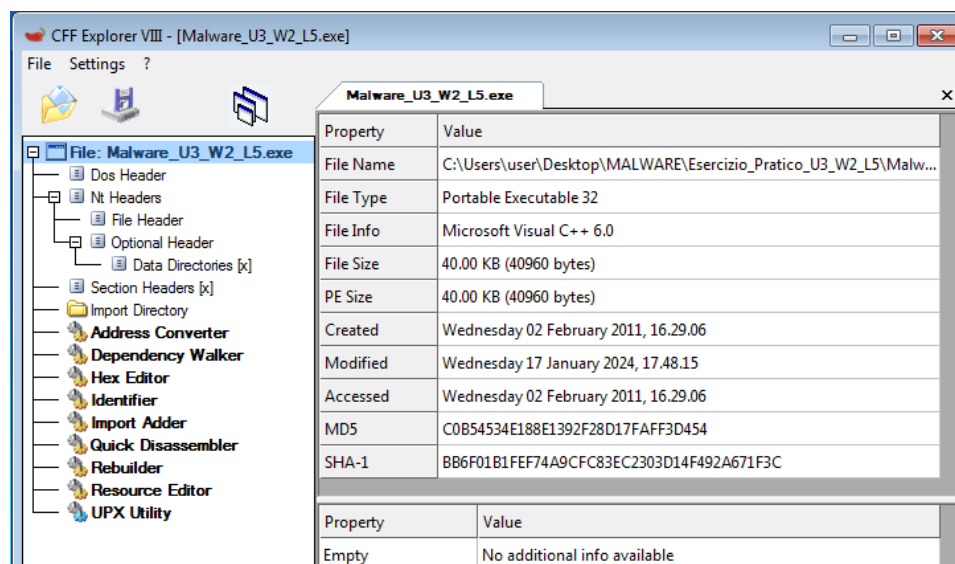
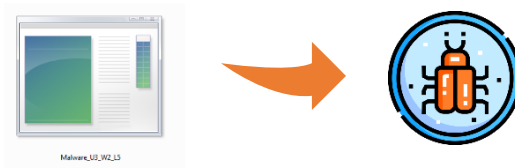


Progetto S10/L5: Analisi statica e Codice Assembly

Parte 1

Analisi statica: Malware_U3_W2_L5

Nella fase iniziale del nostro esercizio, procederemo con un'Analisi statica di base. Per questo, utilizzeremo una macchina virtuale completamente isolata, priva di connessione internet e di qualsiasi collegamento con dispositivi esterni come cartelle condivise o porte USB attive. Questo ambiente sicuro è essenziale per condurre un'analisi accurata di un potenziale malware. Il nostro obiettivo è analizzare il file eseguibile fornito per questa esercitazione. Poiché stiamo eseguendo un'analisi statica, non avvieremo effettivamente il malware, ma ci limiteremo ad esaminare il suo contenuto. A tale scopo, useremo lo strumento CFF Explorer, che ci consente di estrarre informazioni rilevanti sul malware, come le librerie che importa e le sezioni del file. Per iniziare, apriremo il nostro strumento e caricheremo il file del malware.



In seguito, ci spostiamo verso le opzioni laterali e selezioniamo la sezione **"Import Directory"**, dalla quale possiamo identificare le librerie che il malware utilizza nel sistema bersaglio.

Le librerie importate dal file

La sezione **"Import Directory"** fornisce informazioni sulle librerie esterne (DLL) che il file eseguibile corrente sta importando durante l'esecuzione. Questa sezione elenca le librerie DLL che il programma utilizza per accedere a funzionalità esterne o condivise, come funzioni di sistema, servizi di rete e altro ancora. Per ciascuna libreria importata, vengono elencati i dettagli, come il nome della libreria, l'indirizzo virtuale in cui è stata caricata, nonché le funzioni specifiche importate da quella libreria.

Queste informazioni sono fondamentali per comprendere le dipendenze esterne di un file eseguibile e per l'analisi dei malware, in quanto consentono di identificare le risorse esterne utilizzate dal programma durante l'esecuzione.

CFF Explorer VIII - [Malware_U3_W2_L5.exe]

File Settings ?

