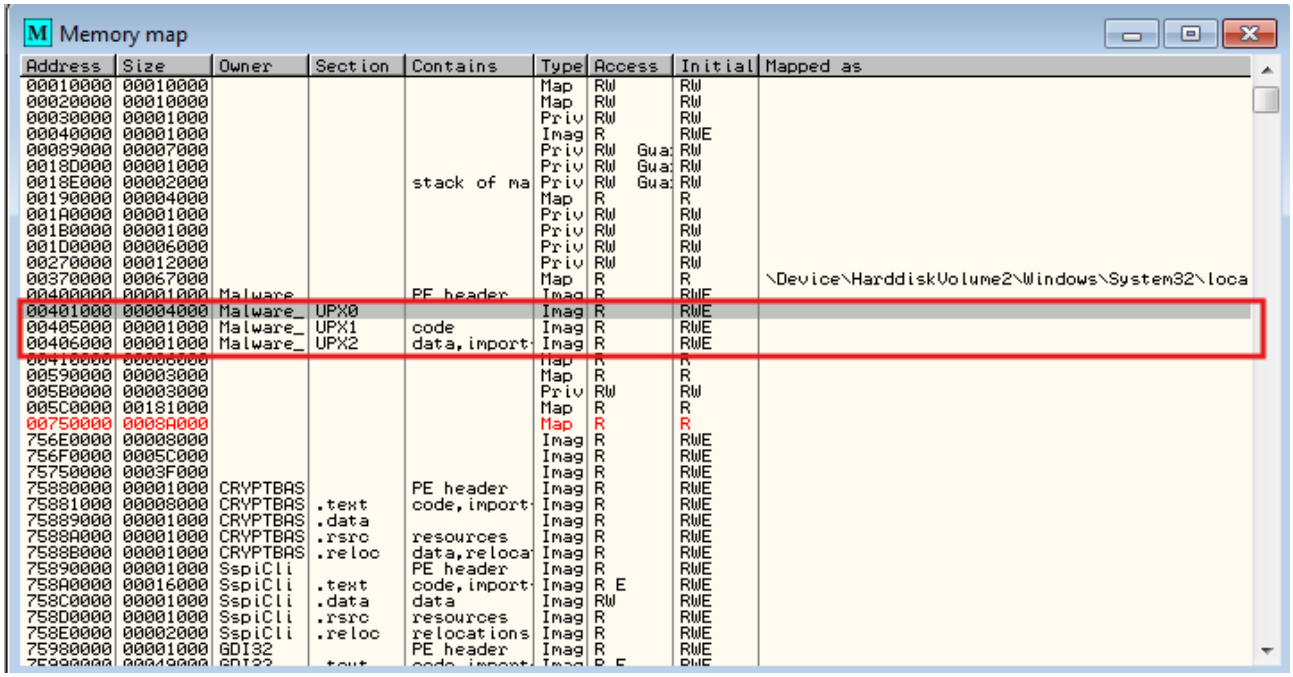


KERNEL32.DLL	6	00000000	00000000	00000000	00006098	00006064
ADVAPI32.dll	1	00000000	00000000	00000000	000060A5	00006080
MSVCRT.dll	1	00000000	00000000	00000000	000060B2	00006088
WININET.dll	1	00000000	00000000	00000000	000060BD	00006090

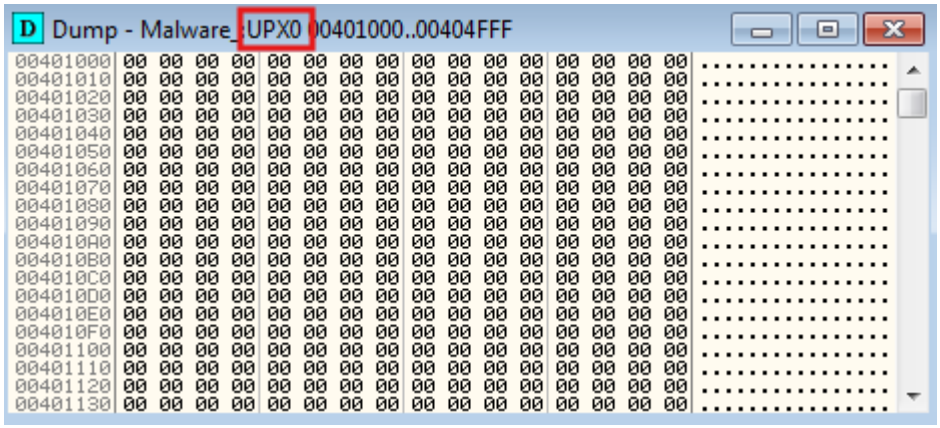
OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
N/A	000060C8	0000	LoadLibraryA
N/A	000060D6	0000	GetProcAddress
N/A	000060E6	0000	VirtualProtect
N/A	000060F6	0000	VirtualAlloc
N/A	00006104	0000	VirtualFree
N/A	00006112	0000	ExitProcess

