

Analisi basica statica

Traccia:

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



Malanalysis



Malwarepr...

I file una volta importati sulla macchina virtuale li dobbiamo unzippare.
Una volta fatto apriamo **Malwareprogram**



Explorer Suite



ida pro



Process Hacker 2

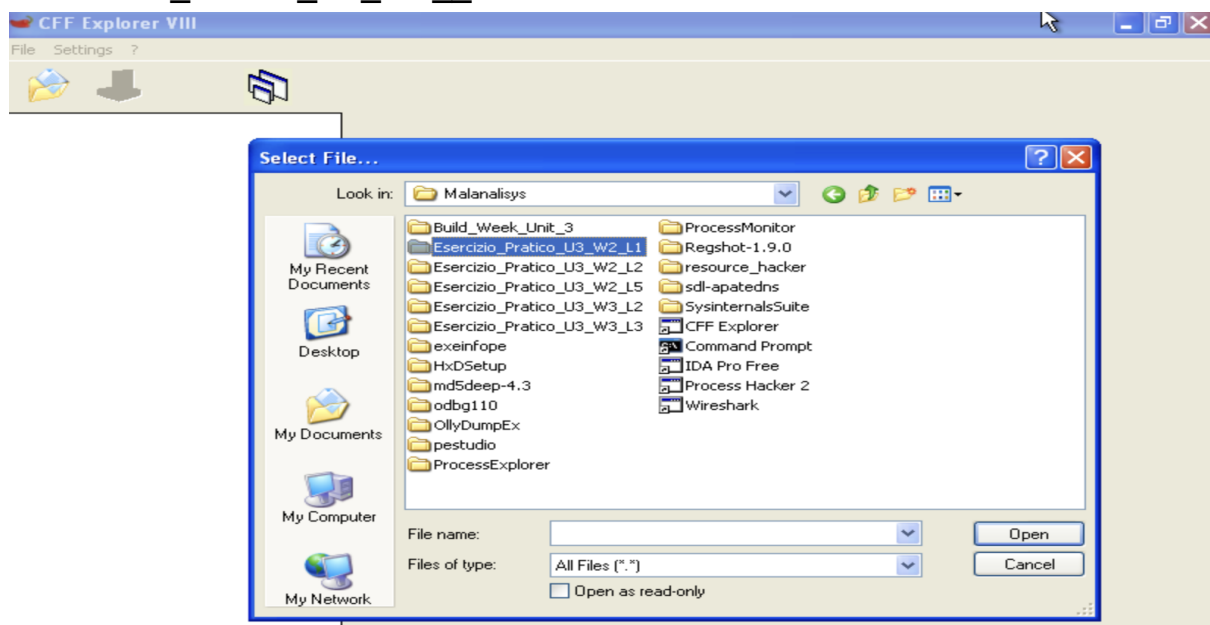


Wireshark

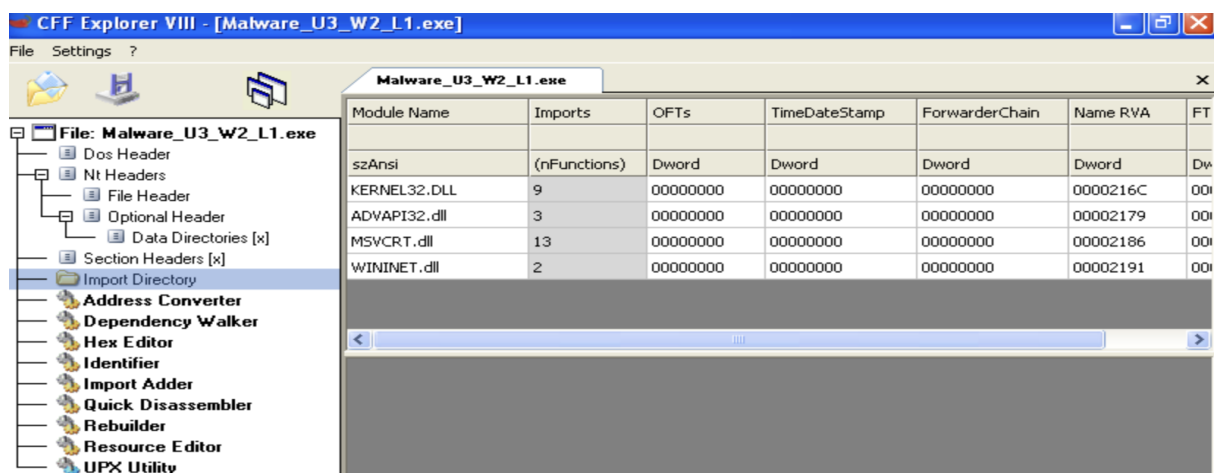
Selezioniamo **Explorer Suite** ed avviamo **CFF Explorer** con run



