ANALISI STATICA BASICA

S10/L1

ELEONORA VIOLA

TABLE OF CONTENT

INTRODUCTION

- Traccia:

Esercizio Analisi statica 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:

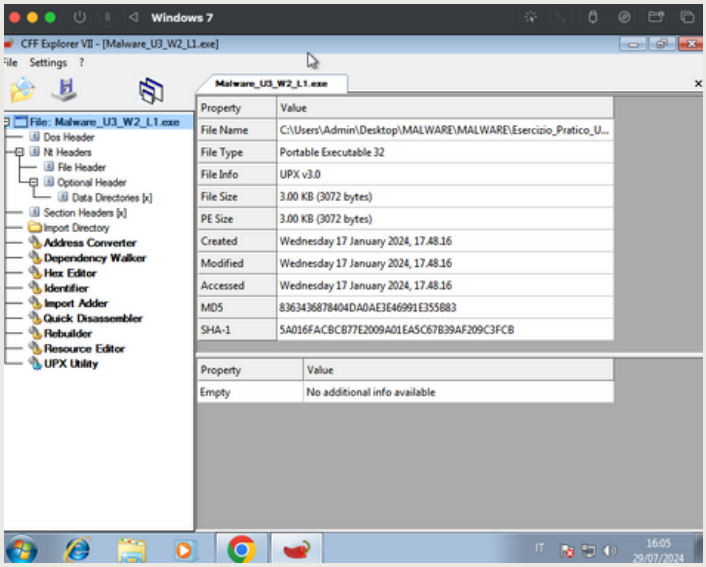
• Indicare le librerie importate dal malware, fornendo una descrizione per ognuna di esse
• Indicare le sezioni di cui si compone il malware, fornendo una descrizione per ognuna di essa
• Aggiungere una considerazione finale sul malware in analisi in base alle informazioni raccolte

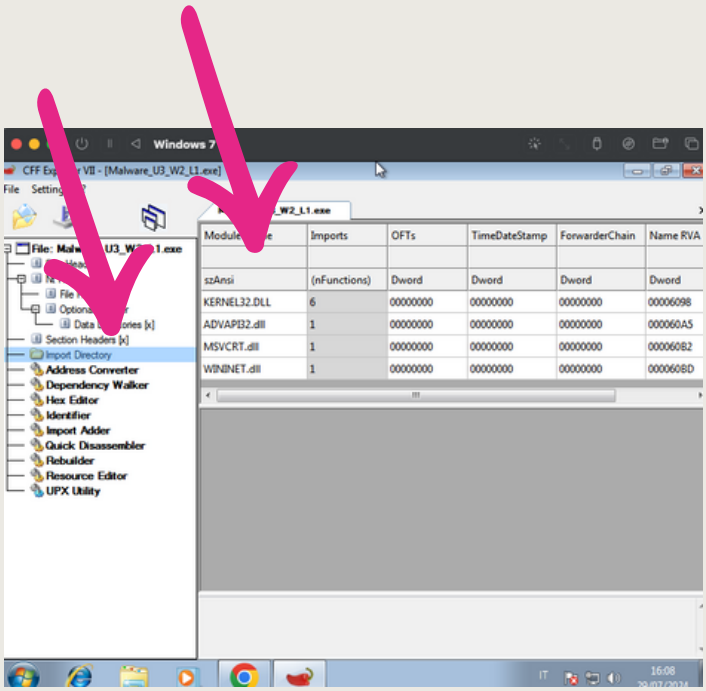
PROCEDURE

Per iniziare con il primo punto della pratica, clicchiamo sull'icona della cartella situata in alto a sinistra, per selezionare il file malware da analizzare.
Apriamo quindi **Malware_U3_W2_L1.exe**.

Nella lista delle voci a sinistra dell'immagine, ci spostiamo su "Import Directory" e nel riquadro di destra verranno mostrate le librerie utilizzate dal malware in questione, che sono:

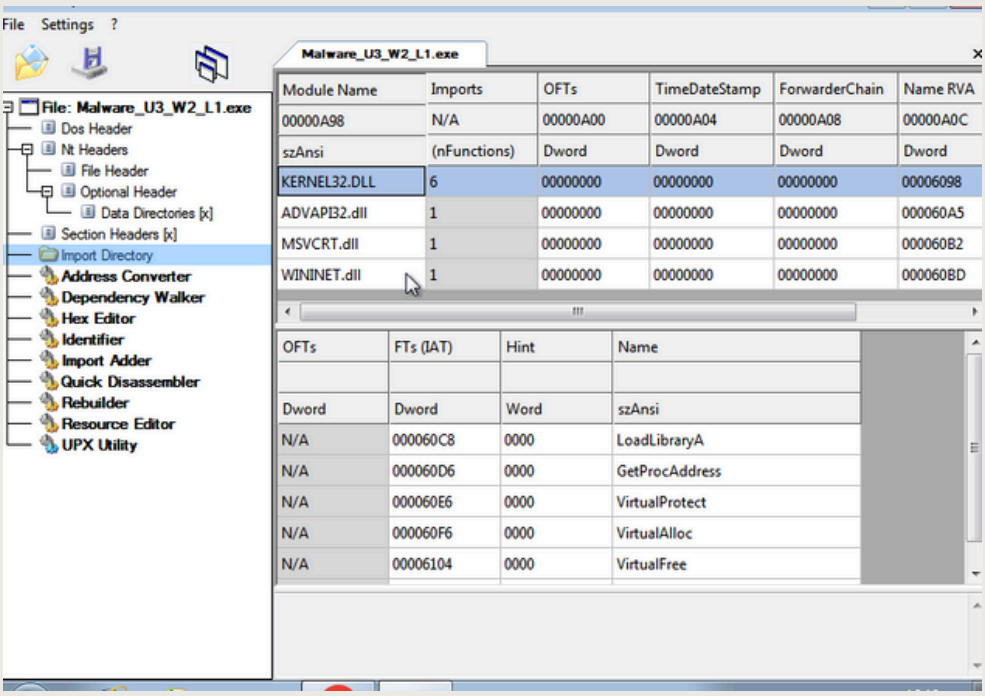
- **KERNEL32.DLL**
- **ADVAPI32.dll**
- **MSVCRT.dll**
- **WININET.dll**





PROCEDURE

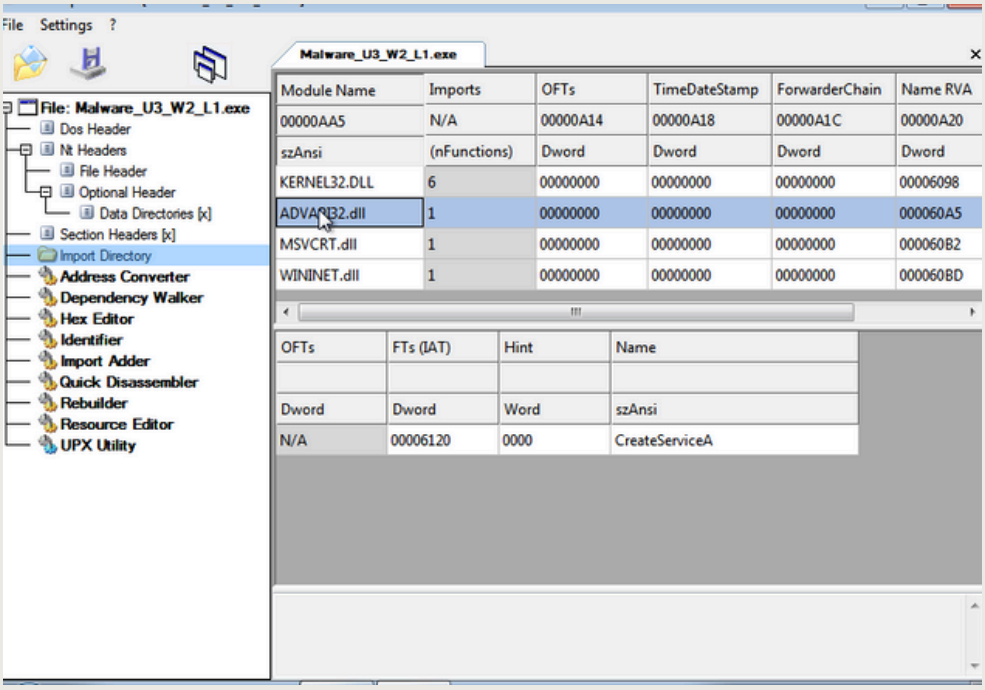
Procederemo ora ad analizzare ciascuna libreria una per una.
Iniziamo con **KERNEL32.DLL**.



