CONSEGNA S10L1

L'obbiettivo di oggi è quello di analizzare un file eseguibile malevolo sulla macchina Windows 7 al fine di:

- 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

Prima di cominciare, diamo qualche definizione:

FILE ESEGUIBILE: Windows utilizza per la maggiore parte dei file eseguibili il formato PE, Portable Executable, che al suo interno le informazioni necessarie al sistema operativo per capire come gestire mil codice del file, come ad esempio le librerie. Moduli o librerie contengono un insieme di funzioni quindi, quando il programma ha bisogno di una funzione, "chiama" una libreria al cui interno è definita la funzione necessaria

MODULO/LIBRERIA: In un sistema informatico, un modulo o una libreria è un insieme di codice predefinito e precompilato che fornisce una serie di funzionalità specifiche, come ad esempio operazioni matematiche, manipolazione di dati, interfaccia utente, comunicazione di rete, e così via. Questi moduli sono progettati per essere riutilizzabili e facilitano lo sviluppo del software, consentendo agli sviluppatori di utilizzare le funzionalità fornite senza dover scrivere il codice da zero. Le librerie possono essere sviluppate internamente o esternamente, e vengono comunemente distribuite sotto forma di file o pacchetti che possono essere integrati nei progetti software.

HEADER PE: L'header del formato Portable Executable (PE) è una parte essenziale di un file eseguibile o di un file DLL (Dynamic Link Library) su piattaforme Windows. Questo header fornisce informazioni cruciali per il caricamento e l'esecuzione del programma da parte del sistema operativo Windows. Tra le informazioni contenute nell'header PE ci sono le Signature, l'header DOS, l'header COFF e la tabella delle sezioni.

Controllare quali sono le librerie e funzioni richieste dall'eseguibile contenute nell'header PE è fondamentale per capire lo scopo del malware. Oltre alle funzioni importate, il file eseguibile può mettere a disposizione di altri programmi o dell'utente delle funzioni da "chiamare": l'header del formato PE contiene un elenco delle funzioni esportate da un eseguibile. Per controllare le funzioni importate ed esportate dal malware, nello specifico del file eseguibile **Malware_U3_W2_L1.exe**; ci serviremo di un tool di Windows: CFF Explorer.
In Fig.1 possiamo vedere la schermata del tool che si apre dopo che abbiamo caricato l'eseguibile che vogliamo caricare.

Malware_U3_W2_L1.exe

Property	Value
File Name	C:\Users\user\Desktop\MALWARE\Esercizio_Pratico_U3_W2_L1\Malw...
File Type	Portable Executable 32
File Info	UPX v3.0
File Size	3.00 KB (3072 bytes)
PE Size	3.00 KB (3072 bytes)
Created	Wednesday 19 January 2011, 11.10.42
Modified	Wednesday 17 January 2024, 17.48.15
Accessed	Wednesday 19 January 2011, 11.10.42
MD5	8363436878404DA0AE3E46991E355B83
SHA-1	5A016FACBCB77E2009A01EA5C67B39AF209C3FCB

Fig.1

Tramite il codice hash in formato MD5, che identifica univocamente il file, possiamo, tramite programmi come Virus Total, andare a informarci circa lo storico di questo malware, vedi Fig.2

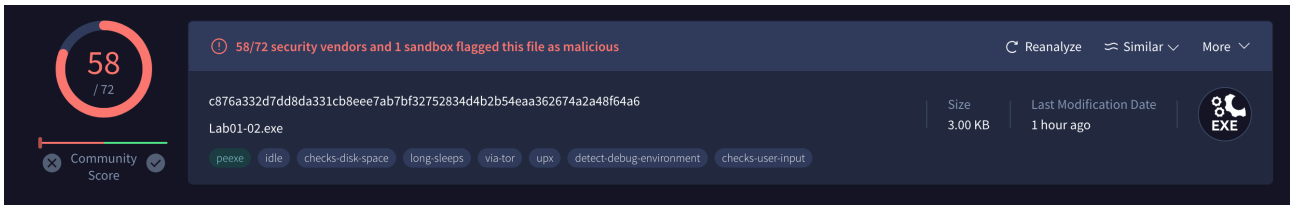


Fig.2

Per informarci sulle librerie importate dal malware, andiamo nella sezione apposita di CFF Explorer. Possiamo vedere che sono state importate 4 librerie, vedi Fig.3

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

Fig.3

Andiamo ad analizzare ogni libreria:

KERNEL32.DLL: Questa libreria contiene le funzioni principali per interagire con il sistema operativo, le funzioni chiamate sono le seguenti (Fig.4):

Name
szAnsi
LoadLibraryA
GetProcAddress
VirtualProtect
VirtualAlloc
VirtualFree
ExitProcess

