Malware analysis

Sommario

Traccia S10/L1

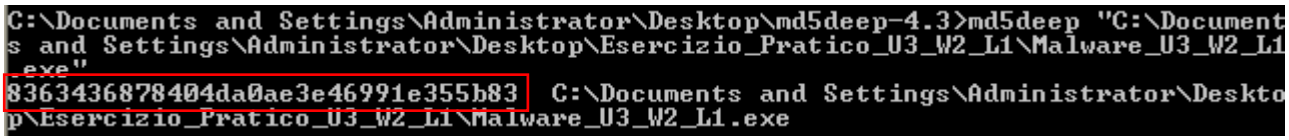
Con riferimento al file eseguibile contenuto nella cartella «Esercizio_Pratico_U3_W2_L1» presente sul desktop della vostra macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti:

- 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

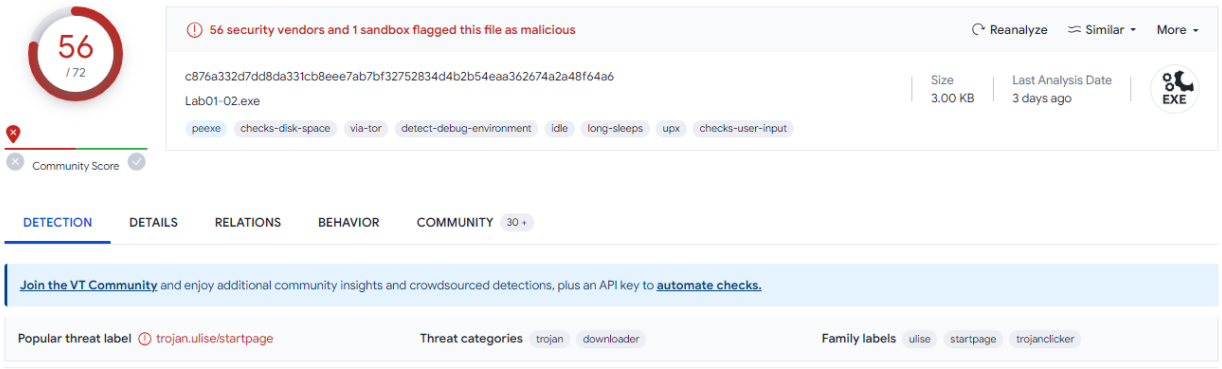
Verifica dell'eseguibile

Per verificare se l'eseguibile indicato nella traccia è effettivamente un malware calcoliamo l'hash tramite md5deep, una utility da riga di commando. L'hash è una stringa alfanumerica univoca che identifica un file, in pratica un'impronta digitale.

Avviamo il command prompt e cambiamo directory con quella contenente l'utility. Inseriamo il comando **md5deep "path del file da hashare".**

