Report sull'Analisi del Codice Malware

Data 16/02/24

Introduzione

Il seguente report analizza il codice di un malware per comprendere il suo comportamento, identificare le funzionalità implementate e spiegare i salti condizionali effettuati. Viene inoltre presentato un diagramma di flusso che illustra i percorsi del codice in base alle condizioni.

Analisi del codice

Di seguito è riportato il primo blocco di codice fornitoci per l'analisi:

Locazione	Istruzione	Operandi
00401040	mov	EAX, 5
00401044	mov	EBX, 10
00401048	cmp	EAX, 5
0040105B	jnz	loc 0040BBA0
0040105F	inc	EBX
00401064	cmp	EBX, 11
00401068	jz	loc 0040FFA0

1. **00401040 mov EAX, 5**: Questa istruzione sposta il valore 5 nel registro EAX. In altre parole, assegna il valore 5 al registro EAX.
2. **00401044 mov EBX, 10**: Questa istruzione sposta il valore 10 nel registro EBX. Similmente alla riga precedente, assegna il valore 10 al registro EBX.
3. **00401048 cmp EAX, 5**: Questa istruzione confronta il valore contenuto nel registro EAX con il valore 5. È un confronto che imposta i flag di stato della CPU in base al risultato.
4. **0040105B jnz loc_0040BBA0, 0040105F**: Questa istruzione esegue un salto condizionale alla locazione **0040BBA0**.

5. **0040105F inc EBX**: Questa istruzione incrementa il valore contenuto nel registro EBX di uno. Se il salto condizionale non è stato eseguito, cioè se il confronto nella riga precedente ha dato esito zero, questa istruzione verrà eseguita.
6. **00401064 cmp EBX, 11**: Questa istruzione confronta il valore contenuto nel registro EBX con il valore 11. È simile al confronto nella riga 3.
7. **00401068 jz loc_0040FFA0**: Questa istruzione esegue un salto condizionale alla locazione **0040FFA0** .

Analisi dei secondo blocchi di codice :

Locazione	Istruzione	Operandi
0040BBA0	mov	EAX, EDI
0040BBA4	push	EAX
0040BBA8	call	DownloadToFile()

1. **0040BBA0 mov EAX, EDI**: Questa istruzione sposta il valore contenuto nel registro EDI nel registro EAX. Nella descrizione specifica del contesto, il valore di EDI corrisponde ad un URL (www.malwaredownload.com).
2. **0040BBA4 push EAX**: Questa istruzione mette il valore contenuto nel registro EAX (presumibilmente l'URL) nello stack. Questo potrebbe essere un parametro da passare a una funzione successiva.
3. **0040BBA8 call DownloadToFile()**: Questa istruzione chiama una funzione denominata **DownloadToFile()**, probabilmente passando l'URL come parametro. La funzione **DownloadToFile()** sembra essere responsabile del download di un file dalla rete.

Analisi del terzo blocco di codice:

Locazione	Istruzione	Operandi
0040FFA0	mov	EDX, EDI
0040FFA4	push	EDX
0040FFA8	call	WinExec()

1. **0040FFA0 mov EDX, EDI**: Questa istruzione sposta il valore contenuto nel registro EDI nel registro EDX. Nella descrizione specifica del contesto, il valore di EDI sembra essere il percorso di un file eseguibile (C:\Program and Settings\Local User\Desktop\Ransomware.exe).
2. **0040FFA4 push EDX**: Questa istruzione mette il valore contenuto nel registro EDX (presumibilmente il percorso del file eseguibile) nello stack. Questo potrebbe essere un parametro da passare a una funzione successiva.
3. **0040FFA8 call WinExec()**: Questa istruzione chiama una funzione denominata **WinExec()**, probabilmente passando il percorso del file eseguibile come parametro. La funzione **WinExec()** è responsabile dell'esecuzione di un file eseguibile su Windows.