Fig.4

- **LoadLibraryA**: Questa funzione viene utilizzata per caricare una libreria dinamica (DLL) nel processo chiamante. Prende in input il percorso del file DLL da caricare e restituisce un handle che identifica la libreria caricata. Il suffisso "A" nella funzione indica che si tratta della versione ANSI della funzione.
- **GetProcAddress:** Utilizzata per ottenere un puntatore a una funzione esportata da una libreria dinamica già caricata. Prende in input l'handle della libreria e il nome della funzione e restituisce il puntatore alla funzione. **Questo è utile quando si desidera chiamare una funzione all'interno di una libreria dinamica senza collegarla staticamente durante la compilazione.**
- **VirtualProtect**: Questa funzione consente di modificare i diritti di accesso per una o più pagine di memoria virtuali nel processo chiamante. È utile per cambiare la protezione della memoria, ad esempio rendendola eseguibile, scrivibile o solo leggibile.
- **VirtualAlloc**: Utilizzata per allocare memoria virtuale nel processo chiamante. Prende in input la dimensione della memoria richiesta e alcune opzioni di allocazione, come la localizzazione e la protezione della memoria, e restituisce un puntatore alla memoria allocata.
- **VirtualFree**: Questa funzione viene utilizzata per liberare la memoria precedentemente allocata con VirtualAlloc. Prende in input il puntatore alla memoria allocata e alcune opzioni aggiuntive, come il tipo di deallocazione da eseguire.
- **ExitProcess**: Utilizzata per terminare il processo chiamante e restituire il controllo al sistema operativo. Prende in input un codice di uscita che rappresenta lo stato di uscita del processo. Quando viene chiamata, il processo termina immediatamente e tutte le risorse associate a esso vengono liberate.

ADVAPI32.dll: La libreria ADVAPI32.DLL (Advanced Services API) fornisce una serie di funzionalità avanzate di servizi. Queste funzioni sono principalmente incentrate sulla gestione dei servizi di sistema, sulla sicurezza e sui privilegi, oltre ad altre operazioni avanzate relative alla gestione dei token di sicurezza, alla crittografia, alla gestione degli eventi e molto altro. La funzione chiamata dall'eseguibile è solo una, Fig.5.

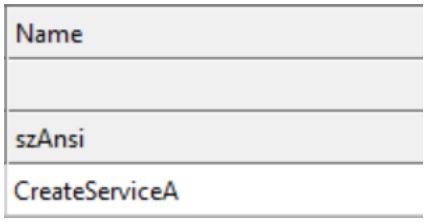
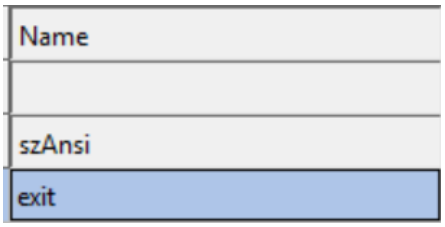


Fig.5

CreateServiceA: Questa funzione è utilizzata per creare un nuovo servizio di sistema in un sistema operativo Windows, consentendo di specificare le impostazioni del servizio e i diritti di accesso necessari. È una parte importante delle API di servizio di Windows e viene utilizzata da applicazioni e servizi per aggiungere nuovi servizi al sistema operativo.

MSVCRT.dll: Questa libreria (Microsoft Visual C Runtime) fornisce una serie di funzioni essenziali per il runtime del linguaggio di programmazione C e C++. Questa libreria è parte dell'ambiente di sviluppo Microsoft Visual C++, ed è necessaria per eseguire programmi scritti in C o C++ su piattaforme Windows, vedi Fig.6.



Fig,6

exit: Questa funzione termina l'esecuzione del programma in c/c++ e restituisce il valore di stato del sistema operativo. Viene anche utilizzata per gestire delle condizioni di errore

WININET.dll: Questa libreria fornisce una serie di funzionalità per l'accesso a risorse su Internet in ambienti Windows. È parte integrante del sistema operativo e offre un'interfaccia per l'accesso a protocolli Internet come HTTP, HTTPS, FTP, e altri, vedi Fig 7

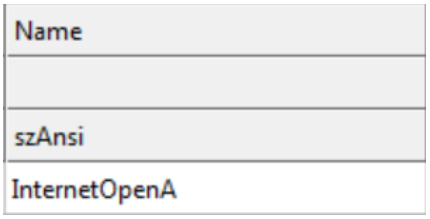


Fig.7

InternetOpenA: Questa funzione inizializza una sessione di accesso a Internet e restituisce un handle che consente all'applicazione di effettuare operazioni di rete utilizzando la libreria WinINet.

SEZIONE DELL'HEADER

L'header del formato PE ci indica anche le sezioni di cui sei compone il software, ogni sezione ha un preciso scopo, vedi Fig.8

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
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

Fig.8

UPX è uno strumento di compressione per file eseguibili: quando un file eseguibile viene compresso utilizzando UPX, il suo contenuto viene suddiviso in diverse sezioni, che UPX stesso gestisce. Una di queste sezioni, chiamata "UPX0", di solito contiene il nucleo dell'eseguibile dopo la compressione, le altre sezioni vengono numerate in modo incrementale, infatti abbiamo UPX0, UPX1, UPX2. Sembra che il malware abbia nascosto il vero nome delle sezioni, quindi non siamo in grado di capire cosa sono.

Considerazioni finali

Sembra essere un malware avanzato, è difficile eseguire un analisi statica basica perché il malware importa le librerie a tempo di esecuzione.