Successivamente andiamo ad aprire nella cartella Malanalysis
L'Esercizio_Pratico_U3_W2__L1



Come richiesto utilizziamo il tool “**CFF Explorer**”, procediamo con lo studio delle librerie importate dal Malware tramite la sezione “**import Directory**” del tool



Possiamo notare che il programma va ad importare quattro librerie per l'esecuzione

KERNEL32.DLL è una libreria di sistema di Windows che serve per interagire con il sistema operativo. Gestisce processi, thread, memoria, file, tempo e risorse.

Essenziale per molti programmi consentendo di eseguire operazioni tipo creare processi, leggere/scrivere file, allocare memoria e gestire il tempo di sistema

ADVAPI32.DLL è una libreria di Windows che fornisce funzionalità di crittografia, gestione dei certificati, autenticazione, controllo degli accessi e altre operazioni di sicurezza

MSVCRT.DLL è una libreria di runtime di Microsoft Visual C++ che contiene le funzioni di runtime standard utilizzate dai programmi C e C++ compilati con il compilatore Microsoft Visual C++ essenziale per il corretto funzionamento dei programmi compilati

WININET.DLL è una libreria Windows che fornisce funzionalità per la comunicazione via internet

Per effettuare richieste HTTP, inviare e ricevere dati tramite protocolli HTTP, FTP, gestire cookie, cache, proxy e altre operazioni di rete

Nella libreria **KERNEL32** ritroviamo

SystemTimeToFileTime
GetModuleFileNameA
CreateWaitableTimerA
ExitProcess
OpenMutexA
SetWaitableTimer
WaitForSingleObject
CreateMutexA
CreateThread

Queste funzioni forniscono meccanismi di sincronizzazione, gestione del tempo e controllo dei thread, essenziali per lo sviluppo di applicazioni su piattaforma Windows

Nella libreria **ADVAPI32** ritroviamo

CreateServiceA
StartServiceCtrlDispatcherA
OpenSCManagerA

Questa libreria è in grado di interagire con i servizi Windows e gestirli in modo programmato fornendo le funzioni per la creazione, gestione e controllo dei servizi

Nella libreria **MSVCRT** ritroviamo

```
_exit  
_XcptFilter  
exit  
__p__initenv  
__getmainargs  
_initterm  
__setusermatherr  
_adjust_fdiv  
__p__commode  
__p__fmode  
__set_app_type  
_except_handler3  
_controlfp
```

Queste funzioni consentono di gestire l'esecuzione del programma come l'uscita, la gestione delle eccezioni, la gestione degli argomenti della riga di comando e altre operazioni nella programmazione in C++