Un altro segnale del fatto che sia stato usato un metodo di offuscamento, sono alcune funzioni che vengono rilevate tra quelle presenti nelle librerie (ad esempio nella KERNEL32.dll) adoperate, dal software malevolo: **VirtualProtect** (modifica la protezione di un'area di pagine memoria di cui è stato eseguito il commit nello spazio degli indirizzi virtuali del processo che l'ha invocata)**, GetProcAddress** (recupera l'indirizzo di una funzione esportata o variabile della libreria dinamica specificata)**, LoadLibraryA** (si occupa di caricare in memoria il modulo specificato nello spazio indirizzi del processo che la ha chiamata.

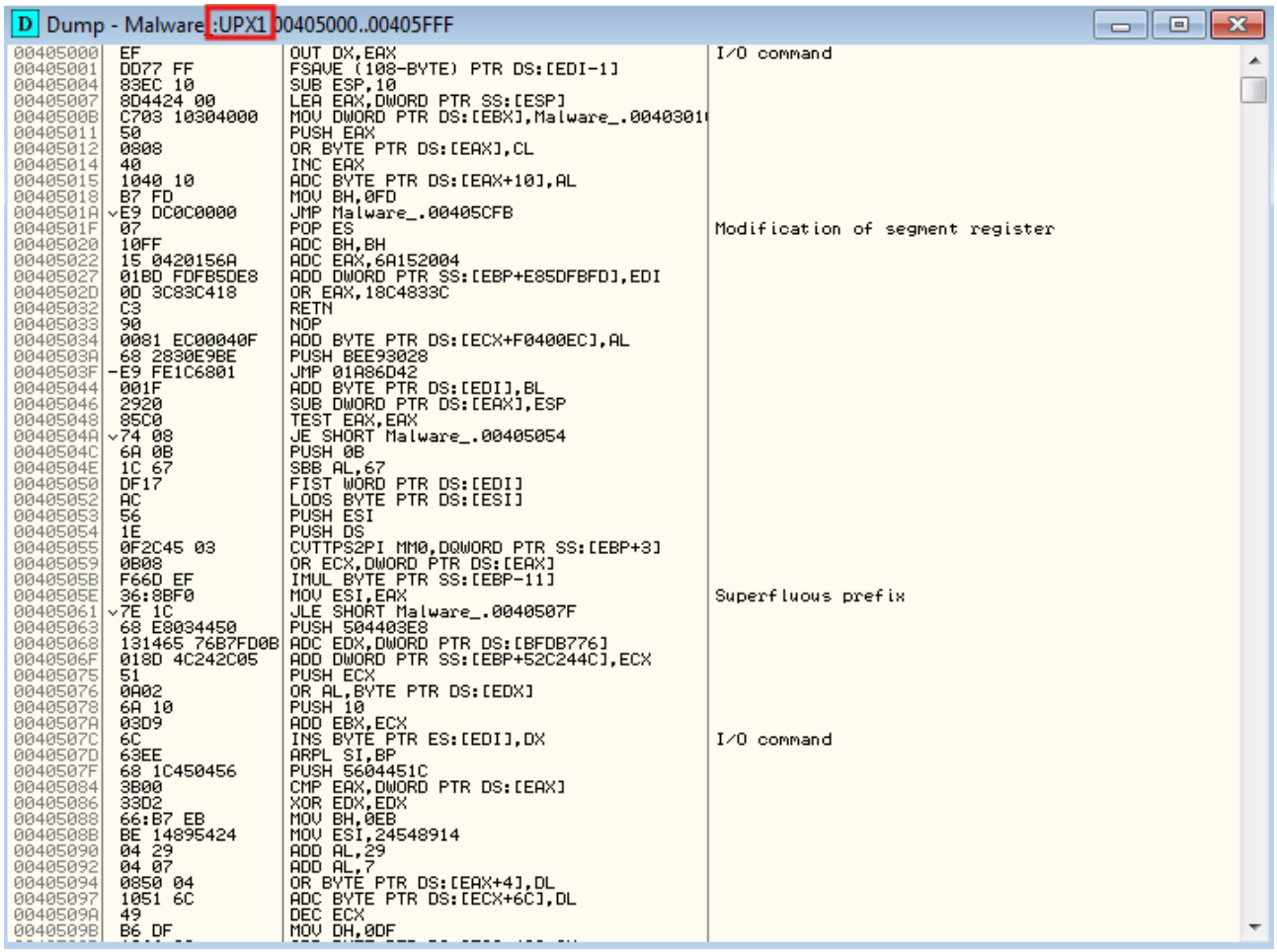
Per decriptare le sezioni nascoste si può usare un altro programma chiamato **OllyDbg**. Si seleziona l'eseguibile del malware e si lancia l'esecuzione. Dopo con **Alt+M** si accede al menu chiamato **Memory Map**. Al suo interno si notano le sezioni che interessano a noi (UPX0, UPX1, UPX2).



