PROGETTO SETTIMANA 11

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

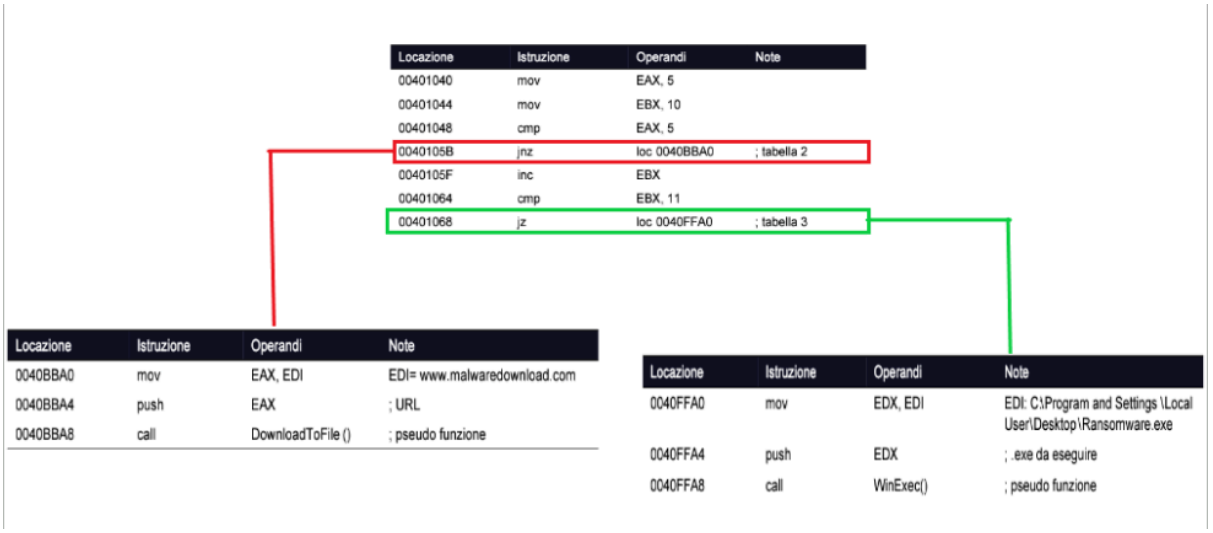
Il termine "Malware" è una contrazione di "software dannoso" (malicious software). Si tratta di programmi progettati per infiltrarsi, danneggiare o ottenere accesso non autorizzato a sistemi informatici o dati, spesso senza il consenso degli utenti. Esistono diversi tipi di malware, tra cui virus, worm, trojan, ransomware e molti altri.

L'istruzione "jz", essa fa parte del linguaggio assembly, il linguaggio di basso livello utilizzato dai processori per eseguire istruzioni. In particolare, "jz" è un'istruzione di salto condizionale. Ciò significa che il flusso di esecuzione del programma salta a un'altra parte del codice solo se una condizione specificata è soddisfatta. In questo caso, "jz" si traduce in "salta se zero" (jump if zero).

Il Malware effettua il salto condizionale presente alla locazione di memoria 00401068. L'istruzione "jz" salta alla locazione di memoria 00401068 solo se il confronto (istruzione "cmp") precedente ha dato come risultato che i due operandi, in questo caso EBX e 11, sono uguali. Questo suggerisce che il Malware sta utilizzando questa istruzione per prendere decisioni basate sullo stato dei registri e del sistema durante l'esecuzione, probabilmente come parte di una logica di controllo o di un algoritmo di comportamento.

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

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.**



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

Il Malware implementa due funzionalità, ovvero scaricamento del file ed esecuzione del file. Ora andiamo a descriverli nello specifico:

Scaricamento del file malevolo: La parte del codice situata a "loc0040BBA0" sembra coinvolgere il download di un file malevolo da un URL specifico. Il malware memorizza l'URL nel registro EAX e poi chiama la funzione "DownloadToFile()" per scaricare il file. Questo processo potrebbe essere parte di un'operazione di infezione, in cui il malware si scarica e si installa ulteriori componenti o payload dannosi.

Esecuzione del file malevolo: La parte del codice situata a "loc0040FFA0" coinvolge l'esecuzione di un file specifico. Il malware sembra memorizzare il percorso del file eseguibile nel registro EDX e poi chiamare la funzione "WinExec()" per avviare

l'esecuzione del file. Questo può essere il processo principale per l'avvio effettivo del malware sul sistema bersaglio.
In sintesi, il malware sembra avere la capacità di scaricare e installare file malevoli da un URL specifico e successivamente eseguire tali file sul sistema bersaglio. Queste funzionalità possono essere utilizzate per scopi dannosi come l'infezione del sistema, il furto di informazioni sensibili o altre attività dannose.

4) **Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione . Aggiungere eventuali dettagli tecnici/teorici.**

Le istruzioni "call" vengono utilizzate per chiamare una procedura o una funzione all'interno del programma. Durante una chiamata di funzione, gli argomenti devono essere passati in modo appropriato per garantire che la funzione chiamata riceva i dati corretti.

Chiamata alla funzione DownloadToFile() (a "loc0040BBA8"):

- **Prima della chiamata, il valore di interesse, l'URL, viene memorizzato nel registro EAX mediante l'istruzione "mov EAX, EDI" a "loc0040BBA0".**
- **Successivamente, il contenuto del registro EAX (che contiene l'URL) viene spinto nello stack con l'istruzione "push EAX" a "loc0040BBA4". Questo passaggio prepara l'argomento per la funzione DownloadToFile().**
- **Infine, la chiamata alla funzione DownloadToFile() viene effettuata con l'istruzione "call DownloadToFile()" a "loc0040BBA8".**

Chiamata alla funzione WinExec() (a "loc0040FFA8"):

- **Prima della chiamata, il valore di interesse, il percorso del file eseguibile, viene memorizzato nel registro EDX mediante l'istruzione "mov EDX, EDI" a "loc0040FFA0".**
- **Successivamente, il contenuto del registro EDX (che contiene il percorso del file eseguibile) viene spinto nello stack con l'istruzione "push EDX" a "loc0040FFA4". Questo passaggio prepara l'argomento per la funzione WinExec().**

- **Infine, la chiamata alla funzione WinExec() viene effettuata con l'istruzione "call WinExec()" a "loc0040FFA8".**

Questo processo di passaggio degli argomenti tramite i registri prima di spingerli nello stack è comune nelle chiamate di funzione nell'assembly x86. L'utilizzo degli stack permette di passare dati tra le funzioni e di mantenere lo stato del programma durante le chiamate di funzione.