REPORT 09/01/2023

FACCILONGO DOMENICO

ANALISI STATICA BASICA

TASK

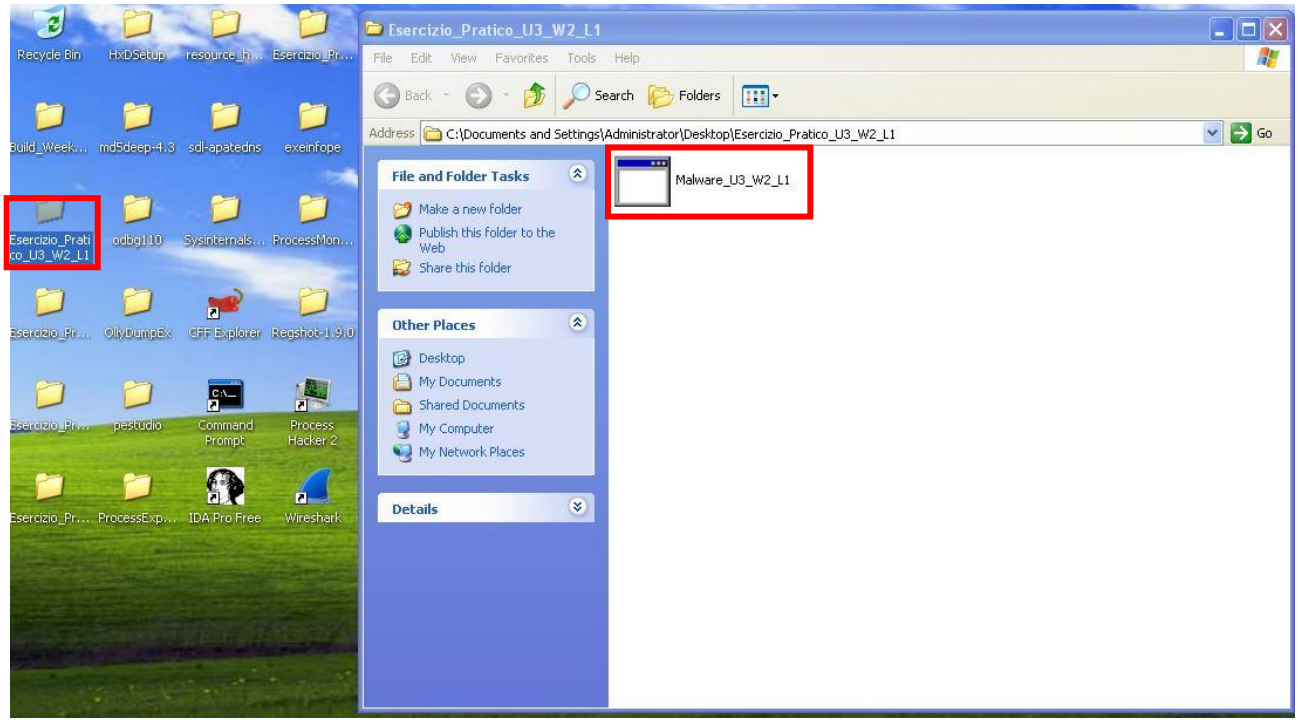
Nella lezione teorica del mattino, abbiamo visto come recuperare informazioni su un malware tramite l'analisi statica basica.