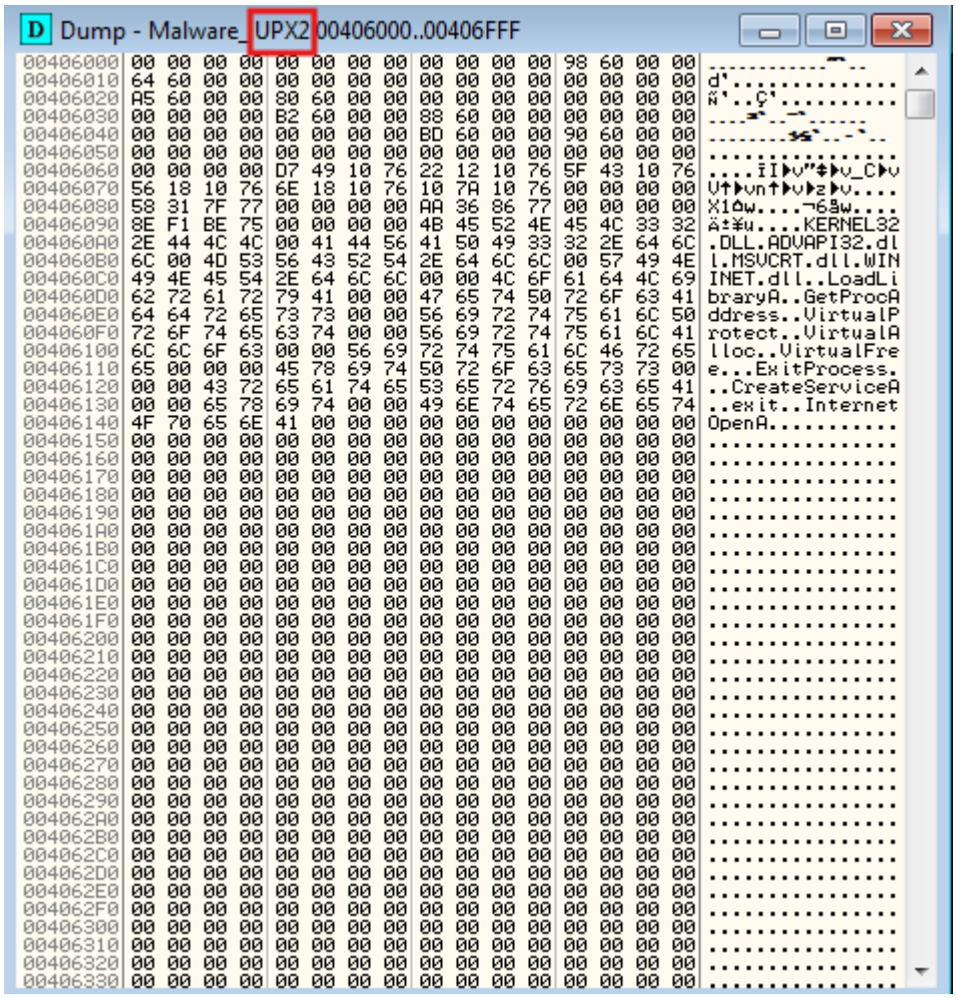
Si nota che tutte le sezioni hanno permessi di lettura, scrittura ed esecuzione. Facendo doppio click sulla sezione UPX0 si ottiene questa finestra.