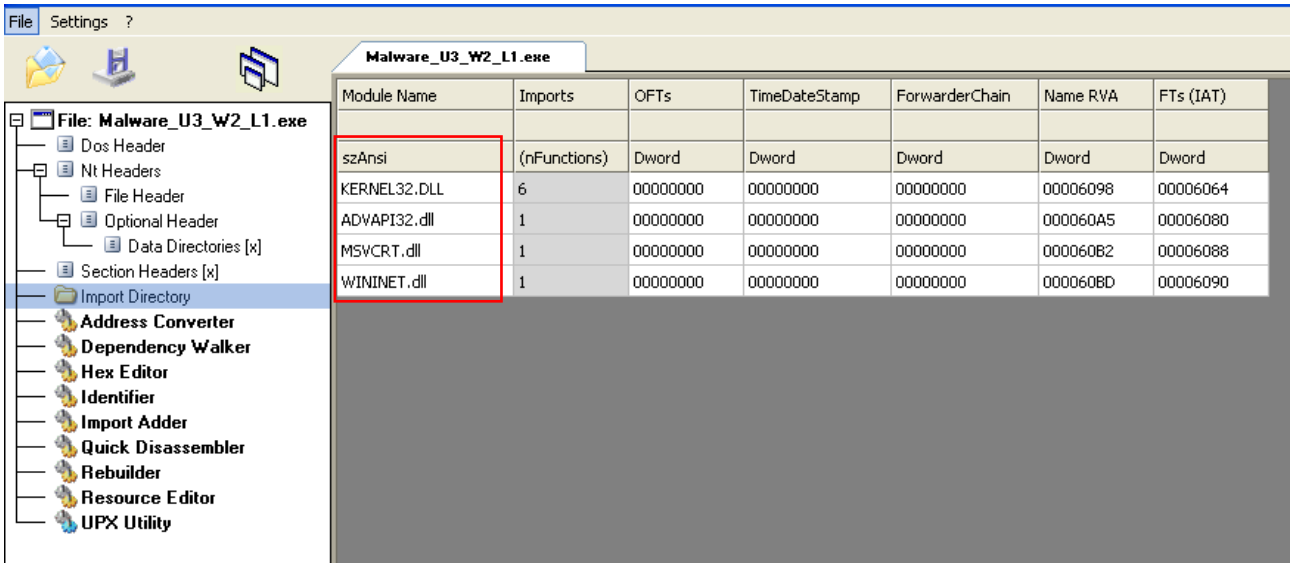


Ora che abbiamo ottenuto l'hash possiamo ricercalo su Virus Total, che mi indica che è un file malevolo.



Analisi librerie

Per analizzare le librerie utilizziamo CFF Explorer. Le librerie sono un insieme di funzioni presenti nel sistema operativo che possono essere richiamate da un software

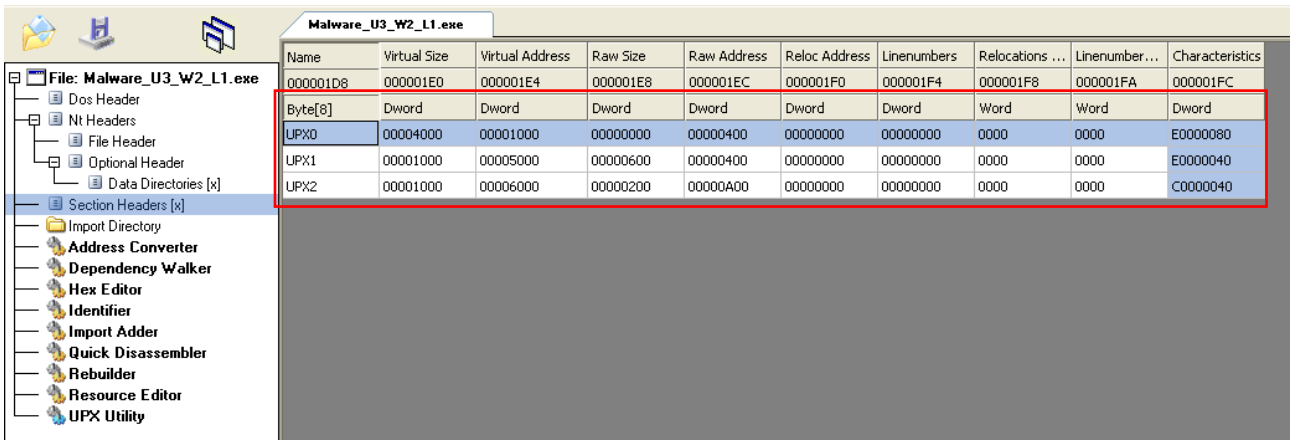


Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.DLL	6	00000000	00000000	00000000	00006098	00006064
ADVAPI32.dll	1	00000000	00000000	00000000	000060A5	00006080
MSVCRT.dll	1	00000000	00000000	00000000	000060B2	00006088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