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

ANALISI E VALUTAZIONI

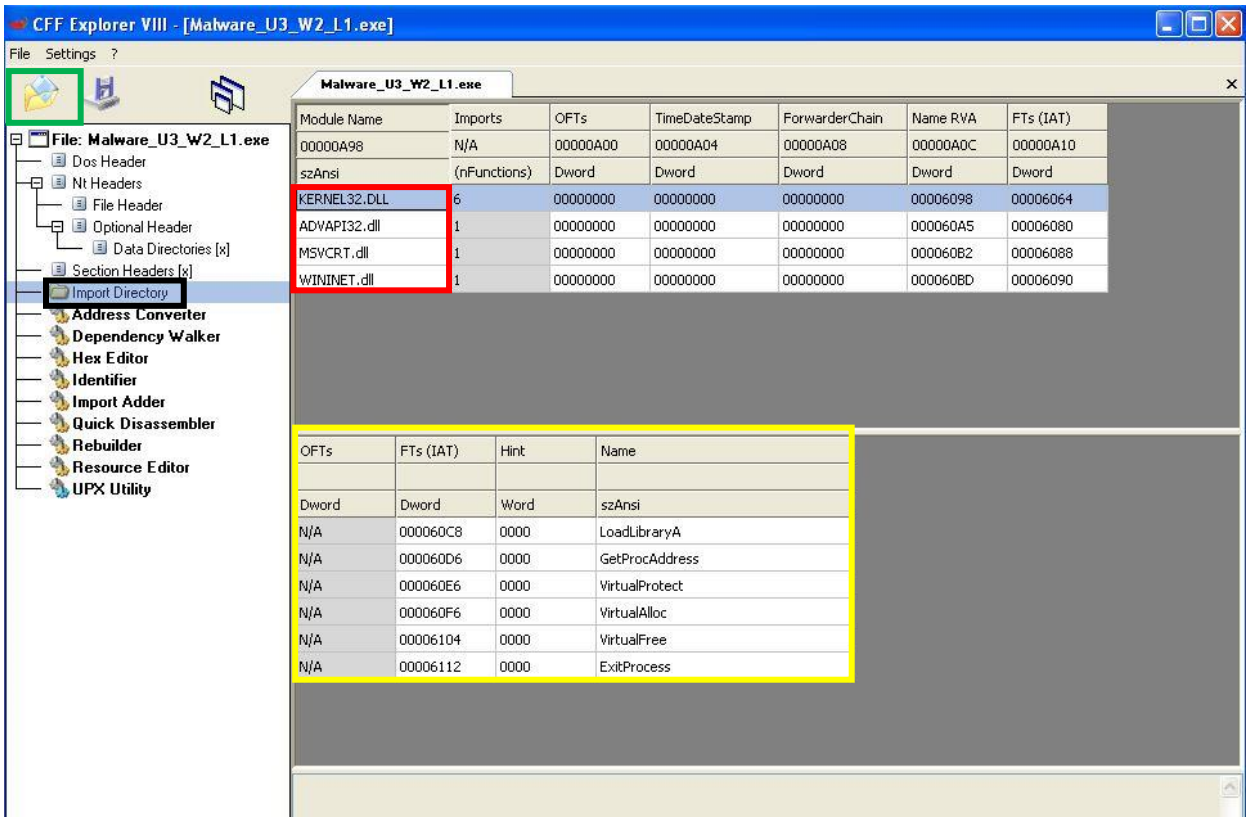
Come prima cosa andiamo a prendere il nostro file in esame nella cartella dal nome <<**Esercizio_Pratico_U3_W2_L1**>> che si troverà sul desktop:



Come passo successivo andiamo ad analizzare il file utilizzando i tool visti nella lezione teoria come: **CFF Explorer** e **Exeinfo PE.**

LIBRERIE IMPORTATE DAL MALWARE

Per andare a scovare le librerie importate dal malware utilizziamo il tool **CFF Explorer** che avremo da desktop. All'apertura, e caricando il file, andiamo nella sezione a sinistra denominata **Import Directory.** Da qui possiamo vedere le librerie che sono state importate:



LEGGENDA:

❑ Permette di caricare il file

❑ Sezione di analisi

❑ Permette di visualizzare a schermo le librerie importate

❑ Permette di visualizzare a schermo le funzioni della libreria selezionata

Andiamo ora ad analizzare le librerie ottenute, abbiamo:

- **KERNEL32.DLL** = Libreria che contiene le funzioni principali per interagire con il sistema oprerativo. Es. gestione della memoria, manipolazione del file.
- **ADVAPI32.dll** = Libreria che contiene le funzioni per interagire con i servizi e i registri del sistema operativo Microsoft.
- **MSVCRT.dll** = Libreria che contiene funzioni per la manipolazione di stringhe, allocazione di memoria e altro come chiamate input/output.
- **WININET.dll** = Libreria che contiene le funzioni per l'implementazione di alcuni protocolli di rete come HTTP, FTP, NTP.