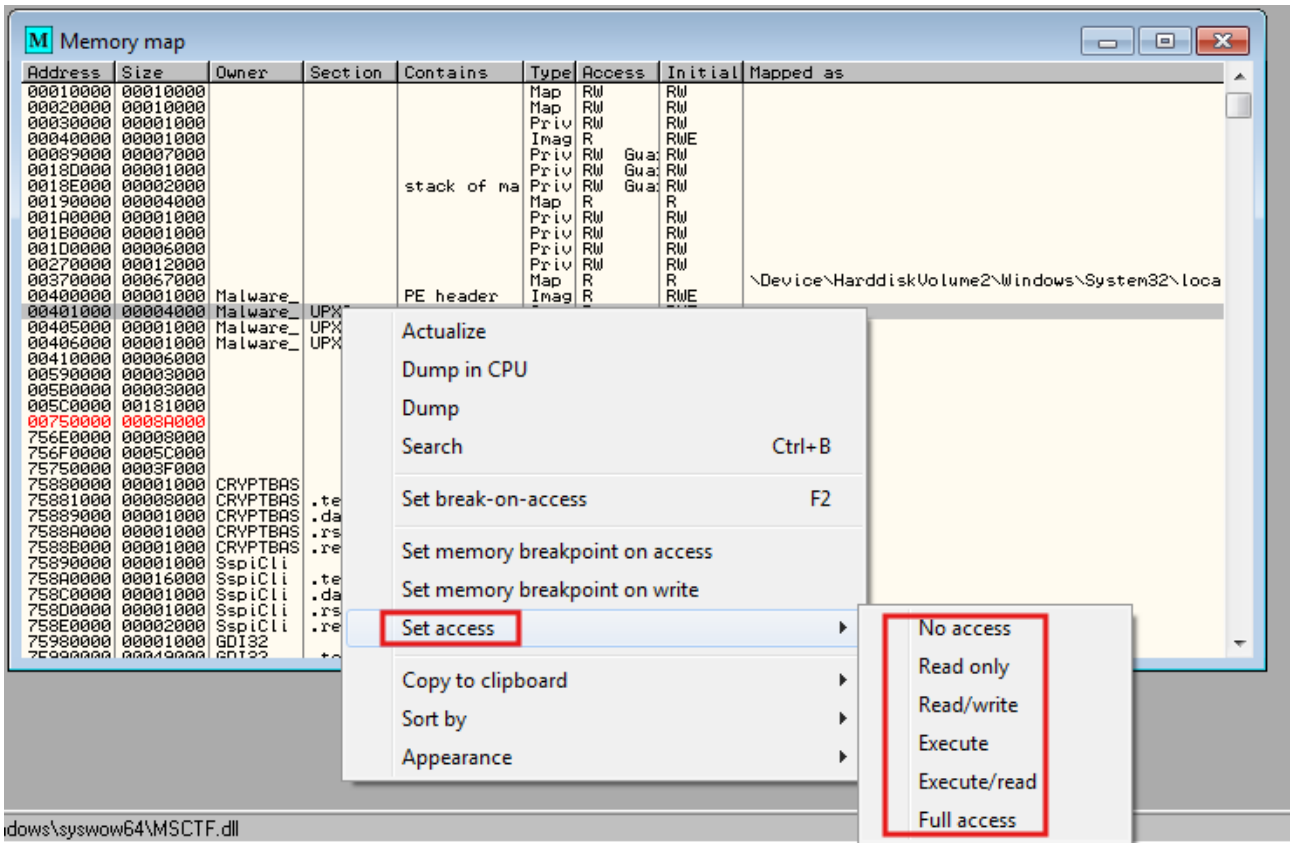
Facendo lo stesso sulla voce **UPX2** si ottiene invece questa finestra contenente delle istruzioni in assembly.