Malware_U3_W2_L5.exe

| Module Name | Imports | OFTs | TimeDateStamp | ForwarderChain | Name RVA | FTs (IAT) |
|--------------|--------------|----------|---------------|----------------|----------|-----------|
| szAnsi | (nFunctions) | Dword | Dword | Dword | Dword | Dword |
| KERNEL32.dll | 44 | 00006518 | 00000000 | 00000000 | 000065EC | 00006000 |
| WININET.dll | 5 | 000065CC | 00000000 | 00000000 | 00006664 | 000060B4 |

Tra queste, notiamo la presenza della libreria **KERNEL32.dll**, dalla quale il malware importa 44 funzioni. Questa libreria è ampiamente diffusa e contiene le funzioni fondamentali per l'interazione con il sistema operativo e i filesystem. Inoltre, individuiamo la libreria **WININET.dll**, dalla quale il malware importa 5 funzioni. Questa libreria, invece, include le funzioni necessarie per implementare alcuni protocolli di rete, come HTTP, FTP e NTP.

Cliccando sulle librerie, possiamo esaminare le funzioni che vengono importate.

CFF Explorer VIII - [Malware_U3_W2_L5.exe]

File Settings ?

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 |

CFF Explorer VIII - [Malware_U3_W2_L5.exe]

File Settings ?

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 |

La libreria **KERNEL32.dll** di Windows fornisce una serie di funzionalità cruciali per il sistema operativo. Tuttavia, se utilizzata da un malware, può essere sfruttata per avviare processi dannosi, manipolare file e registro di sistema, controllare altri processi, e gestire la memoria, consentendo al malware di eseguire una vasta gamma di attività dannose e compromettere la sicurezza del sistema infetto.

Vediamo qualche funzione in particolare della libreria **KERNEL32.dll**.

La funzione **Sleep** della libreria Kernel32.dll introduce ritardi nell'esecuzione delle attività malevole, mascherando il comportamento dannoso, evitando la rilevazione da parte dei sistemi di sicurezza, eludendo i controlli di sicurezza attivi e distribuendo gradualmente il carico delle operazioni dannose nel tempo, il che riduce il rischio di rilevazione e permette al malware di operare in modo più discreto.

La funzione **SetStdHandle** se utilizzata da un malware, potrebbe essere impiegata per manipolare i flussi di input/output standard del processo corrente in modi dannosi, come il dirottamento di output per nascondere informazioni o per ridirigere i flussi di input/output per interagire con altri processi dannosi o con risorse dannose.

La funzione **GetStringTypeW** può essere impiegata per analizzare, identificare e manipolare stringhe di testo sensibili presenti sul sistema infetto, facilitando l'individuazione di informazioni critiche, l'attivazione di azioni dannose, l'elusione della rilevazione da parte degli strumenti di sicurezza e persino la decodifica di stringhe crittografate. In sostanza, questa funzione può essere sfruttata per mettere a rischio la sicurezza e la privacy degli utenti, contribuendo alle attività malevole del malware.

Le funzioni della libreria **WININET.dll**, se utilizzate da un malware, consentono la comunicazione con server di comando e controllo, il download di codice malevolo, il furto di dati sensibili, la distribuzione di malware e attacchi di tipo DoS. Questo amplia le capacità dannose del malware, consentendo agli attaccanti di eseguire una serie di attività dannose e compromettere la sicurezza dei sistemi infetti.

Vediamo qualche funzione in particolare della libreria **WININET.dll**.

La funzione **InternetOpenUrlA**, presente nella libreria WinINet.dll di Windows, consente l'apertura di connessioni a risorse tramite URL su Internet o reti locali. Tuttavia, se utilizzata in un contesto malevolo da parte di malware, può consentire azioni dannose come il download e l'esecuzione di codice dannoso, la comunicazione con server di comando e controllo per ricevere istruzioni o trasferire dati, la distribuzione di malware aggiuntivi e la raccolta di informazioni sensibili. In sintesi, l'utilizzo di questa funzione da parte di malware potrebbe compromettere la sicurezza e l'integrità dei sistemi infetti, nonché la privacy degli utenti.

