ANALISI STATICA BASICA DI UN MALWARE

Daniele Zizzi

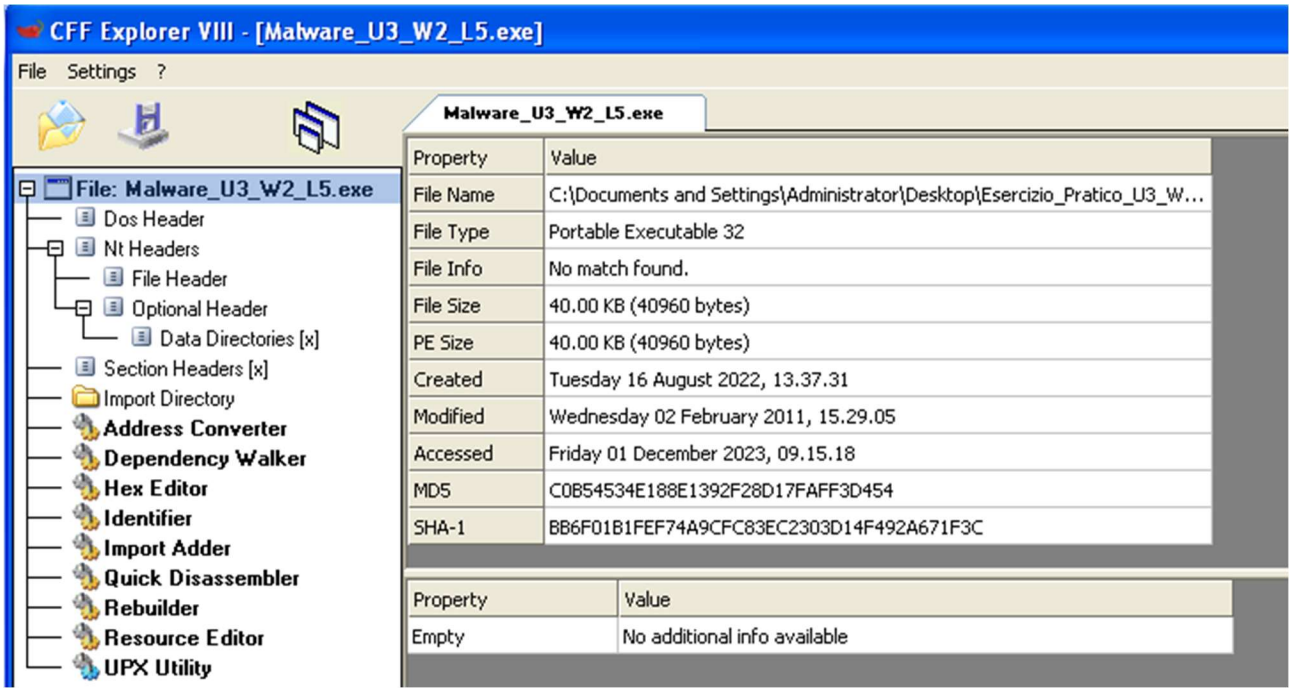
In questo esercizio andremo a visualizzare le parti di cui è composto il malware denominato Malware_U3_W2_L5.exe.

Un malware è un software/righe di codice che hanno come scopo la compromissione di un sistema.

Attraverso l'utilizzo del tool CFF Explorer, possiamo visualizzare la struttura del malware, leggere l'header del PE e le funzioni importate ed esportate.

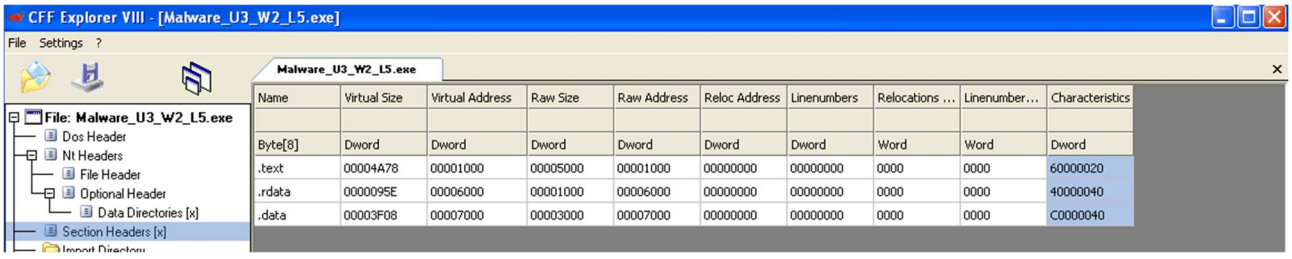
Il malware in questione è di tipo PE, quindi eseguibile portatile a 32 bit.

La pagina principale ci fornisce HASH MD5 e SHA-1, utili per eventuale confronto su virus total e similari.