Salto Condizionale Effettuato

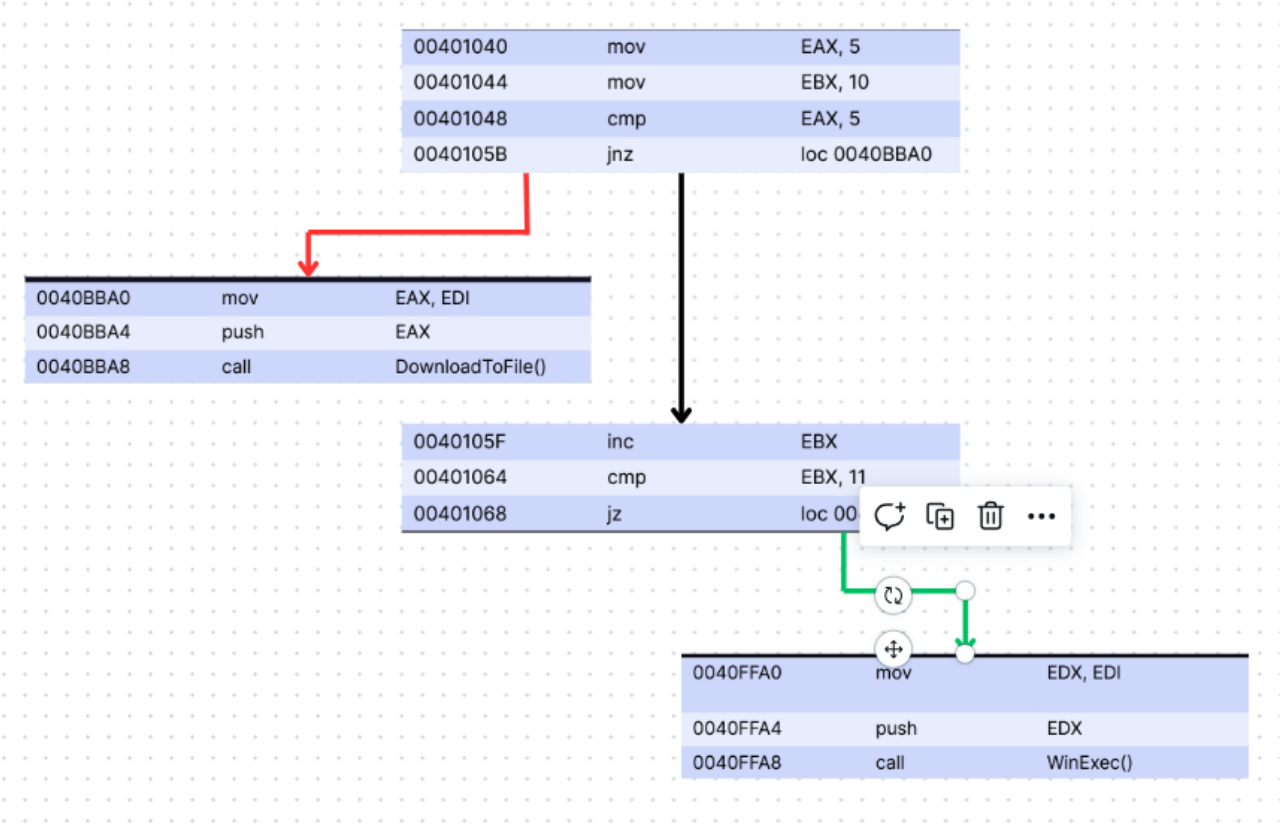
Il malware effettua un salto condizionale basato sul valore di una variabile. Le istruzioni cmp EAX, 5 e jnz loc_0040BBA0 confrontano il contenuto del registro EAX con il valore 5 e eseguono un salto condizionale alla locazione 0040BBA0 se il confronto precedente non ha dato esito positivo.

Nello specifico la funzione “jnz” esegue un salto condizionale solo se il risultato dell’operazione di comparazione è diverso da 0, il salto avviene se il flag (ZF) è a zero (il valore del flag ZF è 0 se è diverso da 0 ed è 1 se il risultato della funzione precedente è 0).

Allo stesso modo, l'istruzione jz loc_0040FFA0 esegue un salto condizionale alla locazione 0040FFA0 solo se il confronto precedente ha dato esito positivo (ovvero, se EBX è uguale a 11).

Il secondo salto condizionale viene eseguito dal malware dalla funzione “jz”, a differenza della funzione “jnz” questa esegue un salto condizionale solo se il risultato dell’operazione precedente è zero, quindi il flag (ZF) avrà come valore 1.

Diagramma di Flusso dei Salti Condizionali



Nel diagramma di flusso, i salti condizionali effettuati sono rappresentati da linee verdi, mentre i salti non effettuati sono rappresentati da linee rosse. Questa rappresentazione grafica fornisce una panoramica visiva chiara dei percorsi presi dal malware in base alle condizioni nel codice.

Possiamo quindi confermare che il primo salto non viene effettuato, questo perché il risultato della funzione " cmp EAX, 10 " confronta il valore 10 con il valore del registro EAX che come valore ha 5. Il confronto restituisce 0 come valore, la funzione jnz effettua il salto condizionale solo se il valore se il flag ZF ha valore 0. Come descritto in precedenza il flag ZF avrà valore 0 solo se il risultato dell'operazione è diverso da 0. In questo caso il valore è 0 quindi il flag ZF avrà 1 come valore ed il salto non viene effettuato.

Nel secondo salto invece, il confronto della funzione " cmp EBX, 11 " compara il valore 11 con il valore del registro EBX. Il registro EBX inizialmente ha valore 10 " mov EBX, 10 ", successivamente il valore viene incrementato di un intero con la funzione " inc EBX ". Il valore del registro è 11.

Successivamente avviene la funzione di comparazione che darà come risultato 0. In questo caso il salto condizionale avviene perché la funzione " jz " effettua un salto condizionale solo se il valore dell'operazione precedente è 0 ed il flag ZF sarà 1.

Funzionalità Implementate nel Malware

Le diverse funzionalità implementate nel malware includono operazioni di controllo del flusso, manipolazione di variabili e interazione con altri elementi del sistema. In particolare, il malware sembra scaricare un file da un URL specifico tramite la chiamata di funzione DownloadToFile() e successivamente se non dovesse riuscire ad eseguire il download esegue un file .exe tramite la chiamata di funzione WinExec().

Entrambe le funzioni richiedono un argomento che specifica il percorso del file da scaricare o eseguire, passato rispettivamente nei registri EAX ed EDX.

Conclusioni

L'analisi del codice del malware fornisce una comprensione approfondita del suo comportamento e delle sue funzionalità. Attraverso l'identificazione dei salti condizionali, la rappresentazione grafica tramite diagramma di flusso e l'analisi delle istruzioni specifiche, è stato possibile ottenere una visione chiara delle azioni eseguite dal malware e delle sue possibili implicazioni per la sicurezza del sistema.