La funzione **InternetCloseHandle** è utilizzata per chiudere le connessioni di rete aperte in programmi Windows. Nei malware, può essere usata per interrompere le comunicazioni con server di sicurezza, nascondere l'attività dannosa, effettuare attacchi di tipo "denial-of-service" o mascherare la presenza del malware nel sistema, complicando la sua individuazione e rimozione.

La funzione **InternetReadFile** nei malware può essere utilizzata per recuperare dati sensibili, scaricare e eseguire codice malevolo, aggiornare il malware, distribuire payload dannosi e comunicare con server di comando e controllo. Questo amplia il potenziale dannoso del malware consentendo di raccogliere informazioni, espandere le proprie funzionalità e coordinare attività dannose attraverso le connessioni Internet.

La funzione **InternetGetConnectedState** è utilizzata dai malware per verificare la connettività Internet del sistema, adattare il comportamento in base allo stato della connessione (come l'avvio di attività dannose solo quando la connessione è attiva), raccogliere informazioni sulle abitudini di navigazione dell'utente e intensificare gli attacchi quando la connessione è disponibile. Questo permette al malware di prendere decisioni più mirate e di condurre attività dannose in modo più efficace.

Dopo aver identificato le librerie importate dal nostro malware, procediamo a individuare il numero di sezioni che compongono il file.

Le sezioni di cui si compone il file.

La sezione **"Section Headers"** fornisce informazioni dettagliate sulle varie sezioni del file eseguibile. Queste informazioni includono, ad esempio, i nomi delle sezioni, gli indirizzi in memoria in cui sono allocate, le dimensioni delle sezioni, i permessi di accesso (come lettura, scrittura, esecuzione) e altre informazioni utili per comprendere la struttura del file.

Questa sezione è essenziale per l'analisi del malware, in quanto consente di identificare e comprendere le varie parti del file eseguibile, incluso il codice eseguibile, i dati statici, le tabelle di importazione ed esportazione delle funzioni.

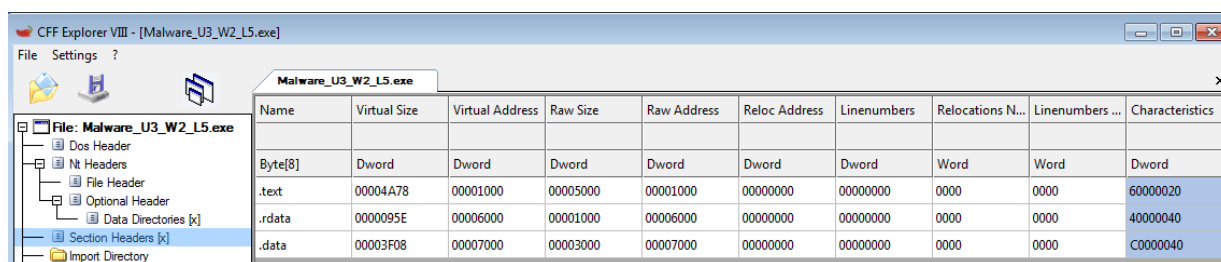
Selezionando la voce **"Section Headers"** dalle opzioni laterali, osserviamo che il file è suddiviso in tre sezioni.

Le sezioni individuate sono le seguenti:

".text", che contiene le istruzioni o il codice che la CPU eseguirà una volta avviato il software;

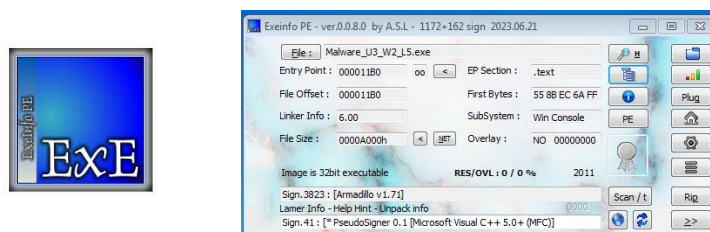
".rdata", che generalmente contiene informazioni sulle librerie e le funzioni importate ed esportate dal file eseguibile;

".data", che tipicamente contiene i dati o le variabili globali del file eseguibile, accessibili da qualsiasi parte del programma.