- KERNEL32.DLL: contiene le funzioni principali per interagire con il sistema operativo
- ADVAPI32.DLL: contiene le funzioni per interagire con i servizi e registri del sistema operativo
- MSVCRT.DLL: contiene funzioni per la manipolazione stringhe, allocazione memoria, chiamate per input/output in stile linguaggio C.
- WININET.DLL: contiene le funzioni per l'implementazione di protocolli di rete come HTTP, FTP, NTP

Analisi sezioni

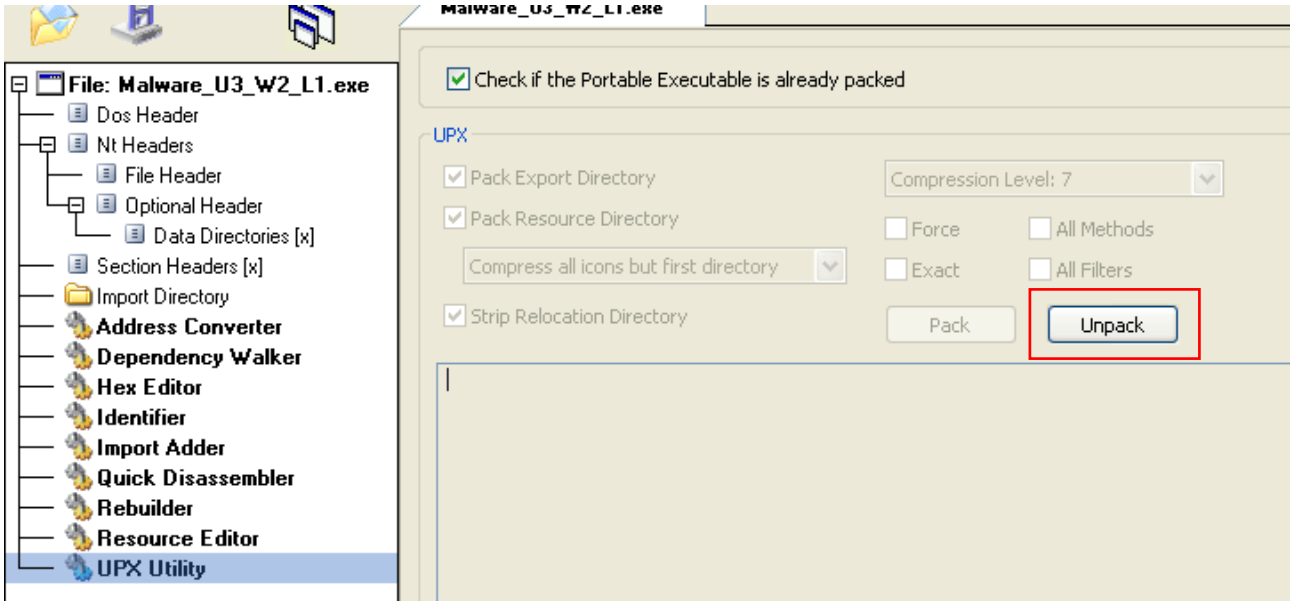
Per analizzare le sezioni dell'eseguibile spostiamoci nella sezione Section Headers.