KERNEL32.DLL è una libreria comunemente utilizzata poiché contiene funzioni fondamentali che consentono di interagire con il sistema operativo. Ad esempio, permette di manipolare file e gestire la memoria.

PROCEDURE

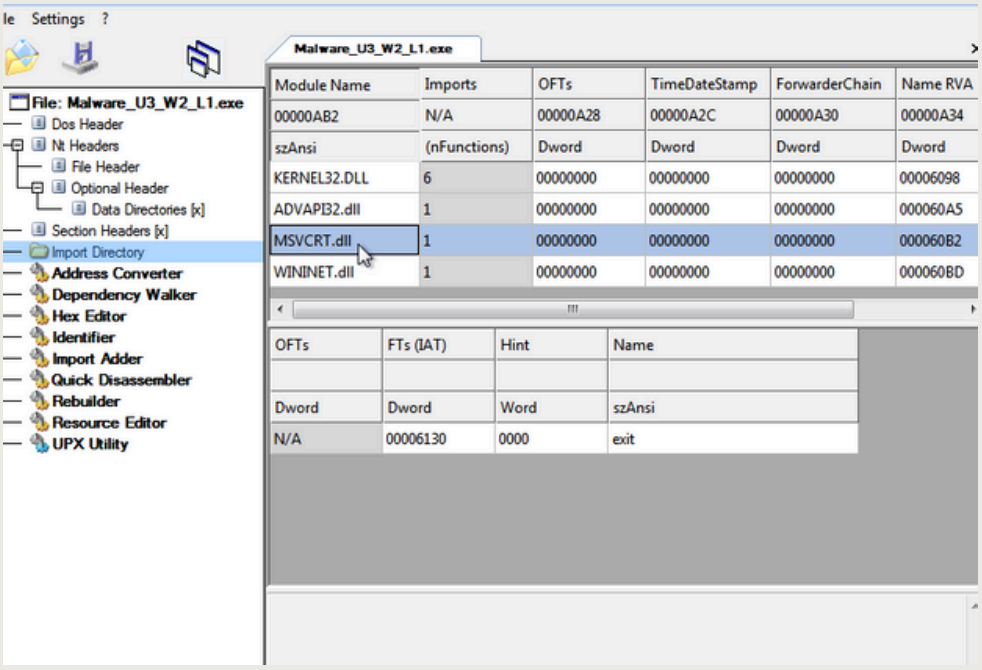
Passiamo ora a **ADVAPI32.dll.**



ADVAPI32.dll è una libreria che include funzioni per l'interazione con i servizi e i registri del sistema operativo Microsoft.
Il registro di sistema di Windows viene utilizzato per gestire e modificare le impostazioni relative alle preferenze dell'utente e alla configurazione del sistema. Potrebbe contenere file residui di programmi non più in uso.