Esso è composto dalle seguenti sezioni:

- .text Dove risiedono istruzioni che verranno eseguite dalla CPU non appena verrà avviato l'eseguibile
- .rdata Contiene le informazioni delle librerie e delle funzioni importate
- .data Contiene le variabili globali del programma. Sono globali poiché devono essere disponibili al di fuori delle funzioni, quindi in tutto il programma.



In importDirectory, troviamo le librerie e le funzioni richiamate dal programma:

- KERNEL32.DLL contiene le funzioni principali per interagire con il sistema operativo, quindi permette di manipolare i dati e gestire la memoria
- WININET.DLL permette di richiamare alcuni protocolli di rete come http(HYPERTEXT TRANSFER PROTOCOL) porta 80, FTP(FILE TRANSFER PROTOCOL) porta 21, NTP(NETWORK TIME PROTOCOL) porta 123.

Tra le funzioni richiamate ci sono LoadLibrary e GetProcAddress, che servono per caricare funzioni addizionali in runtime.

GetVersion recupera la versione sulla piattaforma del sistema operativo.

TerminateProcess termina il processo specificato e tutti i suoi thread.

WriteFile scrive in un file.

Malware_U3_W2_L5.exe

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000065EC	N/A	000064DC	000064E0	000064E4	000064E8	000064EC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000065E4	000065E4	0296	Sleep
00006940	00006940	027C	SetStdHandle
0000692E	0000692E	0156	GetStringTypeW
0000691C	0000691C	0153	GetStringTypeA
0000690C	0000690C	01C0	LCMapStringW
000068FC	000068FC	01BF	LCMapStringA
000068E6	000068E6	01E4	MultiByteToWideChar
00006670	00006670	00CA	GetCommandLineA
00006682	00006682	0174	GetVersion
00006690	00006690	007D	ExitProcess
0000669E	0000669E	029E	TerminateProcess
000066B2	000066B2	00F7	GetCurrentProcess
000066C6	000066C6	02AD	UnhandledExceptionFilter
000066E2	000066E2	0124	GetModuleFileNameA
000066F8	000066F8	00B2	FreeEnvironmentStringsA
00006712	00006712	00B3	FreeEnvironmentStringsW
0000672C	0000672C	02D2	WideCharToMultiByte
00006742	00006742	0106	GetEnvironmentStrings
0000675A	0000675A	0108	GetEnvironmentStringsW
00006774	00006774	026D	SetHandleCount
00006786	00006786	0152	GetStdHandle
00006796	00006796	0115	GetFileType
000067A4	000067A4	0150	GetStartupInfoA
000067B6	000067B6	0126	GetModuleHandleA
000067CA	000067CA	0109	GetEnvironmentVariableA
000067E4	000067E4	0175	GetVersionExA
000067F4	000067F4	019D	HeapDestroy
00006802	00006802	019B	HeapCreate
00006810	00006810	02BF	VirtualFree
0000681E	0000681E	019F	HeapFree
0000682A	0000682A	022F	RtlUnwind
00006836	00006836	02DF	WriteFile
00006842	00006842	0199	HeapAlloc
0000684E	0000684E	00BF	GetCPInfo
0000685A	0000685A	00B9	GetACP
00006864	00006864	0131	GetOEMCP
00006870	00006870	02BB	VirtualAlloc
00006880	00006880	01A2	HeapReAlloc
0000688E	0000688E	013E	GetProcAddress
000068A0	000068A0	01C2	LoadLibraryA
000068B0	000068B0	011A	GetLastError
000068C0	000068C0	00AA	FlushFileBuffers
000068D4	000068D4	026A	SetFilePointer
00006950	00006950	001B	CloseHandle

Malware_U3_W2_L5.exe

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
00006664	N/A	000064F0	000064F4	000064F8	000064FC	00006500
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00006640	00006640	0071	InternetOpenUrlA
0000662A	0000662A	0056	InternetCloseHandle
00006616	00006616	0077	InternetReadFile
000065FA	000065FA	0066	InternetGetConnectedState
00006654	00006654	006F	InternetOpenA

Nella sezione corrente, avvengono chiamate per l'apertura di un URL, lettura di un file e ritorno dello stato di una connessione.