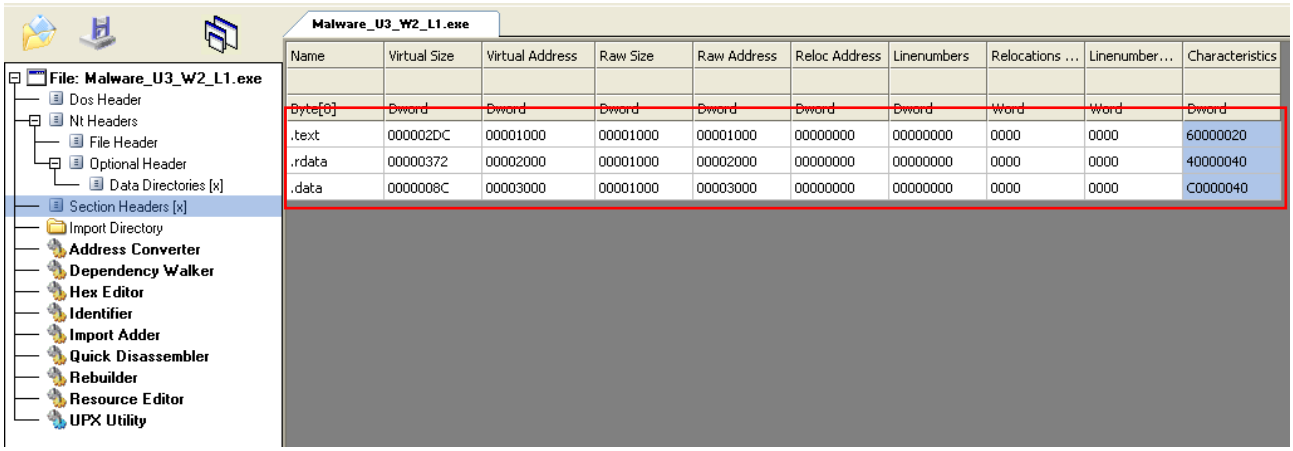


Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
000001D8	000001E0	000001E4	000001E8	000001EC	000001F0	000001F4	000001F8	000001FA	000001FC
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
UPX0	00004000	00001000	00000000	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

L'eseguibile è composto di 3 sezioni. Il nome di queste non è standard (di solito possiamo trovare .text, .data, .rsrc etc…), e questo è un indicatore di impacchettamento, sfruttato dal threat actor per nascondere il tipo di sezione. UPX sta per Ultimate Packer for eXecutables, un programma di compressione di eseguibili open source gratuito.

Con "UPX utility" è possibile decomprimerle cliccando su unpack.





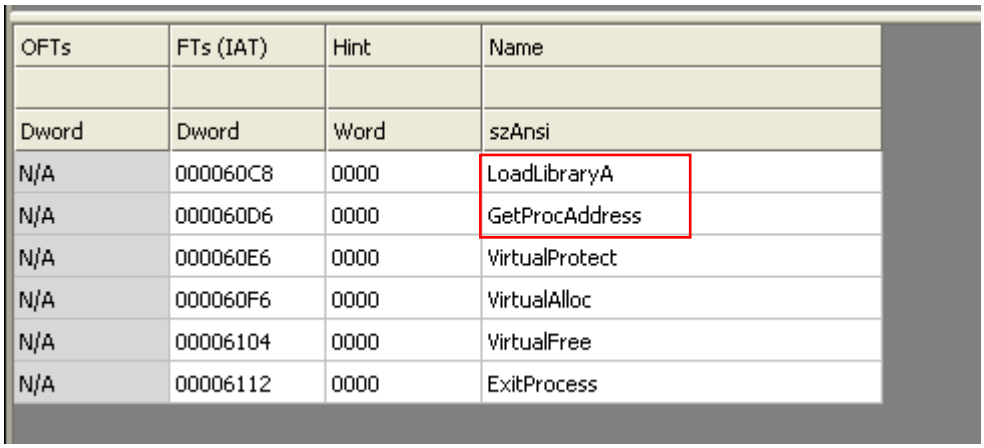
Possiamo notare le sezioni:

- .text: contiene le istruzioni che la CPU eseguirà una volta che il software sarà avviato
- .rdata: include le informazioni sulle librerie e le funzioni importate ed esportate dell'eseguibile
- .data: contiene le variabili globali del programma eseguibile.

Considerazioni finali

Virus Total ci indica che l'eseguibile che abbiamo analizzato è un Trojan, un tipo di malware che si nasconde all'interno di un programma apparentemente innocuo, ma che in realtà contiene istruzioni dannose per il computer.

Con l'analisi statica basica non è possibile stabilire il funzionamento di questo malware, dato che tra le funzioni importate dalla libreria KERNEL32.DLL sono presenti **LoadLibraryA** e **GetProcAddress**, che vengono utilizzate per caricare funzioni addizionali durante l'esecuzione.



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

Con un’analisi dinamica sarà possibile ottenere più informazioni.