Nella libreria **WININET** ritroviamo

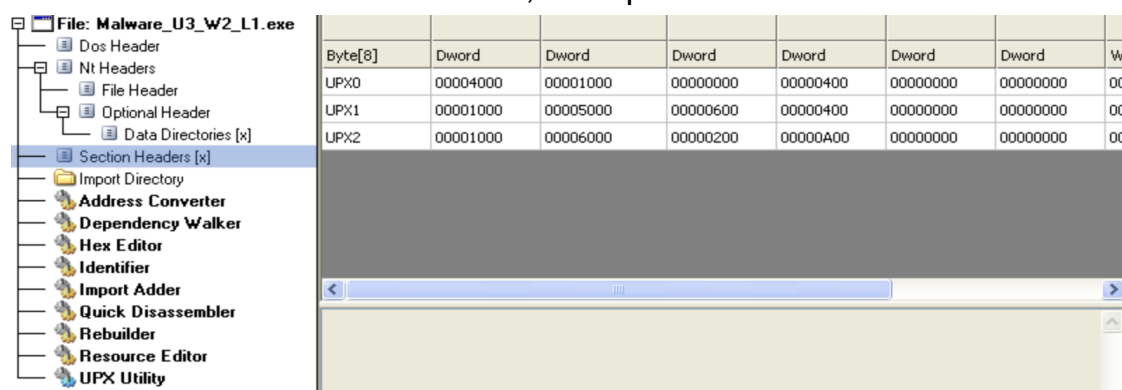
InternetOpenUrlA

InternetOpenA

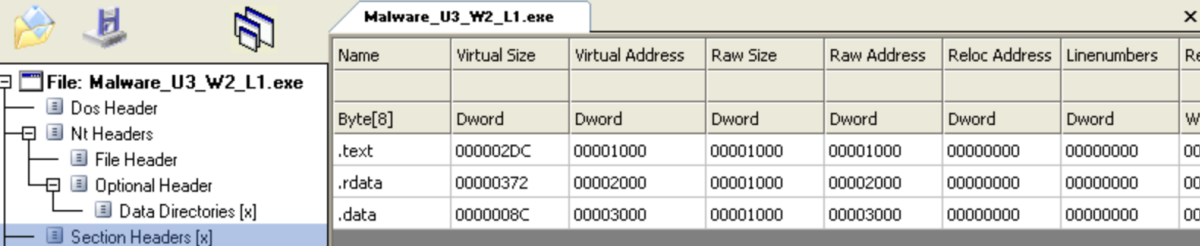
Entrambe le funzioni fanno parte dell' API di WinINET che fornisce la comunicazione con i internet e l'accesso alle risorse WEB

Spesso utilizzate per navigazione WEB, download di file o comunicazione con server remoti all'interno di applicazioni Windows

Le sezioni sono nascoste da UPX, UPX può ridurre le dimensioni del file



Tramite **UPX utility** faccio l' unpacking, il ripristino delle sezioni originali



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Re
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	W
.text	000002DC	00001000	00001000	00001000	00000000	00000000	00
.rdata	00000372	00002000	00001000	00002000	00000000	00000000	00
.data	0000008C	00003000	00001000	00003000	00000000	00000000	00

.text Questa sezione contiene le istruzioni, ovvero le righe di codice che la CPU eseguirà quando il software viene avviato, è la sezione principale di un file eseguibile (contenente il codice)

.rdata Questa sezione contiene informazioni sulle librerie e le funzioni importate ed esportate dall' eseguibile

.data Questa sezione contiene dati e variabili globali del programma eseguibile, le variabili seguenti sono accessibili da qualsiasi parte del programma essendo globali

Conclusioni

Possiamo ipotizzare che il malware vada a sfruttare connessioni internet di tipo HTTP/FTP/NTP tramite alla libreria **Wininet.dll**

potrebbe quindi comunicare con server remoti, scaricare o caricare file o eseguire attività di rete con questi protocolli

Tramite **Kernel32.dll** può accedere a risorse del sistema

Mentre con le librerie **LoadLibrary** e **GetProcAddress** indica che il malware carica dinamicamente alcune funzioni durante l' esecuzione

in questo modo può risultare meno invasivo e visibile con la presenza di una eventuale backdoor

Per fare un analisi più accurata sarebbe necessaria un'analisi dinamica

Ciandri I