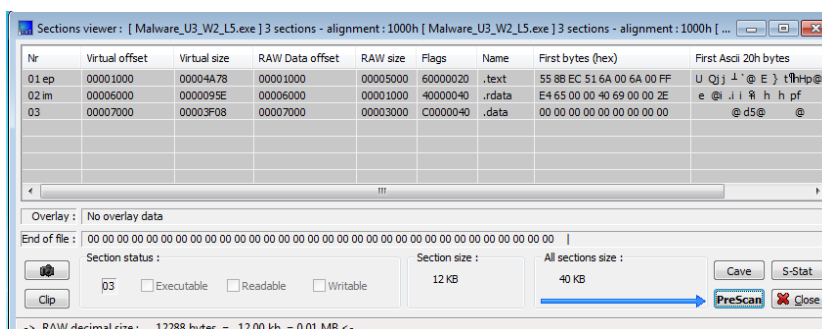


| Name | Virtual Size | Virtual Address | Raw Size | Raw Address | Reloc Address | Linenumbers | Relocations N... | Linenumbers ... | Characteristics |
|---------|--------------|-----------------|----------|-------------|---------------|-------------|------------------|-----------------|-----------------|
| Byte[8] | Dword | Dword | Dword | Dword | Dword | Dword | Word | Word | Dword |
| .text | 00004A78 | 00001000 | 00005000 | 00001000 | 00000000 | 00000000 | 0000 | 0000 | 60000020 |
| .rdata | 0000095E | 00006000 | 00001000 | 00006000 | 00000000 | 00000000 | 0000 | 0000 | 40000040 |
| .data | 00003F08 | 00007000 | 00003000 | 00007000 | 00000000 | 00000000 | 0000 | 0000 | C0000040 |

Inoltre possiamo utilizzare il tool **"ExeinfoPE"**, uno strumento per l'analisi dei file eseguibili di Windows, identificando il formato, l'architettura, e le librerie importate. Fornisce informazioni dettagliate sulle sezioni del file e sulle dipendenze esterne, utili per l'analisi dei malware.



Dopo aver caricato il file eseguibile, clicchiamo sull'icona "section viewer", questo avvierà un pannello che mostra i dettagli di ciascuna sezione del file. Le sezioni elencate sono le stesse che abbiamo individuato utilizzando CFF Explorer.

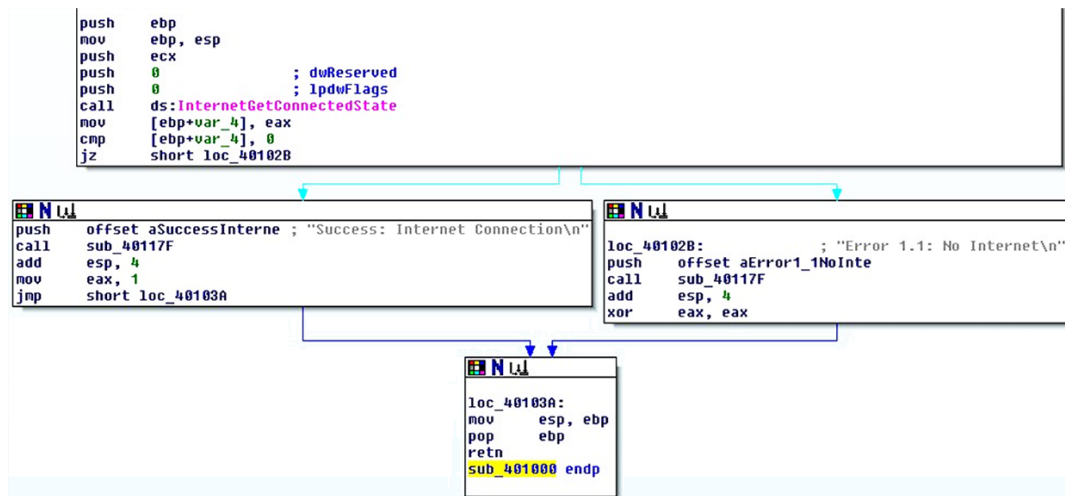


| Nr | Virtual offset | Virtual size | RAW Data offset | RAW size | Flags | Name | First bytes (hex) | First Ascii 20h bytes |
|-------|----------------|--------------|-----------------|----------|----------|--------|----------------------------|---------------------------|
| 01 ep | 00001000 | 00004A78 | 00001000 | 00005000 | 60000020 | .text | 55 8B EC 51 6A 00 6A 00 FF | U Q j 1 ' @ E } t 1 h p @ |
| 02 im | 00006000 | 0000095E | 00006000 | 00001000 | 40000040 | .rdata | E4 65 00 00 40 69 00 00 2E | e @ i i 9 h h p f |
| 03 | 00007000 | 00003F08 | 00007000 | 00003000 | C0000040 | .data | 00 00 00 00 00 00 00 00 | @ d 5 @ @ |

