Progetto S11/L5

Report sull'Analisi del Codice Assembly

Indice

1.Introduzione

Il presente report fornisce un'analisi dettagliata del codice assembly fornito, concentrandosi su diversi aspetti cruciali per la comprensione del funzionamento del malware. Il codice fornito è stato suddiviso in tre tabelle, ciascuna delle quali contiene istruzioni specifiche e informazioni pertinenti per rispondere alle domande poste nella traccia.

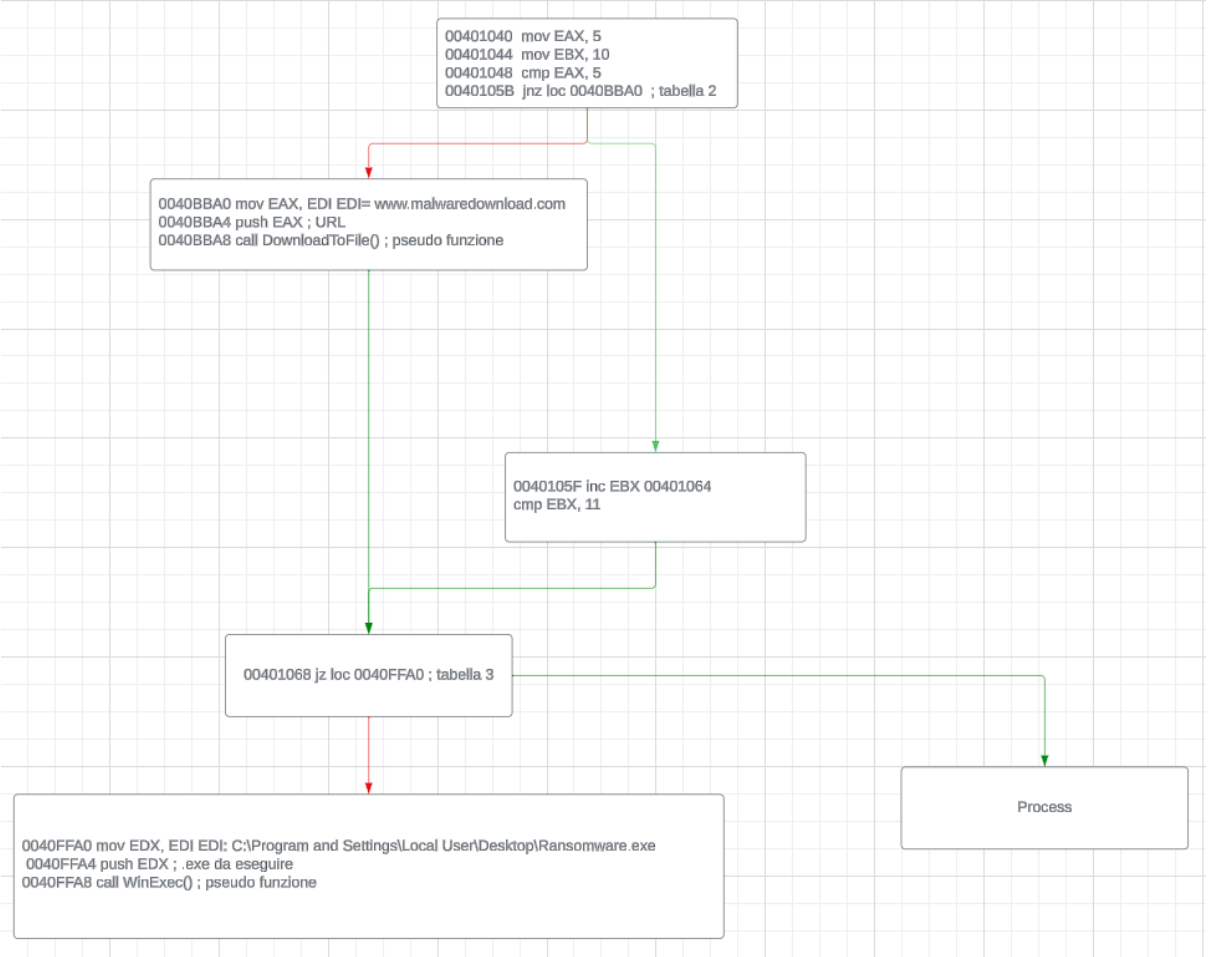
2. Spiegazione del Salto Condizionale

Il salto condizionale effettuato dal malware, come indicato nell'istruzione `jnz loc 0040BBA0`, è motivato dal confronto tra il contenuto del registro EAX e il valore 5. Se il confronto risulta essere diverso, il salto viene eseguito, indirizzando il flusso del programma verso la locazione di memoria 0040BBA0. Questo comportamento indica che il malware prende una decisione basata sul valore presente in EAX.

3. Diagramma di Flusso

Il diagramma di flusso illustra il percorso del programma attraverso le varie istruzioni e i salti condizionali. I salti effettuati sono evidenziati con linee rossi, mentre i salti non effettuati sono rappresentati da linee verdi.

Questo diagramma fornisce una chiara visualizzazione del flusso del programma e delle decisioni prese in base alle condizioni specificate.



4. Funzionalità Implementate nel Malware

Le diverse funzionalità implementate all'interno del malware includono il download di un file da un URL specifico e l'esecuzione di un file eseguibile specifico. Il primo è attuato tramite la chiamata alla funzione `DownloadToFile()`, mentre il secondo è realizzato attraverso la chiamata alla funzione `WinExec()`. Queste funzionalità suggeriscono che il malware potrebbe essere coinvolto in attività dannose, come il download e l'esecuzione di file dannosi.

5. Passaggio degli Argomenti alle Chiamate di Funzione

Le istruzioni "call" presenti nelle tabelle 2 e 3 sono responsabili della chiamata alle funzioni `DownloadToFile()` e `WinExec()`, rispettivamente. Gli argomenti vengono passati attraverso lo stack prima di effettuare la chiamata di funzione. Ad esempio, prima di chiamare `DownloadToFile()`, l'indirizzo dell'URL viene caricato nel registro EAX e successivamente inserito nello stack tramite l'istruzione `push EAX`.

6. Conclusioni

Il codice assembly fornito rivela il funzionamento di un malware con funzionalità specifiche per il download e l'esecuzione di file. L'analisi dettagliata fornisce una comprensione approfondita del comportamento del malware e delle sue caratteristiche principali, consentendo una migliore comprensione e mitigazione dei rischi associati.