Utilizzando il tool strings di sysinternals, possiamo visualizzare tutte le stringhe presenti nell'eseguibile, senza avviarlo

```
C:\Documents and Settings\Administrator\Desktop\SysinternalsSuite>strings ..\Ese
rcizio_Pratico_U3_W2_L5\Malware_U3_W2_L5.exe
```

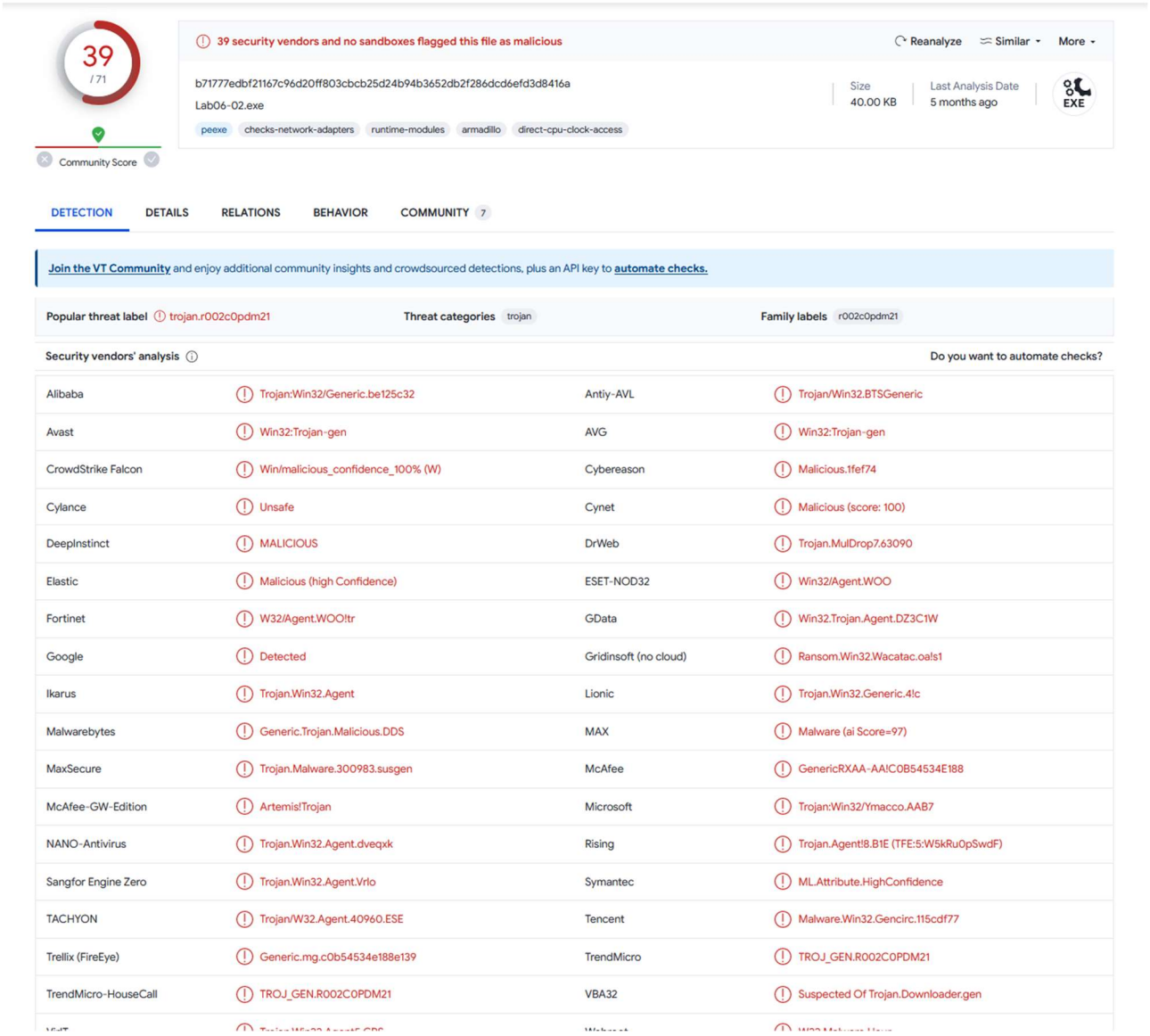
Oltre alle librerie e funzioni richiamate, apre un sito web chiamato www.praticalmalwareanalysis.com.

Dai dati acquisiti, sembra che il programma scriva in un file, apre una connessione ad Internet che con la quale legge il file creato. Quindi i dati vengono inviati all'esterno della macchina verso l'attaccante. Poiché la

connessione parte dalla vittima è un attacco di tipo reverse shell che permette di raggirare il firewall perimetrale. Oppure, potrebbe essere un downloader, poiché apre un sito di dubbia provenienza e scarica altri malware.

Confronto l'md5 con il database di virus total, in modo di avere un ulteriore riscontro circa il malware utilizzato.