Parte 2

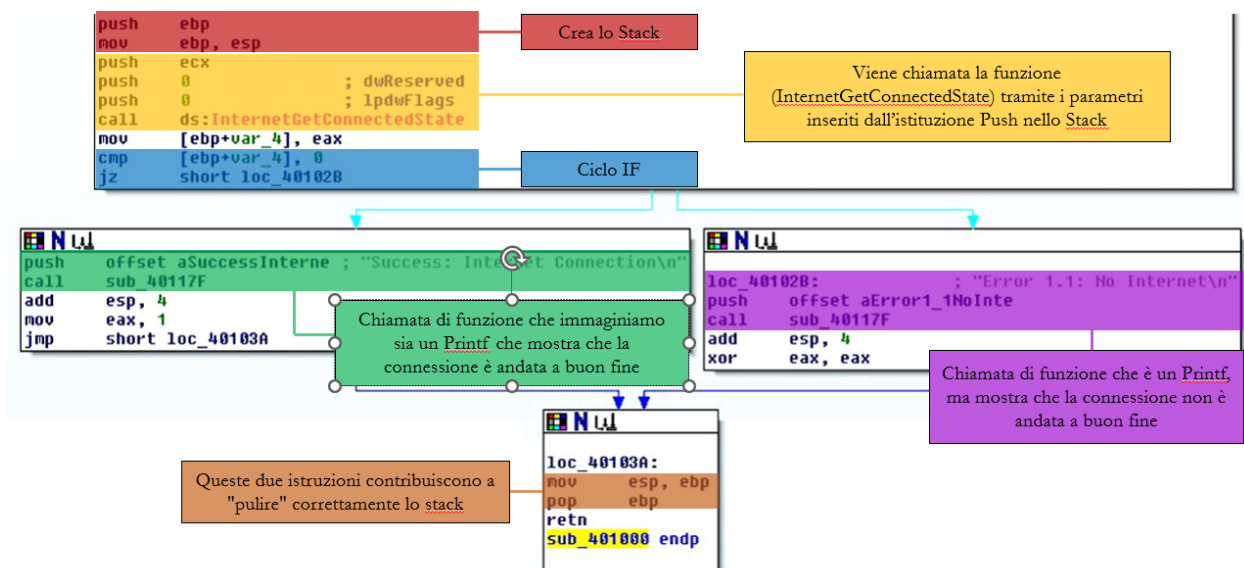
Codice assembly e costrutti noti

Nella seconda parte dell'esercitazione, andiamo ad esaminare il codice assembly di un malware.



Dopo la nostra analisi abbiamo individuato 6 costrutti noti.

Dall'analisi, emerge che il malware esegue una funzione di controllo della connessione sulla macchina bersaglio, utilizzando la chiamata di funzione **"InternetGetConnectedState"**.



Possiamo ipotizzare che la funzionalità implementata nel malware sia finalizzata a verificare lo stato di connessione Internet della macchina bersaglio. Se il valore di ritorno della funzione è diverso da zero, indicando una connessione attiva, il malware potrebbe procedere con determinate azioni o attivare funzionalità specifiche dipendenti dalla presenza di una connessione attiva.

Al contrario, se il valore di ritorno è zero, indicando l'assenza di connessione, il malware potrebbe eseguire azioni diverse o attendere il ripristino della connessione prima di proseguire con le operazioni malevole. In generale, la funzionalità potrebbe essere utilizzata per adattare il comportamento del malware in base allo stato di connessione della macchina bersaglio, consentendo agli attaccanti di eseguire azioni mirate o di ottenere informazioni sul sistema infetto.