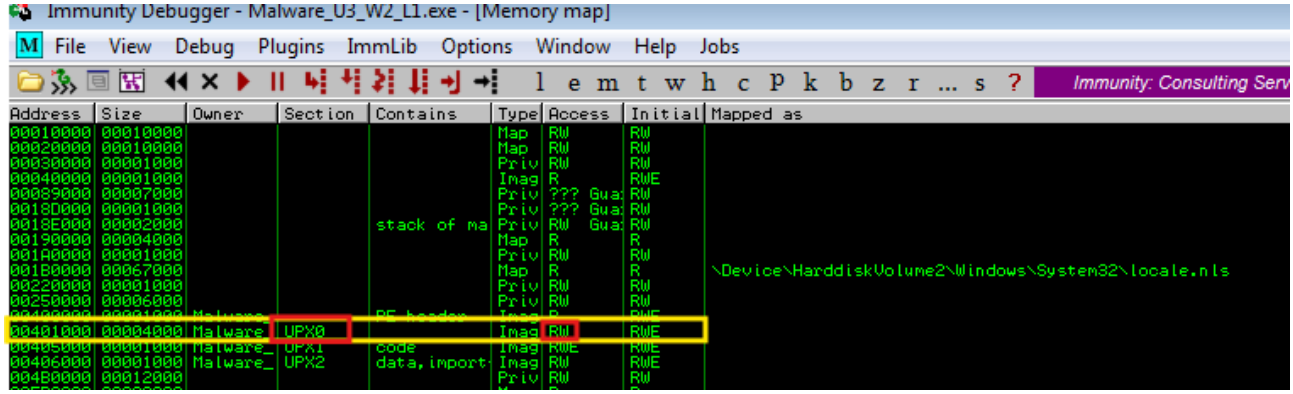
Qui ciò che si ottiene aprendo la voce **UPX2**.



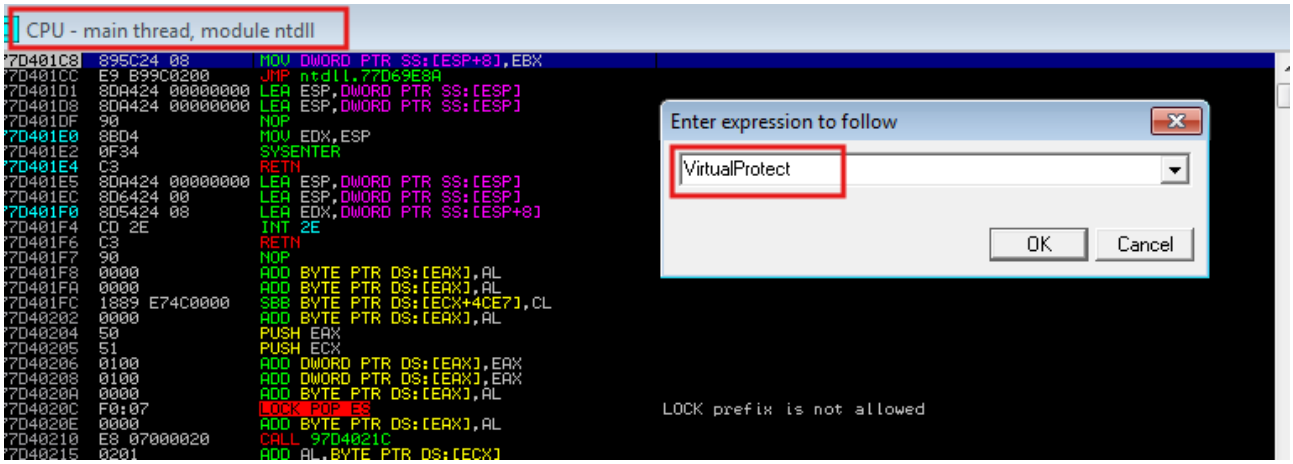
Si devono ora settare i permessi della sezione **UPX0** in sola lettura e scrittura.



Per Effettuare la modifica dei permessi ho usato un altro tool chiamato **Immunity Debugger**, dato che il precedente non riusciva ad applicare le modifiche. Quindi ho impostato i permessi di lettura e scrittura alla sezione **UPX0**.



Nella finestra “CPU – main thread”, con **Ctrl+G** si apre questa finestra. Digitando il comando **VirtualProtect**, si raggiunge la funzione che ha questo nome, che viene evidenziata nella finestra che mostra l’esecuzione del codice. Serve inserire un **breakpoint** proprio in corrispondenza di questa funzione usando il tasto **F2**.



Purtroppo dopo vari tentativi e ricerche non sono riuscito a far funzionare correttamente il programma per estrarre le sezioni del malware.

Dopo aver continuato l'esecuzione del programma con il tasto **F9**, si è raggiunto il **breakpoint** in corrispondenza della funzione **VirtualProtect**.