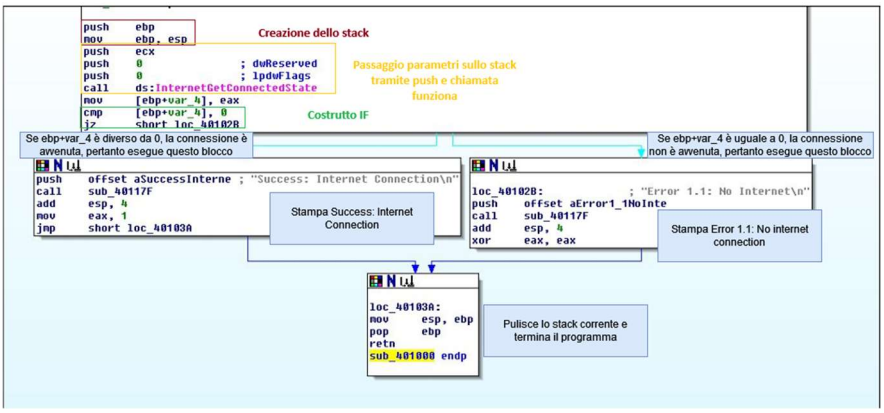
```
C:\Documents and Settings\Administrator\Desktop\md5deep-4.3>md5deep ..\Esercizio
_Pratico_U3_W2_L5\Malware_U3_W2_L5.exe
c0b54534e188e1392f28d17faff3d454  C:\Documents and Settings\Administrator\Deskto
p\Esercizio_Pratico_U3_W2_L5\Malware_U3_W2_L5.exe
```



Il Malware viene identificato come Trojan. Un trojan è un codice malevolo che all'apparenza può sembra un file legittimo. Può iniettarsi in processi di sistema legittimi, al fine di celare la propria identità.

ANALISI CODICE ASSEMBLY

Il programma in questione, verifica se l'host è connesso ad internet. Questa informazione può essere utilizzata da un malware, al fine di verificare la raggiungibilità di un host ed effettuare un attacco.



I costrutti sono sequenze di istruzioni che svolgono un'azione specifica. Controllano il flusso dell'esecuzione del programma, esecuzione di operazioni matematiche o logiche, accesso ai dati.

I comandi utilizzati per manipolare lo stack sono Push e Pop.

Push inserisce dati in cima allo stack.

Pop toglie dati dalla cima dello stack.

Mov permette di copiare un dato. Pur chiamandosi mov(move quindi muovere/spostare), non sposta realmente il dato, ma lo copia solamente, poiché lo spostamento, comporta l'eliminazione del dato dalla sorgente.

Call permette di richiamare delle funzioni, es. stampa.

Add somma sorgente con destinazione.

XOR Porta logica OR ma negata. È utilizzata per resettare un valore di un registro o variabile.

Jmp salta alla locazione di memoria indicata.

JZ, previa comparazione dei dati(Cmp), salta ad una locazione di memoria se il flag ZF(Zero flag) è settato, pertanto se ha come valore 1.

Considerazioni sul programma

Il programma verifica se c'è una connessione attiva, controllando il risultato della chiamata alla funzione "internetgetconnectedstate".

Spiegazione codice riga per riga

1. push ebp - Immette il valore corrente del puntatore base (ebp) nello stack.
2. mov ebp, esp – Copia esp in ebp
3. push ecx - Salva il valore corrente del registro ecx nello stack.
4. push 0 - Passa 0 come parametro alla funzione InternetGetConnectedState.
5. push 0 - Passa un altro 0 come parametro alla funzione InternetGetConnectedState.
6. call ds:InternetGetConnectedState - Chiama la funzione InternetGetConnectedState per verificare la connessione a Internet.
7. mov [ebp+var_4], eax - Salva il risultato della funzione InternetGetConnectedState in una variabile locale.
8. cmp [ebp+var_4], 0 - Confronta il risultato della funzione InternetGetConnectedState con 0 ed imposta lo ZF a 1 se la variabile è uguale a 0, setta a 0 se è diversa.
9. jz short loc_40102B - Se lo ZF è settato (cioè non c'è connessione a Internet), salta all'etichetta loc_40102B.

Se la connessione a Internet è riuscita:

10. push offset aSuccessInterne - Passa l'indirizzo della stringa "Success: Internet Connection\n" come parametro alla funzione sub_40117F.
11. call sub_40117F - Chiama la funzione sub_40117F per stampare la stringa.
12. add esp, 4 - Pulisce lo stack dopo la chiamata alla funzione.
13. mov eax, 1 - Imposta il valore di ritorno a 1.
14. jmp short loc_40103A - Salta all'etichetta loc_40103A.

Se la connessione a Internet non è riuscita:

15. push offset aError1_NoInte - Passa l'indirizzo della stringa "Error 1.1: No Internet\n" come parametro alla funzione sub_40117F.
16. call sub_40117F - Chiama la funzione sub_40117F per stampare la stringa.
17. add esp, 4 - Pulisce lo stack dopo la chiamata alla funzione.
18. xor eax, eax - Imposta il valore di ritorno a 0.

Infine:

19. mov esp, ebp - Ripristina il puntatore stack (esp) al valore del puntatore base (ebp).
20. pop ebp - Ripristina il valore del puntatore base (ebp) dallo stack.
21. retn - Ritorna dal sottoprogramma.