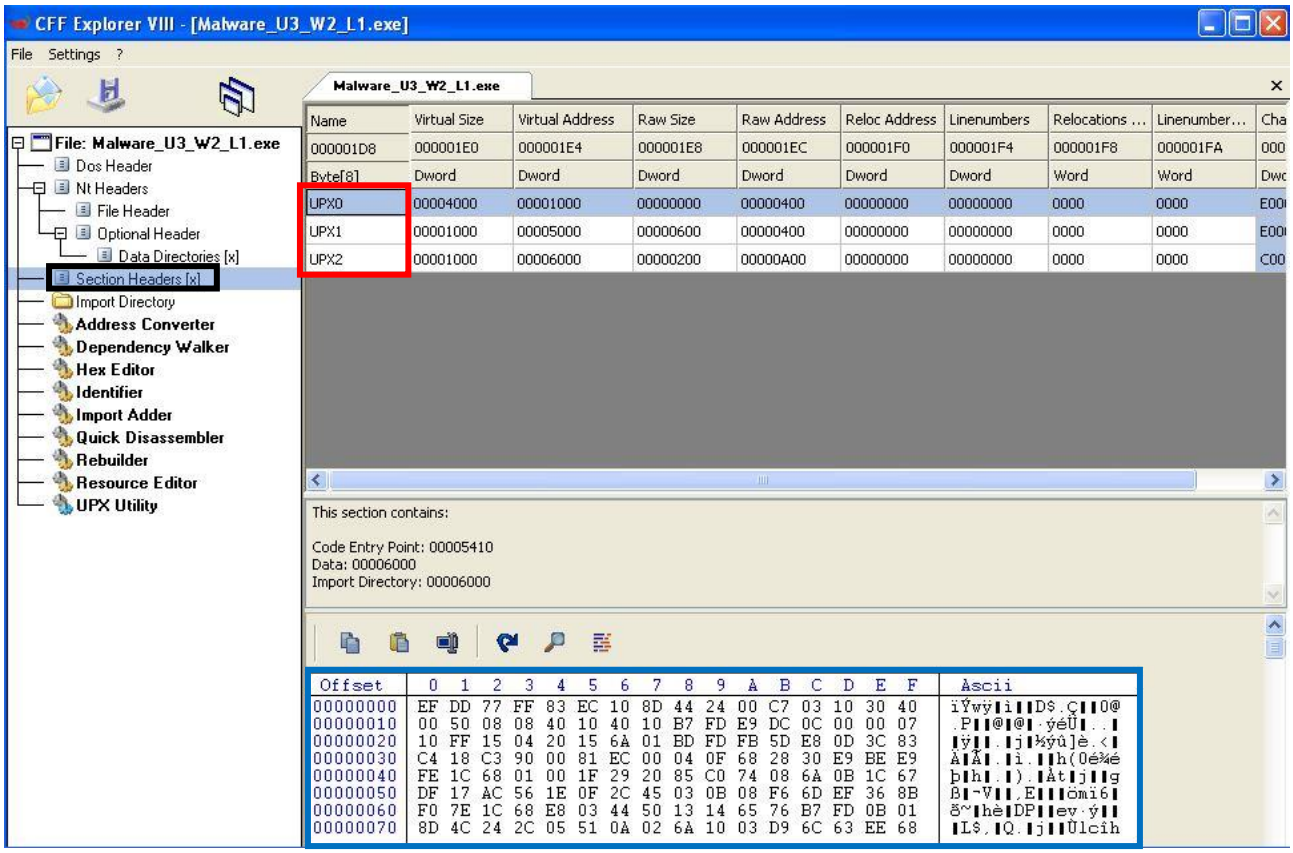
Mentre le funzioni ricavate sono in totale 9, di cui:

- 6 x Kernell32.dll
- 1 x Advapi32.dll
- 1 x Msvcrt.dll
- 1 x Wininet.dll

N/A	00006120	0000	CreateServiceA
N/A	00006130	0000	exit
N/A	00006136	0000	InternetOpenA