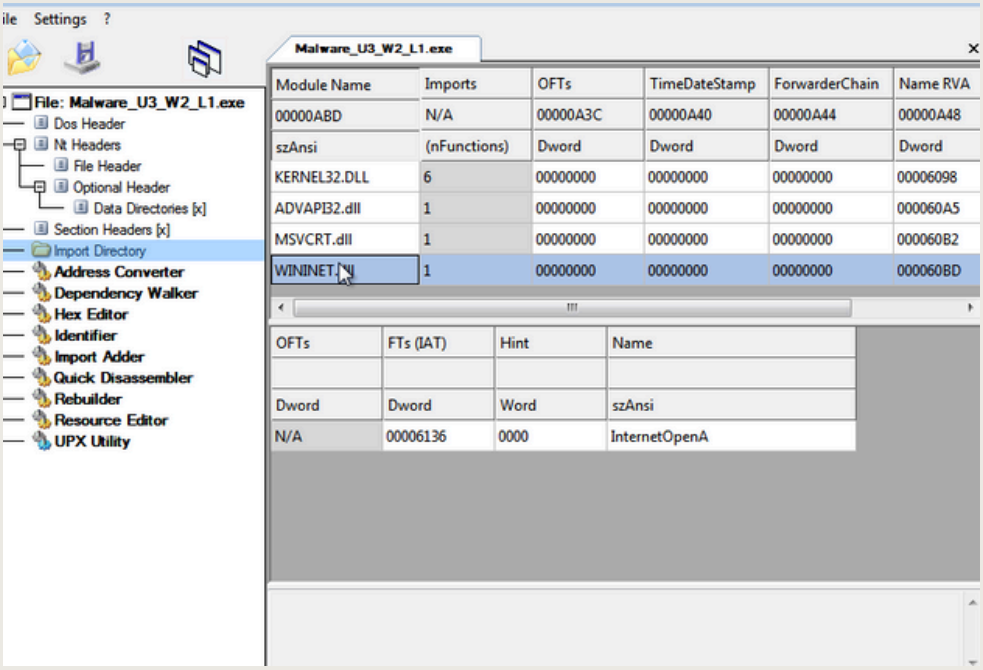
PROCEDURE

Successivamente, analizziamo **MSVCRT.dll**, una libreria che contiene funzioni per la manipolazione di stringhe, l'allocazione di memoria e chiamate per input/output in stile linguaggio C.



Concludiamo infine con l'ultima libreria, **WININET.dll**.

Questa libreria contiene funzioni che permettono l'implementazione di protocolli di rete come HTTP, FTP e NTP.
Essa consente alla DLL Internet di inizializzare le strutture di dati interne e prepararsi per le future chiamate dall'applicazione.



PROCEDURE

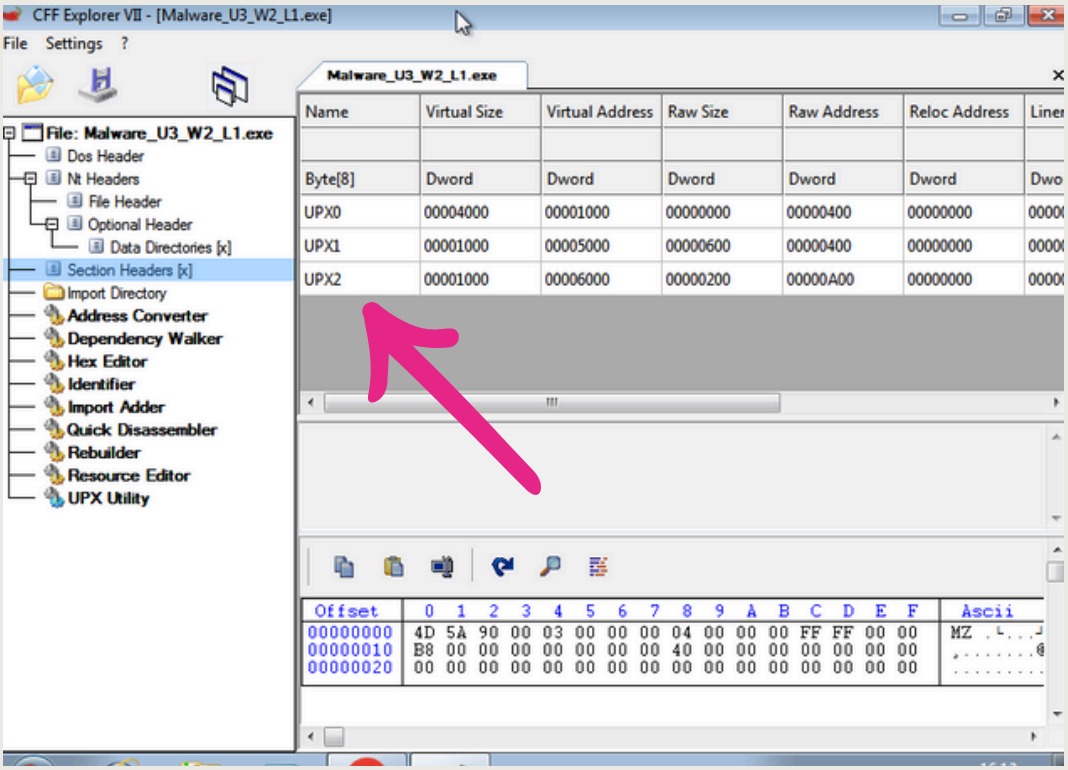
Passiamo ora al secondo punto della traccia, dove analizzeremo le sezioni che compongono il malware.
Dalle scansioni effettuate, troviamo le seguenti sezioni:

- UPX0
- UPX1
- UPX2

Analizzando queste sezioni e facendo una piccola ricerca, scopriamo che UPX (Ultimate Packer for eXecutables) è uno strumento di compressione e decompressione per eseguibili, progettato per ridurre le dimensioni dei file eseguibili.

UPX può essere utilizzato legalmente per comprimere e decomprimere file, ma può anche essere sfruttato dai malware per nascondere il proprio codice o rendere più difficile la loro rilevazione da parte dei software di sicurezza.