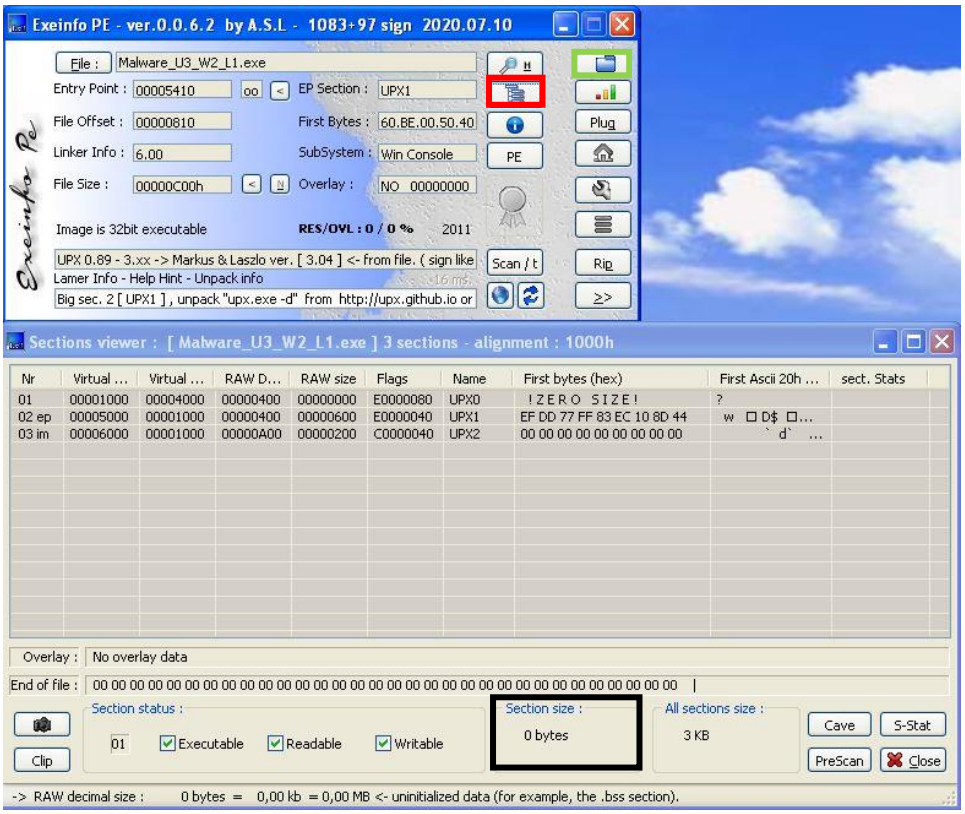
SEZIONI DI CUI SI COMPONE IL MALWARE

In **CFF** basterà andare nella sezione denominata **Section Headers** e tra i risultati ottenuti abbiamo:



Da cui si ottengono le seguenti sezioni, le quali sono compresse sotto forma **UPX** un software in grado di comprimere le sezioni:

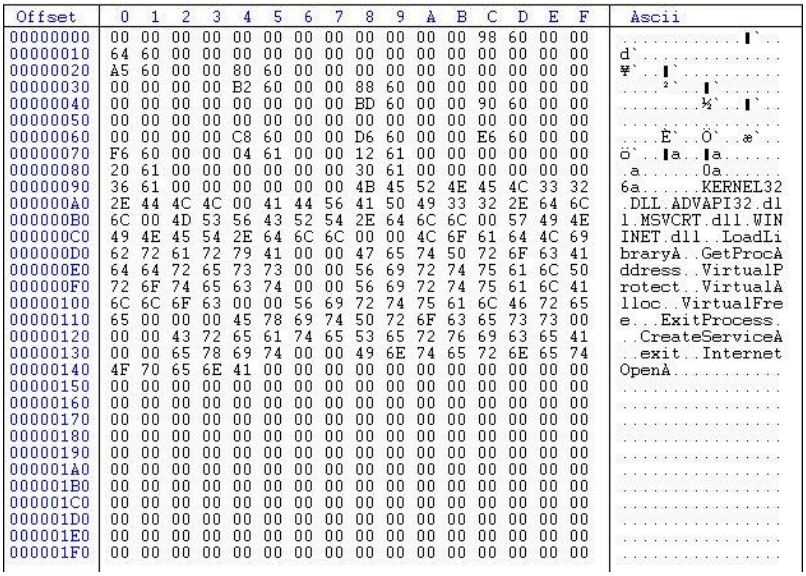
- **UPX0** = Unione di UPX1 e UPX2 di fatti se andassimo a prendere la descrizione nel rettangolo sotto (blu in figura soprastante) questo ci permette di intuire, dopo aver analizzato anche le altre due sezioni, che è l'unione di UPX1 e UPX2. E che pertanto possiamo definirlo come una sezione di tipo .text contenente le istruzioni e le righe di codice che la CPU eseguirà una volta che il software verrà avviato. Si può notare un'altra particolarità, che se andando ad analizzare la stessa sezione presa in esame, sull'altro tool (Exeinfo PE) la grandezza è di circa 0 bytes, idoneo per un file.txt



Per arrivare a questa sezione andiamo a caricare dapprima il file (riquadro verde), successivamente apriamo il file dall'icona (riquadro rosso) e infine notiamo questa particolarità della grandezza, dopo aver selezionato il nostro UPX0, dal riquadro nero in figura soprastante.