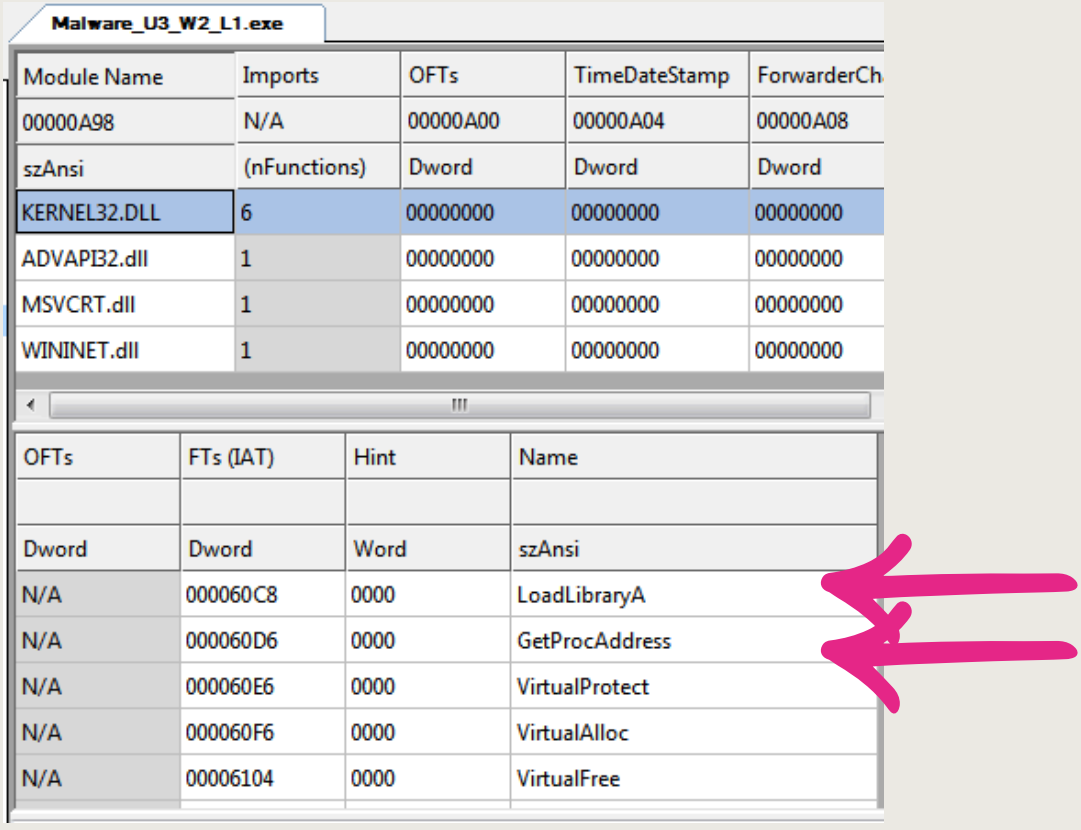
Le "sezioni" di un file eseguibile si riferiscono alle diverse parti che compongono il file, come la sezione del codice e la sezione dei dati. Alcuni malware potrebbero utilizzare tecniche come la compressione UPX per rendere più complessa l'analisi e la rilevazione.
Da ciò possiamo dedurre che il malware ha compresso o decompresso dei file per ridurne il volume. Notiamo anche che questo processo è stato eseguito tre volte, con le sezioni denominate UPX0, UPX1 e UPX2.



CONCLUSION

Concludendo si può affermare che questo malware rende difficile ottenere molte informazioni sul suo funzionamento tramite un'analisi statica di base.

Questo è confermato dalla presenza delle funzioni **LoadLibrary** e **GetProcAddress**, che suggeriscono che il malware carica le librerie durante l'esecuzione - runtime, nascondendo quindi le informazioni sulle librerie importate in fase iniziale.



Malware_U3_W2_L1.exe

Module Name	Imports	OFTs	TimeDateStamp	ForwarderCh
00000A98	N/A	00000A00	00000A04	00000A08
szAnsi	(nFunctions)	Dword	Dword	Dword
KERNEL32.DLL	6	00000000	00000000	00000000
ADVAPI32.dll	1	00000000	00000000	00000000
MSVCRT.dll	1	00000000	00000000	00000000
WININET.dll	1	00000000	00000000	00000000

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