- **UPX1** = Possiamo paragonarlo ad **.data** ovvero che contiene al suo interno tipicamente i dati/variabili del programma eseguibile. Come si può vedere dall'ASCI della foto successiva:

- **UPX2** = Quest'ultimo invece può essere interpretato come un **.rdata** in quanto ci sono scritte sia le librerie importate che le loro funzioni. Come si evince dall'ASCI nell'immagine successiva:



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	00	00	00	00	00	00	00	00	00	00	00	00	98	60	00	00	..... ` ..
00000010	64	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	d` .....
00000020	A5	60	00	00	80	60	00	00	00	00	00	00	00	00	00	00	¥` .. ` .....
00000030	00	00	00	00	B2	60	00	00	88	60	00	00	00	00	00	00	.... ² ` .. ` .....
00000040	00	00	00	00	00	00	00	00	BD	60	00	00	90	60	00	00	..... ½ ` .. ` ..
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000060	00	00	00	00	C8	60	00	00	D6	60	00	00	E6	60	00	00	.... È ` .. Ö ` .. æ ` ..
00000070	F6	60	00	00	04	61	00	00	12	61	00	00	00	00	00	00	ö ` .. ▮a .. ▮a .....
00000080	20	61	00	00	00	00	00	00	30	61	00	00	00	00	00	00	.a ..... 0a .....
00000090	36	61	00	00	00	00	00	00	4B	45	52	4E	45	4C	33	32	6a KERNEL32
000000A0	2E	44	4C	4C	00	41	44	56	41	50	49	33	32	2E	64	6C	.DLL.ADVAPI32.dl
000000B0	6C	00	4D	53	56	43	52	54	2E	64	6C	6C	00	57	49	4E	l.MSVCRT.dll.WIN
000000C0	49	4E	45	54	2E	64	6C	6C	00	00	4C	6F	61	64	4C	69	INET.dll..LoadLi
000000D0	62	72	61	72	79	41	00	00	47	65	74	50	72	6F	63	41	braryA..GetProcA
000000E0	64	64	72	65	73	73	00	00	56	69	72	74	75	61	6C	50	ddress..VirtualP
000000F0	72	6F	74	65	63	74	00	00	56	69	72	74	75	61	6C	41	rotect..VirtualA
00000100	6C	6C	6F	63	00	00	56	69	72	74	75	61	6C	46	72	65	lloc..VirtualFre
00000110	65	00	00	00	45	78	69	74	50	72	6F	63	65	73	73	00	e...ExitProcess.
00000120	00	00	43	72	65	61	74	65	53	65	72	76	69	63	65	41	..CreateServiceA
00000130	00	00	65	78	69	74	00	00	49	6E	74	65	72	6E	65	74	..exit..Internet
00000140	4F	70	65	6E	41	00	00	00	00	00	00	00	00	00	00	00	OpenA .....
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

CONSIDERAZIONE FINALE SUL MALWARE

Possiamo dunque dire in conclusione che attraverso le informazioni ottenute da librerie, funzioni e sezioni, il malware analizzato in questione:

- Funzione Wininet.dll: all'interno della libreria si ha una funzione denominata **InternetOpenA** la quale si presuppone che il malware avvi la connessione con protocolli di tipo HTTP, FTP o NTP.
- Funzione Advapi.dll: al suo interno ritroviamo la funzione **CreateServiceA** e si presuppone cheil malware crei un oggetto sulla macchina vittima.
- Libreria Kernell.dll: al suo interno ritroviamo diverse funzioni che ipotizziamo permettano all'attaccante di interagire con il sistema operativo.

Il tutto però è una supposizione in quanto il malware è compresso con software UPX il chè non permette di visualizzare totalmente le funzioni delle librerie prese in esame.